

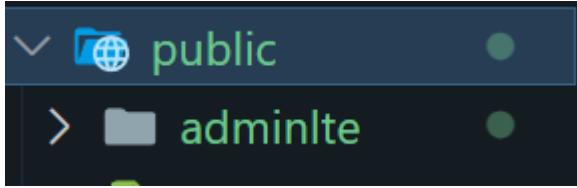
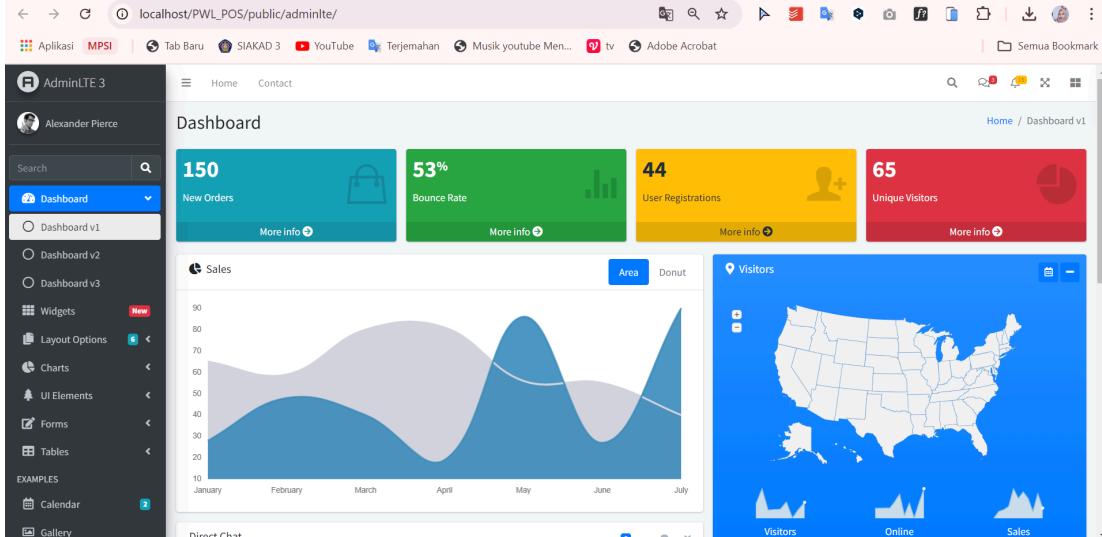


Mata Kuliah : Pemrograman Web Lanjut (PWL)
Program Studi : D4 – Sistem Informasi Bisnis
Semester : 5

Kelas : SIB 3D
NIM : 2241760030
Nama : Fardiyani Afro'ul Yasinta
Jobsheet Ke- : 5
Link Github : https://github.com/Fafayasinta/PWL_POS.git

Laporan Jobsheet

Praktikum Ke-1 | Layouting AdminLTE

Langkah	Jawaban/Deskripsi
1.	Kita download AdminLTE v3.2.0 yang rilis pada 8 Feb 2022
2.	Selepas 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
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)



The screenshot shows the VS Code interface with the following details:

- EXPLORER:** Shows the project structure:
 - PWL_POS
 - public
 - .htaccess
 - favicon.ico
 - index.php
 - robots.txt
 - resources
 - css
 - js
 - views
 - Layout
 - template.blade.php (selected)
 - kategori.blade.php
 - level.blade.php
- Code Editor:** Displays the content of `template.blade.php`:

```
<!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=block">
    <!-- 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>
<body class="hold-transition sidebar-mini">
```

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

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

The screenshot shows the head section of `template.blade.php` with lines 1 through 14 highlighted in blue, indicating the range of modification.

```
<!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=block">
    <!-- 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>
```

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

The screenshot shows the header section of `template.blade.php` with lines 19 through 153 removed, indicated by a large gap in the code.

```
<html lang="en">
</head>
<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>
```



11.	Kita modifikasi baris 25 dan 26 pada <code>template.blade.php</code> <pre><!-- Main Sidebar Container --> <aside class="main-sidebar sidebar-dark-primary elevation-4"> <!-- Brand Logo --> <img src="{{ assets('adminlte/dist/img/AdminLTELogo.png') }}" alt="AdminLTE Logo" PWL - Starter Code </pre>
12.	Selanjutnya kita blok baris 31-693 (baris untuk <i>element 2-sidebar</i>), lalu kita cut , dan paste -kan di file <code>PWL_POS/resource/view/layouts/sidebar.blade.php</code> (buat dulu file <code>sidebar.blade.php</code> jika belum ada). Sehingga tampilan dari file <code>template.blade.php</code> menjadi seperti berikut <pre>22 <!-- Main Sidebar Container --> 23 <aside class="main-sidebar sidebar-dark-primary elevation-4"> 24 <!-- Brand Logo --> 25 26 <img src="{{ assets('adminlte/dist/img/AdminLTELogo.png') }}" alt="Ac 27 PWL - Starter Code 28 29 30 <!-- Sidebar --> 31 @include('Layout.sidebar') 32 <!-- ./sidebar --> 33 </aside></pre>
13.	Selanjutnya perhatikan baris 87-98 (baris untuk <i>element 5-footer</i>), lalu kita cut , dan paste -kan di file <code>PWL_POS/resource/view/layouts/footer.blade.php</code> (buat file <code>footer.blade.php</code> jika belum ada). Sehingga tampilan dari file <code>template.blade.php</code> menjadi seperti berikut <pre>82 </section> 83 <!-- /.content --> 84 </div> 85 <!-- /.content-wrapper --> 86 87 @include('Layout.footer') 88 </div> 89 <!-- ./wrapper --> 90 91 <!-- jQuery --> 92 <script src=".../.../plugins/jquery/jquery.min.js"></script></pre>
14.	Kemudian kita modifikasi file <code>template.blade.php</code> baris 91-100 <pre>91 <!-- jQuery --> 92 <script src="{{ asset('adminlte/plugins/jquery/jquery.min.js') }}"></script> 93 <!-- Bootstrap 4 --> 94 <script src="{{ asset('adminlte/plugins/bootstrap/js/bootstrap.bundle.min.js') }}"></script> 95 <!-- AdminLTE App --> 96 <script src="{{ asset('adminlte/dist/js/adminlte.min.js') }}"></script> 97 </body> 98 </html></pre>



15.	Sekarang masuk pada bagian konten. Konten kita bagi menjadi 2, yaitu elemen untuk breadcrumb dan elemen untuk content .
16.	Perhatikan file <code>template.blade.php</code> pada baris 38-52 kita jadikan sebagai elemen 4-breadcrumb . Kita blok baris 38-52 lalu kita cut , dan paste -kan di file <code>PWL_POS/resource/view/layouts/breadcrumb.blade.php</code> (buat file <code>breadcrumb.blade.php</code> jika belum ada). Sehingga tampilan dari file <code>template.blade.php</code> menjadi seperti berikut
	<pre>37 <!-- Content Header (Page header) --> 38 @include('Layout.breadcrumb') 39 40 <!-- Main content --> 41 <section class="content"> 42 <!-- Default box --> 43 <div class="card"> 44 <div class="card-header"> 45 <h3 class="card-title">Title</h3> 46 </pre>
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 <code>template.blade.php</code> . dan kita ganti dengan kode seperti ini <code>@yield('content')</code>
	<pre><!-- 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></pre>
19.	Hasil akhir pada file utama <code>layouts/template.blade.php</code> adalah seperti berikut



```
1 <body class="hold-transition sidebar-mini">
2 <!-- Site wrapper -->
3 <div class="wrapper">
4   <!-- Navbar -->
5   @include('layouts.header')
6   <!-- /.navbar -->
7
8   <!-- Main Sidebar Container -->
9   <aside class="main-sidebar sidebar-dark-primary elevation-4">
10    <!-- Brand Logo -->
11    <a href="{{ url('/') }}" class="brand-link">
12      
13      <span class="brand-text font-weight-light">PWL - Starter Code</span>
14    </a>
15
16    <!-- Sidebar -->
17    @include('layouts.sidebar')
18    <!-- /.sidebar -->
19  </aside>
20
21  <!-- Content Wrapper. Contains page content -->
22  <div class="content-wrapper">
23    <!-- Content Header (Page header) -->
24    @include('layouts.breadcrumb')
25
26    <!-- Main content -->
27    <section class="content">
28      @yield('content')
29    </section>
30    <!-- /.content -->
31  </div>
32  <!-- /.content-wrapper -->
33
34  @include('layouts.footer')
35 </div>
36 <!-- /.wrapper -->
```

