

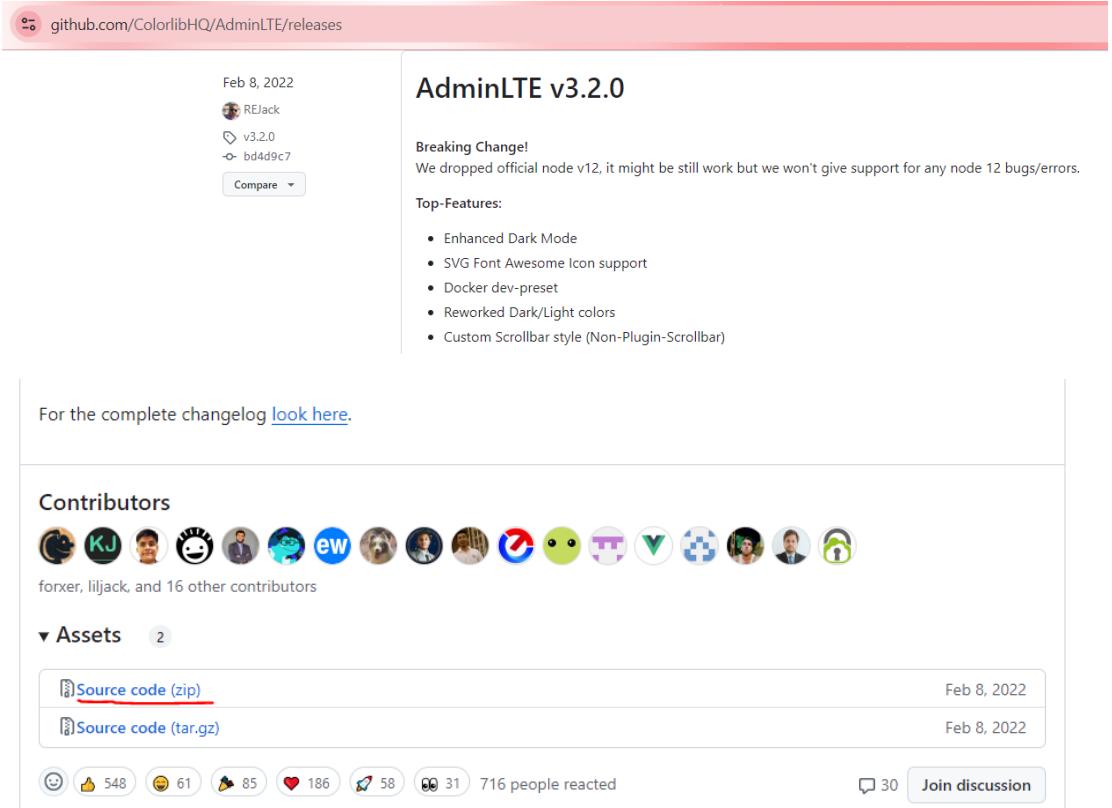


Mata Kuliah : Pemrograman Web Lanjut (PWL)
Program Studi : D4 – Sistem Informasi Bisnis
Semester : 5

Kelas : SIB
NIM : 2241760007
Nama : Anaradi Octa Lavechia
Jobsheet Ke- : 5 (Lima)

Laporan Jobsheet

Praktikum Ke-1 – Layouting AdminLTE

Langkah	Jawaban/Deskripsi
1	<p>Kita download AdminLTE v3.2.0 yang rilis pada 8 Feb 2022</p> 
2	<p>Setelah kita berhasil download, kita ekstrak file yang sudah di download ke folder project PWL_POS/public, kemudian kita <i>rename</i> folder cukup menjadi adminlte</p>



3	<p>Selanjutnya kita buka di browser dengan alamat http://localhost/PWL_POS/public/adminlte maka akan muncul tampilan seperti berikut</p>
4	<p>Kita klik menu Extras > Blank Page, page inilah yang akan menjadi dasar web template</p>



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)



	<pre>EXPLORER ... Welcome template.blade.php U resources > views > layouts > template.blade.php > ... 1 2 <!DOCTYPE html> 3 <html lang="en"> 4 <head> 5 <meta charset="utf-8"> 6 <meta name="viewport" content="width=device-width, initial-scale=1"> 7 <title>AdminLTE 3 Blank Page</title> 8 9 <!-- Google Font: Source Sans Pro --> 10 <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback"> 11 <!-- Font Awesome --> 12 <link rel="stylesheet" href="../../plugins/fontawesome-free/css/all.min.css"> 13 <!-- Theme style --> 14 <link rel="stylesheet" href="../../dist/css/adminlte.min.css"> 15 </head> 16 <body class="hold-transition sidebar-mini"> 17 <!-- Site wrapper --> 18 <div class="wrapper"> 19 <!-- Navbar --> 20 <nav class="main-header navbar navbar-expand navbar-white navbar-light"> 21 <!-- Left navbar links --> 22 <ul class="navbar-nav"> 23 <li class="nav-item"> 24 <i class="fas fa-bars"></i> 25 </pre>
8	File layouts/template.blade.php adalah file utama untuk templating website
9	Pada baris 1-14 file template.blade.php, kita modifikasi
	<pre>resources > views > layouts > template.blade.php > html > head > link 1 <!DOCTYPE html> 2 <html lang="en"> 3 <head> 4 <meta charset="utf-8"> 5 <meta name="viewport" content="width=device-width, initial-scale=1"> 6 <title>{{ config('app.name', 'PWL Laravel Starter Code') }}</title> 7 8 <!-- Google Font: Source Sans Pro --> 9 <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback"> 10 <!-- Font Awesome --> 11 <link rel="stylesheet" href="{{ asset('adminlte/plugins/fontawesome-free/css/all.min.css') }}"> 12 <!-- Theme style --> 13 <link rel="stylesheet" href="{{ asset('adminlte/dist/css/adminlte.min.css') }}"> 14 </head></pre>



	Baris 19 adalah komponen Blade untuk memanggil elemen layouts/header.blade.php agar menjadi satu dengan template.blade.php saat di-render nanti.
11	Kita modifikasi baris 25 dan 26 pada template.blade.php
	<pre>17 <div class="wrapper"> 18 <!-- Navbar --> 19 @include('layouts.header') 20 <!-- /.navbar --> 21 22 <!-- Main Sidebar Container --> 23 <aside class="main-sidebar sidebar-dark-primary elevation-4"> 24 <!-- Brand Logo --> 25 26 27 PWL - Starter Code 28 29 <!-- Sidebar --> 30 <div class="sidebar"></pre>
12	Selanjutnya kita blok baris 31-693 (baris untuk <i>element 2-sidebar</i>), lalu kita cut, dan paste-kan di file PWL_POS/resource/view/layouts/sidebar.blade.php (buat dulu file sidebar.blade.php jika belum ada). Sehingga tampilan dari file template.blade.php menjadi seperti berikut

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

Selanjutnya perhatikan baris 87-98 (baris untuk *element 5-footer*), lalu kita cut, dan paste-kan di file PWL_POS/resource/view/layouts/footer.blade.php (buat file footer.blade.php jika belum ada). Sehingga tampilan dari file template.blade.php menjadi seperti berikut

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



14	Kemudian kita modifikasi file template.blade.php baris 91-100
	<pre>91 <!-- jQuery --> 92 <script src="{{ assets('adminlte/plugins/jquery/jquery.min.js') }}"></script> 93 <!-- Bootstrap 4 --> 94 <script src="{{ assets('adminlte/plugins/bootstrap/js/bootstrap.bundle.min.js') }}"></script> 95 <!-- AdminLTE App --> 96 <script src="{{ assets('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 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
	<pre>30 <!-- Sidebar --> 31 @include('layouts.sidebar') 32 <!-- /.sidebar --> 33 </aside> 34 35 <!-- Content Wrapper. Contains page content --> 36 <div class="content-wrapper"> 37 <!-- Content Header (Page header) --> 38 @include('layouts.breadcrumb') 39 40 <!-- Main content --> 41 <section class="content"> 42 <!-- Default box --> 43 <div class="card"> 44 <div class="card-header"> 45 <h3 class="card-title">Title</h3> 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 template.blade.php. dan kita ganti dengan kode seperti ini @yield('content')
19	Hasil akhir pada file utama layouts/template.blade.php adalah seperti berikut