20.	Selamat kalian sudah selesai dalam melakukan layouting website di laravel.
21.	Jangan lupa commit dan push ke github untuk praktikum 1 ini

Praktikum Ke-2 | Penerapan Layouting

Langkah	Jawaban/Deskripsi
1.	Kita buat file controller dengan nama <code>WelcomeController.php</code>  <pre>app > Http > Controllers > WelcomeController.php > ... 1 <?php 2 3 namespace App\Http\Controllers; 4 5 6 class WelcomeController extends Controller{ 7 8 public function index(): Factory View { 9 \$breadcrumb = (object) [10 'title' => 'Selamat Datang', 11 'list' => ['Home', 'Welcome'] 12]; 13 14 \$activeMenu = 'dashboard'; 15 16 return view(view: 'welcome', data: ['breadcrumb' => \$breadcrumb, 'activeMenu' => \$activeMenu]); 17 } 18}</pre>
2.	Kita buat file pada <code>PWL_POS/resources/views/welcome.blade.php</code>



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

5.	Kita tambahkan kode berikut router web.php <code>Route::get(uri: '/', action: [WelcomeController::class, 'index']);</code>
6.	Sekarang kita coba jalankan di browser dengan mengetikkan url



http://localhost/PWL_POS/public

The screenshot shows a web browser window with the URL http://localhost/PWL_POS/public. The page title is "PWL - Starter Code". The main content area displays a welcome message: "Selamat Datang" and "Hallo, apakah!!!". Below this, a note says "Selamat datang semua, ini adalah halaman utama dari aplikasi ini.". On the left, there is a sidebar menu with the following items: Data Pengguna (Level User, Data User), Data Barang (Kategori Barang, Data Barang), Data Transaksi (Stok Barang, Transaksi Penjualan). At the bottom of the page, it says "Copyright © 2014-2021 AdminLTE.io, All rights reserved." and "Version 3.2.0".

7. Jangan lupa commit dan push ke github PWL_POS kalian

Praktikum Ke-3 | Implementasi jQuery Datatable di AdminLTE

Langkah	Jawaban/Deskripsi
1.	Kita modifikasi proses CRUD pada tabel <code>m_user</code> pada praktikum ini
2.	Kita gunakan library Yajra-datable dengan mengetikkan perintah pada CMD <code>composer require yajra/laravel-datatables:^10.0</code> atau <code>composer require yajra/laravel-datatables-oracle</code> <pre>C:\laragon\www>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 created Running composer update yajra/laravel-datatables Loading composer repositories with package information Updating dependencies Lock file operations: 62 installs, 0 updates, 0 removals - Locking brick/math (0.12.1) - Locking carbonphp/carbon-doctrine-types (2.1.0) - Locking doctrine/inflector (2.0.10) - Locking doctrine/lexer (3.0.1) - Locking egulias/email-validator (4.0.2) - Locking fruitcake/php-coros (v1.3.0) - Locking guzzlehttp/uri-template (v1.0.3) - Locking illuminate/bus (v10.48.22) - Locking illuminate/collections (v10.48.22) - Locking illuminate/conditionable (v10.48.22) - Locking illuminate/console (v10.48.22) - Locking illuminate/container (v10.48.22) - Locking illuminate/contracts (v10.48.22) - Locking illuminate/database (v10.48.22) - Locking illuminate/events (v10.48.22) - Locking illuminate/filesystem (v10.48.22) - Locking illuminate/http (v10.48.22) - Locking illuminate/macroable (v10.48.22) - Locking illuminate/pipeline (v10.48.22) - Locking illuminate/session (v10.48.22)</pre>



3.	Kita modifikasi route <code>web.php</code> untuk proses CRUD user
	<pre>45 46 // index 47 Route::get(uri: '/', action: [WelcomeController::class, 'index']); 48 49 Route::group(attributes: ['prefix' => 'user'], routes: function(): void{ 50 Route::get(uri: '/', action: [UserController::class, 'index']); 51 Route::post(uri: '/list', action: [UserController::class, 'list']); 52 Route::get(uri: '/create', action: [UserController::class, 'create']); 53 Route::post(uri: '/', action: [UserController::class, 'store']); 54 Route::get(uri: '/{id}', action: [UserController::class, 'show']); 55 Route::get(uri: '/{id}/edit', action: [UserController::class, 'edit']); 56 Route::put(uri: '/{id}', action: [UserController::class, 'update']); 57 Route::delete(uri: '/{id}', action: [UserController::class, 'destroy']) 58});</pre>
4.	Kita buat atau modifikasi penuh untuk <code>UserController.php</code> . Kita buat fungsi <code>index()</code> untuk menampilkan halaman awal user
	<pre>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('user.index', ['breadcrumb' => \$breadcrumb, 'page', 'activeMenu' => \$activeMenu]); }</pre>
5.	Lalu kita buat view pada <code>PWL/resources/views/user/index.blade.php</code>



```
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             <table class="table table-bordered table-striped table-hover table-sm" id="table_user">
13                 <thead>
14                     <tr><th>ID</th><th>Username</th><th>>Nama</th><th>Level Pengguna</th><th>Aksi</th><th></th>
15                 </thead>
16                 </table>
17             </div>
18         </div>
19     @endsection
20
21     @push('css')
22     @endpush
23
24     @push('js')
25     <script>
26         $(document).ready(function(){
27             var dataUser = $('#table_user').DataTable({
28                 // serverSide: true, jika ingin menggunakan server side processing
29                 serverSide: true,
30                 ajax:{
31                     "url": "{{ url('user/list') }}",
32                     "dataType" : "json",
33                     "type": "POST"
34                 },
35                 columns: [
36                     {
37                         // nomor urut dari Laravel datatable addIndexColumn()
38                         data: "DT_RowIndex",
39                         className: "text-center",
40                         orderable: false,
41                         searchable: false
42                     },{
43                         data: "username",
44                         className: "",
45                         // orderable: true, jika ingin kolom ini bisa diurutkan
46                         orderable: true,
47                         // searchable: true, jika ingin kolom ini bisa dicari
48                         searchable: true
49                     },{
50                         data: "nama",
51                         className: "",
52                         orderable: true,
53                         searchable: true
54                     },{
55                         // mengambil data Level hasil dari ORM berelasi
56                         data: "level.level_nama",
57                         className: "",
58                         orderable: false,
59                         searchable: false
60                     }-{
61                         data: "aksi",
62                         className: false,
63                         searchhhable: false
64                     }
65                 ]
66             });
67         });
68     </script>
69     @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`

```
7   <meta name="csrf-token" content="{{ csrf_token() }}> <!--Untuk mengirimkan token Laravel CSRF pada setiap request ajax-->
8
9
10  <!-- Google Font: Source Sans Pro -->
11  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">
12
13  <!-- Font Awesome -->
14  <link rel="stylesheet" href="{{ asset('adminlte/plugins/fontawesome-free/css/all.min.css') }}>
15
16  <!--DataTables-->
17  <link rel="stylesheet" href="{{ asset('adminlte/plugins/datatables-bs4/css/dataTables.bootstrap4.min.css') }}>
18  <link rel="stylesheet" href="{{ asset('adminlte/plugins/datatables-responsive/css/responsive.bootstrap4.min.css') }}>
19  <link rel="stylesheet" href="{{ asset('adminlte/plugins/datatables-buttons/css/buttons.bootstrap4.min.css') }}>
20
21  <!-- Theme style -->
22  <link rel="stylesheet" href="{{ asset('adminlte/dist/css/adminlte.min.css') }}>
23  @stack('css') <!--Digunakan untuk memanggil custom css dari perintah push('css') pada masing-masing view-->
24
25
26  <!-- jQuery -->
27  <script src="{{ asset('adminlte/plugins/jquery/jquery.min.js') }}></script>
28  <!-- Bootstrap 4 -->
29  <script src="{{ asset('adminlte/plugins/bootstrap/js/bootstrap.bundle.min.js') }}></script>
30
31  <!-- DataTables & Plugins -->
32  <script src="{{ asset('adminlte/plugins/datatables/jquery.dataTables.min.js') }}></script>
33  <script src="{{ asset('adminlte/plugins/datatables-bs4/js/dataTables.bootstrap4.min.js') }}></script>
34  <script src="{{ asset('adminlte/plugins/datatables-responsive/js/dataTables.responsive.min.js') }}></script>
35  <script src="{{ asset('adminlte/plugins/datatables-responsive/js/responsive.bootstrap4.min.js') }}></script>
36  <script src="{{ asset('adminlte/plugins/datatables-buttons/js/dataTables.buttons.min.js') }}></script>
37  <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.bootstrap4.min.js') }}></script>
38  <script src="{{ asset('adminlte/plugins/jsszip/jsszip.min.js') }}></script>
39  <script src="{{ asset('adminlte/plugins/pdfmake/pdfmake.min.js') }}></script>
40  <script src="{{ asset('adminlte/plugins/pdfmake/vfs_fonts.js') }}></script>
41  <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.html5.min.js') }}></script>
42  <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.print.min.js') }}></script>
43  <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.colvis.min.js') }}></script>
44
45  <!-- AdminLTE App -->
46  <script src="{{ asset('adminlte/dist/js/adminlte.min.js') }}></script>
47
48  <script>
49    // Untuk mengirimkan token Laravel CSRF pada setiap request ajax
50    $.ajaxSetup({headers: {'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')}});
51  </script>
52  @stack('js') <!--Digunakan untuk panggil custom dari perintah push('js') pada masing-masing view-->
53
54
55  </body>
56
57  </html>
```

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

```
// Ambil data user dalam bentuk json untuk datatables
1 reference | 0 overrides
public function list(Request $request): JsonResponse {
    $users = UserModel::select(columns: 'user_id', 'username', 'nama', 'level_id')->with(relations: 'level');

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

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



ID	Username	Nama	Level Pengguna	Aksi
1	admin	Administrator	Administrator	[Detail] [Edit] [Hapus]
2	manager	Manager	Manager	[Detail] [Edit] [Hapus]
3	staff	Staff/Kasir	Staff/Kasir	[Detail] [Edit] [Hapus]
4	customer_1	Pelanggan pertama	Pelanggan	[Detail] [Edit] [Hapus]
5	manager_dua	Manager2	Manager	[Detail] [Edit] [Hapus]
6	manager_tiga	Manager 3	Manager	[Detail] [Edit] [Hapus]
7	manager22	Manager Dua Dua	Manager	[Detail] [Edit] [Hapus]
8	manager33	Manager Tiga Tiga	Manager	[Detail] [Edit] [Hapus]
9	manager56	Manager55	Manager	[Detail] [Edit] [Hapus]
10	manager12	Manager11	Manager	[Detail] [Edit] [Hapus]

Saat klik menu data user maka akan muncul list data user yang sebelumnya telah tersimpan di database.

9. Selanjutnya kita modifikasi [UserController.php](#) untuk form tambah data user

```
// Menampilkan halaman form tambah user
1 reference | 0 overrides
public function create(): Factory|View{
    $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', data: ['breadcrumb' => $breadcrumb, 'page' => $page, 'level' => $level, 'activeMenu' => $activeMenu])
}
```

10. Sekarang kita buat form untuk menambah data, kita buat file [PWL/resources/views/user/create.blade.php](#)



```
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>
26             <div class="form-group row">
27                 <label class="col-1 control-label col-form-label">Username</label>
28                 <div class="col-11">
29                     <input type="text" class="form-control" id="username" name="username" value="{{ old('username') }}" required>
30                     @error('username')
31                         <small class="form-text text-danger">{{ $message }}</small>
32                     @enderror
33                 </div>
34             </div>
35             <div class="form-group row">
36                 <label class="col-1 control-label col-form-label">Nama</label>
37                 <div class="col-11">
38                     <input type="text" class="form-control" id="nama" name="nama" value="{{ old('nama') }}" required>
39                     @error('nama')
40                         <small class="form-text text-danger">{{ $message }}</small>
41                     @enderror
42                 </div>
43             </div>
44             <div class="form-group row">
45                 <label class="col-1 control-label col-form-label">Password</label>
46                 <div class="col-11">
47                     <input type="password" class="form-control" id="password" name="password" required>
48                     @error('password')
49                         <small class="form-text text-danger">{{ $message }}</small>
50                     @enderror
51                 </div>
52             </div>
53             <div class="form-group row">
54                 <label class="col-1 control-label col-form-label"></label>
55                 <div class="col-11">
56                     <button type="submit" class="btn btn-primary btn-sm">Simpan</button>
57                     <a class="btn btn-sm btn-default ml-1" href="{{ url('user') }}>Kembali</a>
58                 </div>
59             </div>
60         </form>
61     </div>
62     </div>
63     @endsection
64     @push('css')
65     @endpush
66     @push('js')
67     @endpush
68 
```

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



```
// Menyimpan data user baru
1 reference | 0 overrides
public function store(Request $request): mixed|RedirectResponse{
    $request->validate([
        // username harus diisi, berupa string, minimal 3 karakter, dan bernilai unik di tabel m_user kolom username
        'username' => 'required|string|min:3|unique:m_user,username',
        'nama'      => 'required|string|max:100', // nama harus diisi, berupa string, dan maksimal 100 karakter
        'password'  => 'required|min:5',
        'level_id'   => 'required|integer'
    ]);

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

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

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

Tambah User

Tambah user baru

Level: Pelanggan

Username: ysntfarr

Nama: Fardiyani Afro'ul Yasinta

Password: *****

Simpan Kembali

Daftar User

Daftar user yang terdaftar dalam sistem

Show 10 entries Search: _____

ID	Username	Nama	Level Pengguna	Aksi
11	ysntfarr	Fardiyani Afro'ul Yasinta	Pelanggan	Detail Edit Hapus

Showing 11 to 11 of 11 entries

Previous 1 2 Next



13.	Selanjutnya, kita masuk pada bagian menampilkan detail data user (klik tombol) pada halaman user. Route yang bertugas untuk menangkap request detail adalah
14.	Jadi kita buat/modifikasi fungsi show() pada UserController.php seperti berikut
15	Kemudian kita buat <i>view</i> di PWL/resources/views/user/show.blade.php

```
// Menampilkan detail user
1 reference | 0 overrides
public function show(string $id): Factory|View{
    $user = UserModel::with('relations: level')->find(id: $id); // Mengambil data user beserta relasi Level
    $breadcrumb = (object) [
        'title' => 'Detail User',
        'list' => ['Home', 'User', 'Detail']
    ];
    $page = (object) [
        'title' => 'Detail User'
    ];
    $activeMenu = 'user'; // Set menu yang sedang aktif