	<pre>18 <!-- Navbar --> 19 @include('layouts.header') 20 <!-- /.navbar --> 21 22 <!-- Main Sidebar Container --> 23 <aside class="main-sidebar sidebar-dark-primary elevation-4"> 24 <!-- Brand Logo --> 25 26 <img src="{{ assets('adminlte/dist/img/AdminLTELogo.png') }}" alt="AdminLTE Logo" class="brand-image" 27 PWL - Starter Code 28 29 30 <!-- Sidebar --> 31 @include('layouts.sidebar') 32 <!-- /.sidebar --> 33 </aside> 34 35 <!-- Content Wrapper. Contains page content --> 36 <div class="content-wrapper"> 37 <!-- Content Header (Page header) --> 38 @include('layouts.breadcrumb') 39 <!-- Main content --> 40 <section class="content"> 41 @yield('content') 42 </section> 43 <!-- /.content --> 44 </div> 45 <!-- /.content-wrapper --> 46 47 @include('layouts.footer') 48 </div> 49 <!-- /.wrapper --></pre>
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 WelcomeController.php

```
app > Http > Controllers > WelcomeController.php > PHP Intelephense > WelcomeController > index
1  <?php
2
3  namespace App\Http\Controllers;
4
5  0 references | 0 implementations
6  class WelcomeController extends Controller
7  {
8    0 references | 0 overrides
9    public function index(): Factory|View
10   [
11     $breadcrumb = (object) [
12       'title' => 'Selamat Datang',
13       'list' => ['Home', 'Welcome']
14     ];
15
16     $activeMenu = 'dashboard';
17
18     return view(view: 'welcome', data: ['breadcrumb' => $breadcrumb, 'activeMenu' => $activeMenu]);
19   }
20 }
```



2	Kita buat file pada PWL_POS/resources/views/welcome.blade.php	
	<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-title">Halo, apakahbar!!!</h3> 8 <div class="card-tools"></div> 9 </div> 10 <div class="card-body"> 11 <h3>Selamat datang semua, ini adalah halaman utama dari aplikasi ini.</h3> 12 </div> 13 </div> 14 @endsection</pre>	
3	Kita modifikasi file PWL_POS/resources/views/layouts/breadcrumb.blade.php	
	<pre>resources > views > layouts > 🗂 breadcrumb.blade.php > ... 1 <section class="content-header"> 2 <div class="container-fluid"> 3 <div class="row mb-2"> 4 <div class="col-sm-6"><h1>{{ \$breadcrumb-> title }}</h1></div> 5 <div class="col-sm-6"> 6 <ol class="breadcrumb float-sm-right"> 7 <foreach (\$breadcrumb->list as \$key => \$value) 8 <if(\$key == count(\$breadcrumb->list) -1) 9 <li class="breadcrumb-item active">{{ \$value }} 10 <else 11 <li class="breadcrumb-item">{{ \$value }} 12 <endif 13 <endforeach 14 15 </div> 16 </div> 17 </div><!-- /.container-fluid --> 18 </section></pre>	
4	Kita modifikasi file PWL_POS/resources/views/layouts/sidebar.blade.php	
	<pre><div class="sidebar"> <!-- Sidebar user (optional) --> <div class="user-panel mt-3 pb-3 mb-3 d-flex"> <div class="image"></pre>	



```

</div>
<div class="info">
    <a href="#" class="d-block">Alexander Pierce</a>
</div>
</div>

<!-- 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">
        <!-- Add icons to the links using the .nav-icon class
             with font-awesome or any other icon font library -->
        <li class="nav-item">
            <a href="{{ url('/') }}" class="nav-link" {{ ($activeMenu == 'dashboard')? 'active' : '' }}>
                <i class="nav-icon fas fa-tachometer-alt"></i>
                <p>Dashboard</p>
            </a>
        </li>
        <li class="nav-header">Data Pengguna</li>
        <li class="nav-item">
            <a href="{{ url('/level') }}" class="nav-link" {{ ($activeMenu == 'level')? 'active' : '' }}>
                <i class="nav-icon fas fa-layer-group"></i>
                <p>Level User</p>
            </a>
        </li>
        <li class="nav-item">
            <a href="{{ url('/user') }}" class="nav-link" {{ ($activeMenu == 'user')? 'active' : '' }}>
```



```
<i class="nav-icon far fa-user"></i>
<p>Data User</p>
</a>
</li>
<li class="nav-header">Data Barang</li>
<li class="nav-item">
    <a href="{{ url('/kategori') }}" class="nav-link {{ ($activeMenu == 'kategori')? 'active' : '' }}>
        <i class="nav-icon far fa-bookmark"></i>
        <p>Kategori Barang</p>
    </a>
</li>
</ul>
</li>
<li class="nav-item">
    <a href=".{{ url('/barang') }}" class="nav-link {{ ($activeMenu == 'barang')? 'active' : '' }}>
        <i class="nav-icon far fa-list-alt"></i>
        <p>Data Barang</p>
    </a>
</li>
<li class="nav-header">Data Transaksi</li>
<li class="nav-item">
    <a href="{{ url('/stok') }}" class="nav-link {{ ($activeMenu == 'stok')? 'active' : '' }}>
        <i class="nav-icon fas fa-cubes"></i>
        <p>Stok Barang</p>
    </a>
</li>
<li class="nav-item">
    <a href="{{ url('/barang') }}" class="nav-link {{ ($activeMenu == 'penjualan')? 'active' : '' }}>
        <i class="nav-icon fas fa-cash-register"></i>
        <p>Transaksi Penjualan</p>
    </a>
</li>
</ul>
</nav>
<!-- /.sidebar-menu -->
</div>
```