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

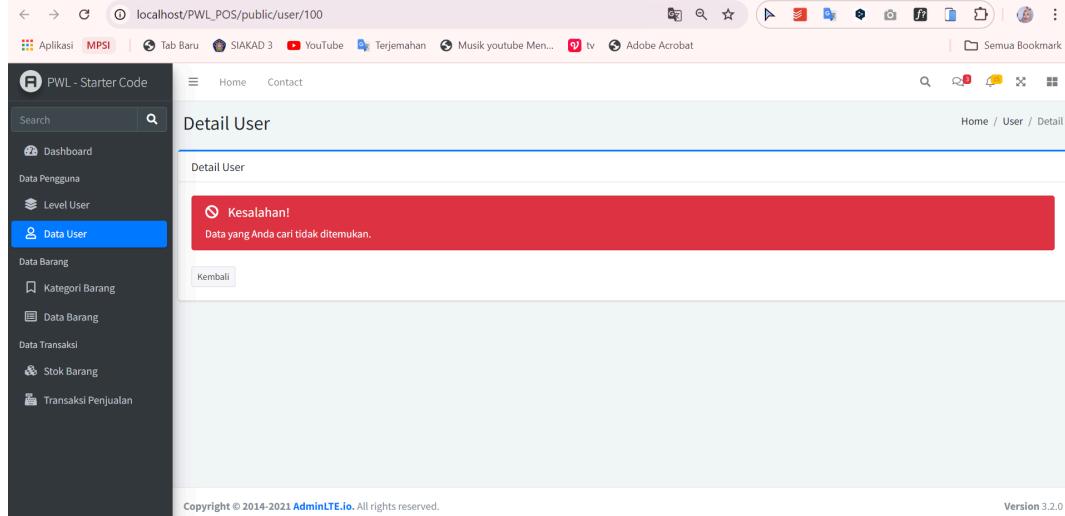


```
1  @extends('layouts.template')
2  @section('content')
3  <div class="card card-outline card-primary">
4      <div class="card-header">
5          <h3 class="card-title">{{ $page->title }}</h3>
6          <div class="card-tools"></div>
7      </div>
8      <div class="card-body">
9          @empty($user)
10             <div class="alert alert-danger alert-dismissible">
11                 <i class="icon fas fa-ban"></i> Kesalahan!
12                 Data yang Anda cari tidak ditemukan.
13             </div>
14         @else
15             <table class="table table-bordered table-striped table-hover table-sm">
16                 <tr>
17                     <th>ID</th>
18                     <td>{{ $user->user_id }}</td>
19                 </tr>
20                 <tr>
21                     <th>Level</th>
22                     <td>{{ $user->level->level_nama }}</td>
23                 </tr>
24                 <tr>
25                     <th>Username</th>
26                     <td>{{ $user->username }}</td>
27                 </tr>
28                 <tr>
29                     <th>Nama</th>
30                     <td>{{ $user->nama }}</td>
31                 </tr>
32                 <tr>
33                     <th>Password</th>
34                     <td>*****</td>
35                 </tr>
36             </table>
37         @endempty
38         <a href="{{ url('user') }}" class="btn btn-sm btn-default mt-2">Kembali</a>
39     </div>
40   </div>
41   @endsection
42
43   @push('css')
44   @endpush
45
46   @push('js')
47   @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!!! |
|--|-----|---|

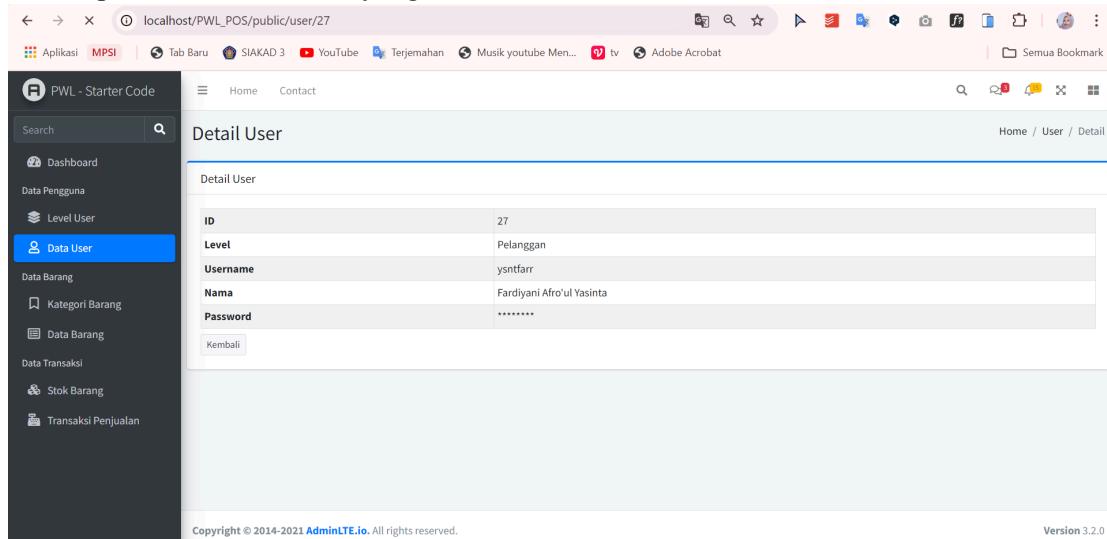


Menampilkan error, karena data yang dicari tidak ada di database.



A screenshot of a web application interface. The title bar shows the URL as localhost/PWL_POS/public/user/100. The page header includes a logo for 'PWL - Starter Code' and navigation links for Home and Contact. On the left, there is a sidebar menu with several items: Dashboard, Data Pengguna, Level User, Data User (which is highlighted in blue), Data Barang, Kategori Barang, Data Barang, Data Transaksi, Stok Barang, and Transaksi Penjualan. The main content area is titled 'Detail User' and contains a red error box with the text 'Kesalahan! Data yang Anda cari tidak ditemukan.' (Error! The data you searched for was not found.). Below the error box is a 'Kembali' (Back) button. At the bottom of the page, there is a copyright notice: 'Copyright © 2014-2023 AdminLTE.io. All rights reserved.' and a 'Version 3.2.0' link.

Menampilkan detail list data yang tersedia di database.



A screenshot of a web application interface, similar to the one above. The title bar shows the URL as localhost/PWL_POS/public/user/27. The page header includes a logo for 'PWL - Starter Code' and navigation links for Home and Contact. On the left, there is a sidebar menu with several items: Dashboard, Data Pengguna, Level User, Data User (which is highlighted in blue), Data Barang, Kategori Barang, Data Barang, Data Transaksi, Stok Barang, and Transaksi Penjualan. The main content area is titled 'Detail User' and displays a table of user information. The table has columns for ID, Level, Username, Nama, and Password. The data shown is: ID: 27, Level: Pelanggan, Username: ysntfarr, Nama: Fardiyani Afro'ul Yasinta, and Password: (redacted). Below the table is a 'Kembali' (Back) button. At the bottom of the page, there is a copyright notice: 'Copyright © 2014-2021 AdminLTE.io. All rights reserved.' and a 'Version 3.2.0' link.

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

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



```
1 // menampilkan halaman form edit user
2 public function edit(string $id)
3 {
4     $user = UserModel::find($id);
5     $level = LevelModel::all();
6     $breadcrumb = (object) [
7         'title' => 'Edit User',
8         'list' => ['Home', 'User', 'Edit']
9     ];
10    $page = (object) [
11        'title' => 'Edit user'
12    ];
13    $activeMenu = 'user'; // set menu yang sedang aktif
14
15    return view('user.edit', ['breadcrumb' => $breadcrumb, 'page' => $page, 'user' => $user,
16    'level' => $level, 'activeMenu' => $activeMenu]);
17 }
18
19 // Menyimpan perubahan data user
20 public function update(Request $request, string $id)
21 {
22     $request->validate([
23         // Username harus diisi, berupa string, minimal 3 karakter,
24         // dan bernilai unik di tabel user kecuali untuk user yang sedang diedit
25         'username' => 'required|string|min:3|unique:m_user,username,' . $id . ',user_id',
26         'nama' => 'required|string|max:100', // Nama harus diisi, berupa string, dan maksimal 100 karakter
27         'password' => 'nullable|min:5', // Password bisa diisi (minimal 5 karakter) dan bisa tidak diisi
28         'level_id' => 'required|integer' // Level ID harus diisi dan berupa angka
29     ]);
30
31     $user = UserModel::find($id);
32     $user->update(['username' => $request->username, 'nama' => $request->nama,
33     'password' => $request->password ? bcrypt($request->password) : $user->password, 'level_id' => $request->level_id]);
34
35     return redirect('/user')->with('success', 'Data user berhasil diubah');
36 }
37 }
```

19. Selanjutnya, kita buat *view* untuk melakukan proses edit data user di [PWL/resources/views/user/edit.blade.php](#)



```
1 @extends('layouts.template')
2 @section('content')
3 <div class="card card-outline card-primary">
4     <div class="card-header">
5         <h3 class="card-title">{{ $page->title }}</h3>
6         <div class="card-tools"></div>
7     </div>
8     <div class="card-body">
9         @empty($user)
10            <div class="alert alert-danger alert-dismissible">
11                <h5><i class="icon fas fa-ban"></i> Kesalahan!</h5>
12                Data yang Anda cari tidak ditemukan.
13            </div>
14            <a href="{{ url('user') }}" class="btn btn-sm btn-default mt-2">Kembali</a>
15        @else
16            <Form method="POST" action="{{ url('/user/' . $user->user_id) }}" class="form-horizontal">
17                @csrf
18                {{!! method_field('PUT') !!}} <!-- Tambahkan method PUT untuk proses edit -->
19
20                <div class="form-group row">
21                    <label class="col-1 control-label col-form-label">Level</label>
22                    <div class="col-11">
23                        <select class="form-control" id="level_id" name="level_id" required>
24                            <option value=""> Pilih Level </option>
25                            @foreach($level as $item)
26                                <option value="{{ $item->level_id }}>
27                                    @if($item->level_id == $user->level_id) selected @endif
28                                    {{ $item->level_name }}</option>
29                            @endforeach
30                        </select>
31                        @error('level_id')
32                            <small class="form-text text-danger">{{ $message }}</small>
33                        @enderror
34                    </div>
35                </div>
36
37                <div class="form-group row">
38                    <label class="col-1 control-label col-form-label">Username</label>
39                    <div class="col-11">
40                        <input type="text" class="form-control" id="username" name="username"
41                            value="{{ old('username', $user->username) }}" required>
42                        @error('username')
43                            <small class="form-text text-danger">{{ $message }}</small>
44                        @enderror
45                    </div>
46                </div>
47
48                <div class="form-group row">
49                    <label class="col-1 control-label col-form-label">Nama</label>
50                    <div class="col-11">
51                        <input type="text" class="form-control" id="nama" name="nama"
52                            value="{{ old('nama', $user->name) }}" required>
53                        @error('nama')
54                            <small class="form-text text-danger">{{ $message }}</small>
55                        @enderror
56                    </div>
57                </div>
58
59                <div class="form-group row">
60                    <label class="col-1 control-label col-form-label">Password</label>
61                    <div class="col-11">
62                        <input type="password" class="form-control" id="password" name="password">
63                        @error('password')
64                            <small class="form-text text-danger">{{ $message }}</small>
65                        @else
66                            <small class="form-text text-muted">Abalkan (jangan diisi) jika tidak ingin mengganti password user.</small>
67                        @enderror
68                    </div>
69                </div>
70
71                <div class="form-group row">
72                    <div class="col-11 offset-1">
73                        <button type="submit" class="btn btn-primary btn-sm">Simpan</button>
74                        <a class="btn btn-sm btn-default ml-1" href="{{ url('user') }}">Kembali</a>
75                    </div>
76                </div>
77            </Form>
78            @empty
79        </div>
80    @endsection
81
82    @push('css')
83    @endpush
84
85    @push('js')
86    @endpush
87
```



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

Tampilan diatas merupakan halaman edit user, dapat mengubah password user.

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
1 reference | 0 overrides
public function destroy(string $id): mixed|RedirectResponse{
    $check = UserModel::find(id: $id); // untuk mengecek apakah data user dengan id yang dimaksud ada atau tidak
    if (!$check) {
        return redirect(to: '/user')->with(key: 'error', value: "Data user tidak ditemukan");
    }

    try {
        UserModel::destroy(ids: $id); // Hapus data user
        return redirect(to: '/user')->with(key: 'success', value: "Data user berhasil dihapus");
    } catch (QueryException $e) { // gunakan QueryException yang sudah diimpor
        // Jika terjadi error ketika menghapus data, redirect kembali ke halaman dengan membawa pesan error
        return redirect(to: '/user')->with(key: 'error', value: "Data user gagal dihapus karena masih terdapat tabel lain yang terkait dengan data ini");
    }
};
```

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

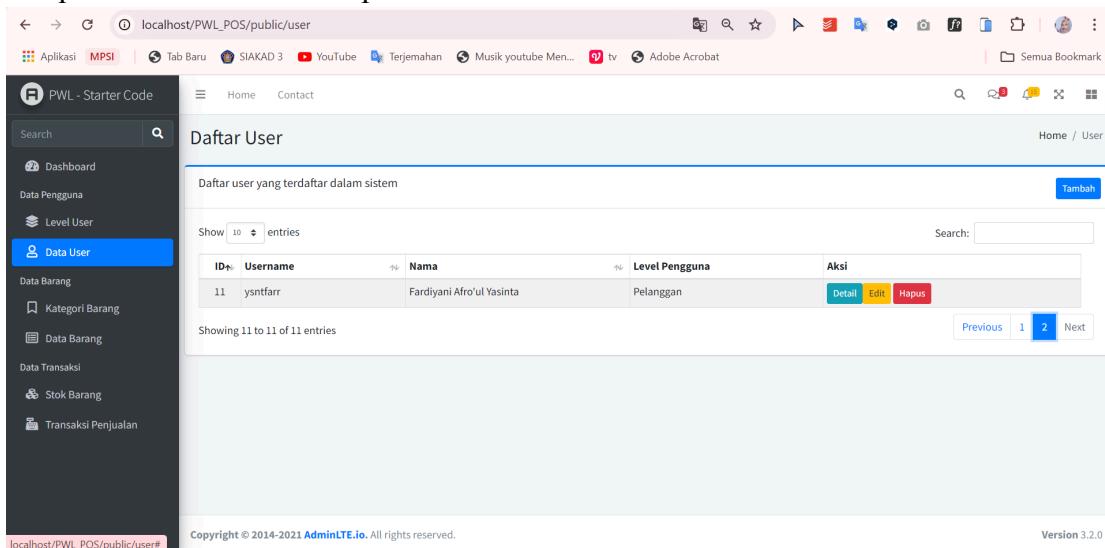


```
1  @extends('layouts.template')
2
3  @section('content')
4      <div class="card card-outline card-primary">
5          <div class="card-header">
6              <h3 class="card-title">{{ $page->title }}</h3>
7              <div class="card-tools">
8                  <a class="btn btn-sm btn-primary 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             @if (session('error'))
16                 <div class="alert alert-danger">{{ session('error') }}</div>
17             @endif
18             <table class="table table-bordered table-striped table-hover table-sm" id="table_user">
19                 <thead>
20                     <tr><th>ID</th><th>Username</th><th>>Nama</th><th>Level Pengguna</th><th>Aksi</th></tr>
21                 </thead>
22                 <tbody>
23                 </tbody>
24             </table>
25         </div>
26     @endsection
27
28     @push('css')
29     @endpush
30
31     @push('js')
32         <script>
33             $(document).ready(function() {
34                 var dataUser = $('#table_user').DataTable({
35                     // serverSide: true, jika ingin menggunakan server side processing
36                     serverSide: true,
37                     ajax: {
38                         "url": "{{ url('user/list') }}",
39                         "dataType": "json",
40                         "type": "POST"
41                     },
42                     columns: [
43                         {
44                             // nomor urut dari laravel datatable addIndexColumn()
45                             data: "DT_RowIndex",
46                             className: "text-center",
47                             orderable: false,
48                             searchable: false
49                         },
50                         {
51                             data: "username",
52                             className: "",
53                             // orderable: true, jika ingin kolom ini bisa diurutkan
54                             orderable: true,
55                             // searchable: true, jika ingin kolom ini bisa dicari
56                             searchable: true
57                         },
58                         {
59                             data: "nama",
60                             className: "",
61                             orderable: true,
62                             searchable: true
63                         },
64                         {
65                             // mengambil data Level hasil dari ORM berelasi
66                             data: "level.level_nama",
67                             className: "",
68                             orderable: false,
69                             searchable: false
70                         },
71                         {
72                             data: "aksi",
73                             className: "",
74                             orderable: false,
75                             searchable: false
76                         }
77                     ]
78                 });
79             });
80         </script>
81     @endpush
82 
```