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



	<p>http://localhost/PWL_POS/public/</p>
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 m_user pada praktikum ini
2	Kita gunakan library Yajra-datable dengan mengetikkan perintah pada CMD composer require yajra/laravel-datatables:^10.0 atau composer require yajra/laravel-datatables-oracle



```
Command Prompt
- Installing psr/composer-autoload ([1.2.2,3)): Extracting archive
- Installing illuminate/middleware ([v10.48.22]): Extracting archive
- Installing psr/simple-cache ([0.3.0]): Extracting archive
- Installing psr/container ([1.0.7]): Extracting archive
- Installing psr/http-message ([1.0.1]): Extracting archive
- Installing illuminate/collection ([v10.48.22]): Extracting archive
- Installing symfony/polyfill-ctype ([v1.31.0]): Extracting archive
- Installing illuminate/support ([v10.48.22]): Extracting archive
- Installing illuminate/container ([v10.48.22]): Extracting archive
- Installing illuminate/queue ([v10.48.22]): Extracting archive
- Installing illuminate/events ([v10.48.22]): Extracting archive
- Installing symfony/finders ([v6.4.1]): Extracting archive
- Installing illuminate/filesystem ([v10.48.22]): Extracting archive
- Installing illuminate/session ([v10.48.22]): Extracting archive
- Installing illuminate/translation ([v10.48.22]): Extracting archive
- Installing illuminate/configurable-laravel ([v1.31.0]): Extracting archive
- Installing symfony/stripe ([v6.4.1]): Extracting archive
- Installing symfony/service-contract ([v2.5.0]): Extracting archive
- Installing symfony/console ([v6.4.12]): Extracting archive
- Installing laravel/framework ([v10.48.22]): Extracting archive
- Installing illuminate/queue ([v10.48.22]): Extracting archive
- Installing psr/event-dispatcher ([1.0.0]): Extracting archive
- Installing symfony/event-dispatcher-contracts ([v3.3.0]): Extracting archive
- Installing symfony/event-dispatcher ([v6.4.1]): Extracting archive
- Installing symfony/var-dumper ([v6.4.1]): Extracting archive
- Installing psr/cache ([1.0.0]): Extracting archive
- Installing illuminate/database ([v10.40.22]): Extracting archive
- Installing yajra/laravel-databases-oracle ([v10.11.0]): Extracting archive
- Installing yajra/laravel-databases-html ([v10.12.0]): Extracting archive
- Installing league/fractal ([v10.20.1]): Extracting archive
- Installing yajra/laravel-database ([v10.48.22]): Extracting archive
- Installing illuminate/console ([v10.48.22]): Extracting archive
- Installing yajra/laravel-database-edition ([v1.25.0]): Extracting archive
- Installing yajra/laravel-database-buttons ([v10.0.5]): Extracting archive
- Installing yajra/laravel-database ([v10.0.5]): Extracting archive

0 packages suggested by dependencies are added by now. To see details,
run 'composer suggest' or file a bug report.

? packages you are using are looking for funding.
Use the 'composer fund' command to find out more!
No security vulnerability advisories found.

C:\Users\MSI>
```

Activate Windows
Go to Settings to activate Windows.

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

```
routes > 🌐 web.php > 📁 Closure
1  <?php
2  |
3  |
4  use App\Http\Controllers\WelcomeController;
5  use Illuminate\Support\Facades\Route;
6  use App\Http\Controllers\UserController;
7  |
8  Route::get('/', [WelcomeController::class, 'index']);
9  |
10 Route::group(['prefix' => 'user'], function () {
11     Route::get('/', [UserController::class, 'index']); // menampilkan halaman awal user
12     Route::post('/list', [UserController::class, 'list']); // menampilkan data user dalam bentuk json untuk datatables
13     Route::get('/create', [UserController::class, 'create']); // menampilkan halaman form tambah user
14     Route::post('/store', [UserController::class, 'store']); // menyimpan data user baru
15     Route::get('/{id}', [UserController::class, 'show']); // menampilkan detail user
16     Route::get('/{id}/edit', [UserController::class, 'edit']); // menampilkan halaman form edit user
17     Route::put('/{id}', [UserController::class, 'update']); // menyimpan perubahan data user
18     Route::delete('/{id}', [UserController::class, 'destroy']); // menghapus data user
19 });
20
21
22
23
24
25
26
27
28
```

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

```
app > Http > Controllers > 🌐 UserController.php ...
1  <?php
2  |
3  namespace App\Http\Controllers;
4  |
5  use App\Models\UserModel;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Hash;
8  |
9  9 references | 0 implementations
10 class UserController extends Controller
11 {
12     // Menampilkan halaman awal user
13     public function index(): Factory|View
14     {
15         $breadcrumb = (object) [
16             'title' => 'Daftar User',
17             'list' => ['Home', 'User']
18         ];
19
20         $page = (object) [
21             'title' => 'Daftar user yang terdaftar dalam sistem'
22         ];
23
24         $activeMenu = 'user'; //set menu yang sedang aktif
25
26         return view('user.index', data: ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
27     }
28 }
```



5

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

```
resources > views > user > index.blade.php > ...
1  @extends('layouts.template')
2
3  @section('content')
4      <div class="card card-outline card-primary">
5          <div class="card-header">
6              <h3 class="card-title">{{ $page->title }}</h3>
7              <div class="card-tools">
8                  <a class="btn btn-sm btn-primary mt-1" href="{{ url('user/create') }}>Tambah</a>
9              </div>
10         </div>
11         <div class="card-body">
12             <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></tr>
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: "",
63                             orderable: false,
64                             searchable: false
65                         }
66                     ]
67                 });
68             });
69         </script>
70     @endpush
71 
```