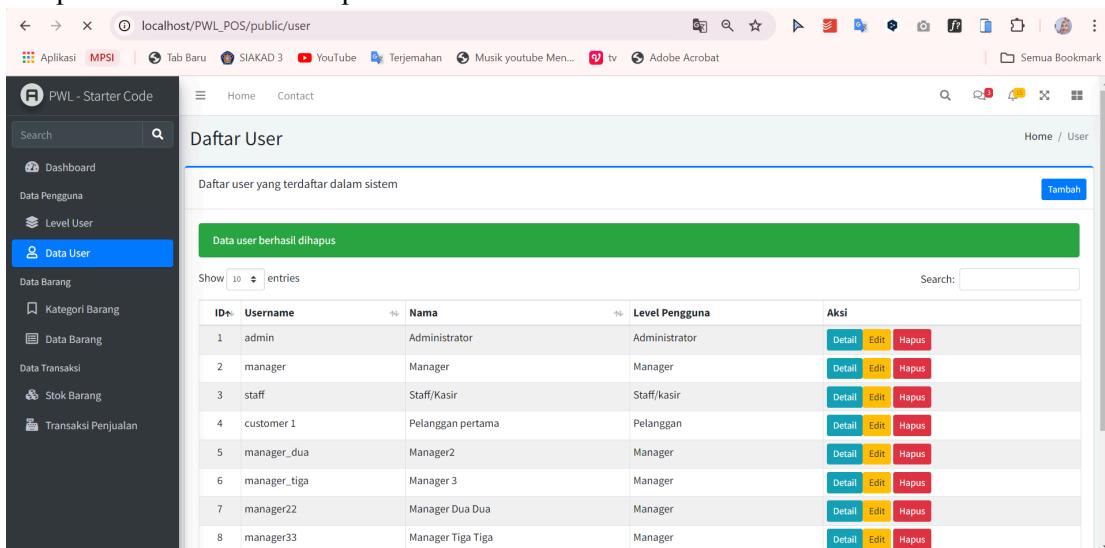
24. Kemudian jalankan browser untuk menghapus salah satu data user. Amati dan laporan!

Tampilan data sebelum dihapus



The screenshot shows a web application interface for managing users. On the left is a sidebar with various menu items: Dashboard, Data Pengguna, Level User, Data User (which is selected and highlighted in blue), Data Barang, Kategori Barang, Data Barang, Data Transaksi, Stok Barang, and Transaksi Penjualan. The main content area is titled 'Daftar User' and displays a table of users. The table has columns for ID, Username, Name, Level Pengguna, and Aksi (Actions). There is one entry: ID 11, Username ysntfarr, Name Fardiyani Afro'ul Yasinta, Level Pengguna Pelanggan. Below the table, it says 'Showing 11 to 11 of 11 entries'. At the bottom right of the table, there are buttons for 'Detail', 'Edit', and 'Hapus' (Delete). The footer of the page includes copyright information: 'Copyright © 2014-2021 AdminLTE.io. All rights reserved.' and 'Version 3.2.0'.

Tampilan data sesudah dihapus



This screenshot shows the same 'Daftar User' page after a user has been deleted. The table now lists eight users. The first user, with ID 1, Username admin, and Name Administrator, is highlighted in green with the message 'Data user berhasil dihapus' (Data user successfully deleted) above the table. The other seven users listed are: ID 2, manager; ID 3, staff; ID 4, customer 1; ID 5, manager_dua; ID 6, manager_tiga; ID 7, manager22; and ID 8, manager33. Each user row has 'Detail', 'Edit', and 'Hapus' buttons.

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 Ke-4 | Implementasi Filtering Datatables

Langkah	Jawaban/Deskripsi
1.	<p>Kita modifikasi fungsi index() di UserController.php untuk menambahkan data yang ingin dijadikan kategori untuk data filtering</p> <pre>2 References to Overrides public function index(): Factory View{ \$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(view: 'user.index', data: ['breadcrumb' => \$breadcrumb, 'page' => \$page, 'level' => \$level, 'activeMenu' => \$activeMenu]); }</pre>
2.	<p>Kemudian kita modifikasi view untuk menampilkan data filtering pada PWL/resources/views/user/index.blade.php</p> <pre>@endif <div class="row"> <div class="col-md-12"> <div class="form-group row"> <label class="col-1 control-label col-form-label>Filter: </label> <div class="col-3"> <select class="form-control" id="level_id" name="level_id" required> <option value="">- Semua -</option> @foreach(\$level as \$item) <option value="{{ \$item->level_id }}>{{ \$item->level_nama }}</option> @endforeach </select> <small class="form-text text-muted">Level Pengguna</small> </div> </div> </div> <table class="table table-bordered table-striped table-hover table-sm" id="table_user"> <thead></div></div></div></pre>
3.	<p>Selanjutnya, tetap pada view index.blade.php, kita tambahkan kode berikut pada deklarasi ajax di datatable. Kode ini digunakan untuk mengirimkan data untuk filtering</p>



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

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

```
// Ambil data user dalam bentuk json untuk datatables
1 reference | 0 overrides
public function list(Request $request): JsonResponse {
    $users = UserModel::select(columns: 'user_id', 'username', 'nama', 'level_id')
        ->with(relations: 'level');

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

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



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

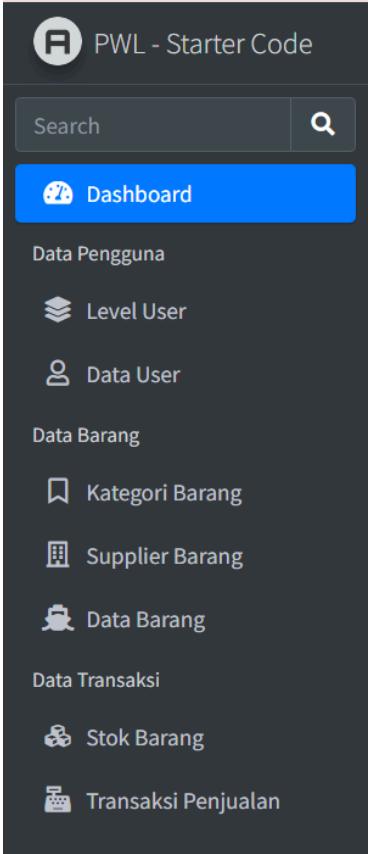
ID	Username	Nama	Level Pengguna	Aksi
1	admin	Administrator	Administrator	[Detail] [Edit] [Hapus]
2	manager	Manager	Manager	[Detail] [Edit] [Hapus]
3	staff	Staff/Kasir	Staff/kasir	[Detail] [Edit] [Hapus]
4	customer_1	Pelanggan pertama	Pelanggan	[Detail] [Edit] [Hapus]
5	manager_dua	Manager2	Manager	[Detail] [Edit] [Hapus]
6	manager_tiga	Manager 3	Manager	[Detail] [Edit] [Hapus]
7	manager22	Manager Dua Dua	Manager	[Detail] [Edit] [Hapus]
8	manager33	Manager Tiga Tiga	Manager	[Detail] [Edit] [Hapus]

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

8. **Jangan lupa commit dan push ke github PWL_POS kalian**



Tugas

Langkah	Jawaban/Deskripsi
1	<p>Implementasikan web layout dan datatables, pada menu berikut ini</p> <p>✓ Tabel m_level ✓ Tabel m_kategori ✓ Tabel m_supplier ✓ Tabel m_barang</p> 



1. Tampilan Level User

The screenshot shows a table titled 'Daftar Level' listing four user levels:

ID	Level Kode	Level Name	Aksi
1	ADM	Administrator	[Detail] [Edit] [Hapus]
2	MNG	Manager	[Detail] [Edit] [Hapus]
3	STF	Staff/kasir	[Detail] [Edit] [Hapus]
4	PLG	Pelanggan	[Detail] [Edit] [Hapus]

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

- Tambah Level User

The screenshot shows a form titled 'Tambah Level' with fields for 'Level kode' (MHS) and 'Level name' (Mahasiswa). There are 'Simpan' and 'Kembali' buttons at the bottom.

Data level tersimpan

The screenshot shows the 'Daftar Level' table with five entries, including the newly added 'MHS' level:

ID	Level Kode	Level Name	Aksi
1	ADM	Administrator	[Detail] [Edit] [Hapus]
2	MNG	Manager	[Detail] [Edit] [Hapus]
3	STF	Staff/kasir	[Detail] [Edit] [Hapus]
4	PLG	Pelanggan	[Detail] [Edit] [Hapus]
5	MHS	Mahasiswa	[Detail] [Edit] [Hapus]

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

- Detail Level User



The screenshot shows a web application interface for managing user levels. On the left is a sidebar with navigation links: Dashboard, Data Pengguna, Level User (which is selected and highlighted in blue), Data Barang, Data Transaksi, Stok Barang, and Transaksi Penjualan. The main content area has a title 'Detail Level'. Below it is a table with one row containing the following data:

ID	6
Level	MHS
Username	Mahasiswa
Kembali	

At the bottom of the page, there is a copyright notice 'Copyright © 2014-2021 AdminLTE.io. All rights reserved.' and a version number 'Version 3.2.0'.

- Edit Level User

This screenshot shows the 'Edit Level' page. The sidebar and layout are identical to the previous 'Detail Level' page. The main content area has a title 'Edit Level'. Below it is a form with two fields: 'Level Kode' (MHS) and 'Level Name' (Mahasiswa). At the bottom of the form are two buttons: 'Simpan' (Save) and 'Kembali' (Back).

- Hapus Level User

This screenshot shows the 'Daftar Level' (List of Levels) page. The sidebar and layout are consistent. A modal dialog box is open in the center, asking 'localhost menyatakan Apakah Anda yakin menghapus data ini?' with 'Oke' (OK) and 'Batal' (Cancel) buttons. Below the dialog is a table titled 'Daftar level yang terdapat dalam sistem'. The table has columns: ID, Level Kode, Level Name, and Aksi (Actions). The data is as follows:

ID	Level Kode	Level Name	Aksi
1	ADM	Administrator	[Detail] [Edit] [Hapus]
2	MNG	Manager	[Detail] [Edit] [Hapus]
3	STF	Staff/kasir	[Detail] [Edit] [Hapus]
4	PLG	Pelanggan	[Detail] [Edit] [Hapus]
5	MHS	Mahasiswa	[Detail] [Edit] [Hapus]

At the bottom of the page, there is a copyright notice 'Copyright © 2014-2021 AdminLTE.io. All rights reserved.' and a version number 'Version 3.2.0'.

Data terhapus



The screenshot shows a table titled 'Daftar Level' with the following data:

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

A green success message at the top right says 'Data level berhasil dihapus'.

2. Tampilan Kategori Barang

The screenshot shows a table titled 'Daftar Kategori' with the following data:

ID	Kategori Kode	Kategori Nama	Aksi
1	EK	Elektronik	[Detail] [Edit] [Hapus]
2	KC	Kecantikan	[Detail] [Edit] [Hapus]
3	OB	Obat-obatan	[Detail] [Edit] [Hapus]
4	MM	Makanan dan Minuman	[Detail] [Edit] [Hapus]
5	PK	Pakaian	[Detail] [Edit] [Hapus]

A green success message at the top right says 'Data kategori berhasil dihapus'.

- Tambah Kategori Barang

The screenshot shows a form titled 'Tambah Kategori baru' with the following fields:

- Kode Kategori: ITR
- Nama kategori: Interior

At the bottom are 'Simpan' and 'Kembali' buttons.



Data tersimpan

ID	Kategori Kode	Kategori Nama	Aksi
1	EK	Elektronik	[Detail] [Edit] [Hapus]
2	KC	Kecantikan	[Detail] [Edit] [Hapus]
3	OB	Obat-obatan	[Detail] [Edit] [Hapus]
4	MM	Makanan dan Minuman	[Detail] [Edit] [Hapus]
5	PK	Pakaian	[Detail] [Edit] [Hapus]
6	ITR	Interior	[Detail] [Edit] [Hapus]

- Detail Kategori Barang

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

Version 3.2.0

- Edit Kategori Barang

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

Version 3.2.0

- Hapus Kategori Barang



A screenshot of a web application interface. At the top, there's a navigation bar with links like 'Aplikasi', 'MPSI', 'Tab Baru', 'SIAKAD 3', 'YouTube', and 'Semua Bookmark'. Below the navigation is a sub-navigation menu with items such as 'Dashboard', 'Data Pengguna', 'Level User', 'Data User', 'Data Barang', 'Kategori Barang', 'Supplier Barang', 'Data Barang', 'Data Transaksi', 'Stok Barang', and 'Transaksi Penjualan'. The main content area is titled 'Daftar Kategori' and shows a table of categories. A modal dialog box is overlaid on the page, asking 'localhost menyatakan Apakah Anda yakin menghapus data ini?' with 'Oke' and 'Batal' buttons. The table data is as follows:

ID	Kategori Kode	Kategori Nama	Aksi
1	EK	Elektronik	[Detail] [Edit] [Hapus]
2	KC	Kecantikan	[Detail] [Edit] [Hapus]
3	OB	Obat-obatan	[Detail] [Edit] [Hapus]
4	MM	Makanan dan Minuman	[Detail] [Edit] [Hapus]
5	PK	Pakaian	[Detail] [Edit] [Hapus]
6	ITR	Interior	[Detail] [Edit] [Hapus]

At the bottom of the page, there are copyright and version information: 'Copyright © 2014-2021 AdminLTE.io. All rights reserved.' and 'Version 3.2.0'.

Data terhapus

A screenshot of the same web application interface as the previous one. The main content area is titled 'Daftar Kategori' and shows a table of categories. A green success message box is displayed at the top of the table area, stating 'Data Kategori berhasil dihapus'. The table data is as follows:

ID	Kategori Kode	Kategori Nama	Aksi
1	EK	Elektronik	[Detail] [Edit] [Hapus]
2	KC	Kecantikan	[Detail] [Edit] [Hapus]
3	OB	Obat-obatan	[Detail] [Edit] [Hapus]
4	MM	Makanan dan Minuman	[Detail] [Edit] [Hapus]
5	PK	Pakaian	[Detail] [Edit] [Hapus]

At the bottom of the page, there are copyright and version information: 'Copyright © 2014-2021 AdminLTE.io. All rights reserved.' and 'Version 3.2.0'.

3. Tampilan Supplier Barang

A screenshot of the web application interface. The main content area is titled 'Daftar Supplier' and shows a table of suppliers. The table data is as follows:

ID	Kode Supplier	Nama Supplier	Supplier Alamat	Aksi
1	SHB	Sumber Healthy dan Beauty	Jakarta	[Detail] [Edit] [Hapus]
2	SAS	Serba Ada Shop	Malang	[Detail] [Edit] [Hapus]
3	CEK	Chin Elektronik	Surabaya	[Detail] [Edit] [Hapus]

At the bottom of the page, there are copyright and version information: 'Copyright © 2014-2021 AdminLTE.io. All rights reserved.' and 'Version 3.2.0'.

- Tambah Sumpplier Barang



Tambah Supplier

Tambah supplier baru

Kode supplier: FAY

Nama supplier: Fafa Beauty

Alamat supplier: Sidoarjo

Simpan Kembali

Data tersimpan

Daftar Supplier

Daftar supplier yang terdaftar dalam sistem

Data supplier berhasil disimpan

ID	Kode Supplier	Nama Supplier	Supplier Alamat	Aksi
1	SHB	Sumber Healthy dan Beauty	Jakarta	[Detail] [Edit] [Hapus]
2	SAS	Serba Ada Shop	Malang	[Detail] [Edit] [Hapus]
3	CEK	Chin Elektronik	Surabaya	[Detail] [Edit] [Hapus]
4	FAY	Fafa Beauty	Sidoarjo	[Detail] [Edit] [Hapus]

- Detail Supplier Barang

Detail Supplier

Detail supplier

ID	Kode supplier	Nama supplier
4	FAY	Fafa Beauty
		Sidoarjo

- Edit Supplier Barang



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

Version 3.2.0

- Hapus Supplier Barang

localhost menyatakan
Apakah Anda yakin menghapus data ini?

Oke Batal

ID	Kode Supplier	Nama Supplier	Supplier Alamat	Aksi
1	SHB	Sumber Healthy dan Beauty	Jakarta	[Detail] [Edit] [Hapus]
2	SAS	Serba Ada Shop	Malang	[Detail] [Edit] [Hapus]
3	CEK	Chin Elektronik	Surabaya	[Detail] [Edit] [Hapus]
4	FAY	Fafa Beauty	Sidoarjo	[Detail] [Edit] [Hapus]

Showing 1 to 4 of 4 entries

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

Version 3.2.0

Data terhapus

Data supplier berhasil dihapus

ID	Kode Supplier	Nama Supplier	Supplier Alamat	Aksi
1	SHB	Sumber Healthy dan Beauty	Jakarta	[Detail] [Edit] [Hapus]
2	SAS	Serba Ada Shop	Malang	[Detail] [Edit] [Hapus]
3	CEK	Chin Elektronik	Surabaya	[Detail] [Edit] [Hapus]

Showing 1 to 3 of 3 entries

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

Version 3.2.0



4. Tampilan Data Barang

ID	Nama barang	Kode barang	Harga jual	Harga beli	Kategori barang	Aksi
1	Lampu LED Philips 9W	E01	Rp 20.000	Rp 25.000	Elektronik	[Detail] [Edit] [Hapus]
2	Kabel HDMI 1.5 Meter	E02	Rp 20.000	Rp 35.000	Elektronik	[Detail] [Edit] [Hapus]
3	Charger USB Anker 18W	E03	Rp 30.000	Rp 40.000	Elektronik	[Detail] [Edit] [Hapus]
4	Mie Instan Indomie Goreng	M01	Rp 2.500	Rp 3.000	Makanan dan Minuman	[Detail] [Edit] [Hapus]
5	Susu UHT Frisian Flag 1L	M02	Rp 12.000	Rp 15.000	Makanan dan Minuman	[Detail] [Edit] [Hapus]
6	Keripik Singkong Balado	M03	Rp 7.000	Rp 10.000	Makanan dan Minuman	[Detail] [Edit] [Hapus]
7	Kaos Polos Hitam	P01	Rp 25.000	Rp 50.000	Pakaian	[Detail] [Edit] [Hapus]
8	Blouse Wanita Batik	P02	Rp 50.000	Rp 70.000	Pakaian	[Detail] [Edit] [Hapus]

- Tambah Data Barang

Tambah Barang baru

Kode Barang: M04

Nama Barang: Seblak ff

Kategori: Makanan dan Minuman

Harga Beli: 20000

Harga Jual: 25000

[Simpan] [Kembali]

Data tersimpan

Data barang berhasil disimpan

ID	Nama barang	Kode barang	Harga jual	Harga beli	Kategori barang	Aksi
11	Facial Wash Garnier	K02	Rp 15.000	Rp 25.000	Kecantikan	[Detail] [Edit] [Hapus]
12	Body Lotion Vaseline	K03	Rp 250.000	Rp 350.000	Kecantikan	[Detail] [Edit] [Hapus]
13	Paracetamol 500mg	O01	Rp 500	Rp 1.000	Obat-obatan	[Detail] [Edit] [Hapus]
14	Vitamin C 500mg	O02	Rp 2.000	Rp 3.000	Obat-obatan	[Detail] [Edit] [Hapus]
15	Amoxicillin 500mg	O03	Rp 1.000	Rp 2.000	Obat-obatan	[Detail] [Edit] [Hapus]
16	Seblak ff	M04	Rp 20.000	Rp 25.000	Makanan dan Minuman	[Detail] [Edit] [Hapus]



- Detail Data Barang

The screenshot shows a web application interface for managing inventory. On the left is a sidebar with various menu items like Dashboard, Data Pengguna, Level User, Data User, Data Barang, Kategori Barang, Supplier Barang, and Data Barang (which is highlighted in blue). The main content area has a title 'Detail barang'. Below it is a table with the following data:

ID	16
Kode barang	M04
Harga jual	Rp 25.000
Harga beli	Rp 20.000
Kategori	Makanan dan Minuman

At the bottom of the page, there is a 'Kembali' button and a copyright notice: 'Copyright © 2014-2021 AdminLTE.io. All rights reserved.' and 'Version 3.2.0'.

- Edit Data Barang

This screenshot shows the 'Edit data barang' page. The sidebar and overall layout are identical to the previous 'Detail barang' page. The main content area has a title 'Edit data barang' and a form with the following fields:

Kategori	Makanan dan Minuman
Kode barang	M04
Nama barang	Seblak ff
Harga jual	25000
Harga beli	20000

Below the form are 'Simpan' and 'Kembali' buttons. The page includes the same copyright notice and version information as the previous screen.

- Hapus Data Barang

This screenshot shows a list of products with a confirmation dialog box overlaid. The dialog asks 'Apakah Anda yakin menghapus data ini?' with 'Oke' and 'Batal' buttons. The main content area shows a table of products:

ID	Nama barang	Kode barang	Harga jual	Harga beli	Kategori barang	Aksi
11	Facial Wash Garnier	K02	Rp 15.000	Rp 25.000	Kecantikan	[Detail] [Edit] [Hapus]
12	Body Lotion Vaseline	K03	Rp 250.000	Rp 350.000	Kecantikan	[Detail] [Edit] [Hapus]
13	Paracetamol 500mg	O01	Rp 500	Rp 1.000	Obat-obatan	[Detail] [Edit] [Hapus]
14	Vitamin C 500mg	O02	Rp 2.000	Rp 3.000	Obat-obatan	[Detail] [Edit] [Hapus]
15	Amoxicillin 500mg	O03	Rp 1.000	Rp 2.000	Obat-obatan	[Detail] [Edit] [Hapus]
16	Seblak ff	M04	Rp 20.000	Rp 25.000	Makanan dan Minuman	[Detail] [Edit] [Hapus]

At the bottom, there is a message 'Showing 11 to 16 of 16 entries' and a navigation bar with 'Previous' and 'Next' buttons. The page includes the same copyright notice and version information as the previous screens.

Data terhapus



The screenshot shows a web application interface for managing inventory. On the left, there is a sidebar menu with options like Dashboard, Data Pengguna, Level User, Data User, Data Barang, Kategori Barang, Supplier Barang, and Data Barang (which is highlighted). The main content area displays a table titled "Daftar Barang yang terdaftar dalam sistem". The table has columns for ID, Nama barang, Kode barang, Harga jual, Harga beli, Kategori barang, and Aksi (Actions). The table contains 15 entries. At the top of the main content area, there is a message: "Data barang berhasil dihapus" (Item data successfully deleted). Below the table, there are buttons for "Previous", "1", "2", and "Next". At the bottom of the page, there is a copyright notice: "Copyright © 2014-2021 AdminLTE.io. All rights reserved." and "Version 3.2.0".

ID	Nama barang	Kode barang	Harga jual	Harga beli	Kategori barang	Aksi
11	Facial Wash Garnier	K02	Rp 15.000	Rp 25.000	Kecantikan	[Detail] [Edit] [Hapus]
12	Body Lotion Vaseline	K03	Rp 250.000	Rp 350.000	Kecantikan	[Detail] [Edit] [Hapus]
13	Paracetamol 500mg	O01	Rp 500	Rp 1.000	Obat-obatan	[Detail] [Edit] [Hapus]
14	Vitamin C 500mg	O02	Rp 2.000	Rp 3.000	Obat-obatan	[Detail] [Edit] [Hapus]
15	Amoxicillin 500mg	O03	Rp 1.000	Rp 2.000	Obat-obatan	[Detail] [Edit] [Hapus]