6

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



```
resources > views > layouts > template.blade.php > @html > @head > @link
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="utf-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1">
6    <title>{{ config('app.name', 'PWL Laravel Starter Code') }}</title>
7
8    <meta name="csrf-token" content="{{ csrf_token() }}> <!-- Untuk mengirimkan token Laravel CSRF pada setiap request ajax-->
9
10   <!-- Google Font: Source Sans Pro -->
11   <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,700&display=swap">
12   <!-- Font Awesome -->
13   <link rel="stylesheet" href="{{ asset('adminlte/plugins/fontawesome-free/css/all.min.css') }}>
14   <!-- DataTables -->
15   <link rel="stylesheet" href="{{ asset('adminlte/plugins/datatables-bs4/css/dataTables.bootstrap4.min.css') }}>
16   <link rel="stylesheet" href="{{ asset('adminlte/plugins/datatables-responsive/css/responsive.bootstrap4.min.css') }}>
17   <link rel="stylesheet" href="{{ asset('adminlte/plugins/datatables-buttons/css/buttons.bootstrap4.min.css') }}>
18   <!-- Theme style -->
19   <link rel="stylesheet" href="{{ asset('adminlte/dist/css/adminlte.min.css') }}>
20
21   @stack('css') <!-- Digunakan untuk memanggil custom css dari perintah push('css') pada masing-masing view -->
22 </head>
23
24 <body class="hold-transition sidebar-mini">
25 <!-- Site wrapper -->
26 <div class="wrapper">
27   <!-- Navbar -->
28   @include('layouts.header')
29   <!-- /.navbar -->
30
31   <!-- Main Sidebar Container -->
32   <aside class="main-sidebar sidebar-dark-primary elevation-4">
33     <!-- Brand Logo -->
34     <a href="{{ url('/') }}" class="brand-link">
35       | 
36       | <span class="brand-text font-weight-light">PWL - Starter Code</span>
37     </a>
38
39     <!-- Sidebar -->
40     @include('layouts.sidebar')
41     <!-- /.sidebar -->
42   </aside>
43
44   <!-- Content Wrapper. Contains page content -->
45   <div class="content-wrapper">
46     <!-- Content Header (Page Header) -->
47     @include('layouts.breadcrumb')
48
49     <!-- Main content -->
50     <section class="content">
51       | @yield('content')
52     </section>
53     <!-- /.content -->
54   </div>
55   <!-- /.content-wrapper -->
56
57   @include('layouts.footer')
58 </div>
59   <!-- /.wrapper -->
60
61   <!-- jQuery -->
62   <script src="{{ asset('adminlte/plugins/jquery/jquery.min.js') }}></script>
63   <!-- Bootstrap 4 -->
64   <script src="{{ asset('adminlte/plugins/bootstrap/js/bootstrap.bundle.min.js') }}></script>
65   <!-- DataTables & Plugins -->
66   <script src="{{ asset('adminlte/plugins/datatables/jquery.dataTables.min.js') }}></script>
67   <script src="{{ asset('adminlte/plugins/datatables-bs4/js/dataTables.bootstrap4.min.js') }}></script>
68   <script src="{{ asset('adminlte/plugins/datatables-responsive/js/dataTables.responsive.min.js') }}></script>
69   <script src="{{ asset('adminlte/plugins/datatables-responsive/js/responsive.bootstrap4.min.js') }}></script>
70   <script src="{{ asset('adminlte/plugins/datatables-buttons/js/dataTables.buttons.min.js') }}></script>
71   <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.bootstrap4.min.js') }}></script>
72   <script src="{{ asset('adminlte/plugins/jsszlip/jsszlip.min.js') }}></script>
73   <script src="{{ asset('adminlte/plugins/pdfmake/pdfmake.min.js') }}></script>
74   <script src="{{ asset('adminlte/plugins/vfs_fonts.js') }}></script>
75   <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.html5.min.js') }}></script>
76   <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.print.min.js') }}></script>
77   <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.colvis.min.js') }}></script>
78
79   <!-- AdminLTE App -->
80   <script src="{{ asset('adminlte/dist/js/adminlte.min.js') }}></script>
81   <script>
82     // Untuk mengirimkan token Laravel CSRF pada setiap request ajax
83     $ajaxSetup({headers: {'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')}});
84   </script>
85   @stack('js') <!-- Digunakan untuk memanggil custom js dari perintah push('js') pada masing-masing view -->
86 </body>
87 </html>
```

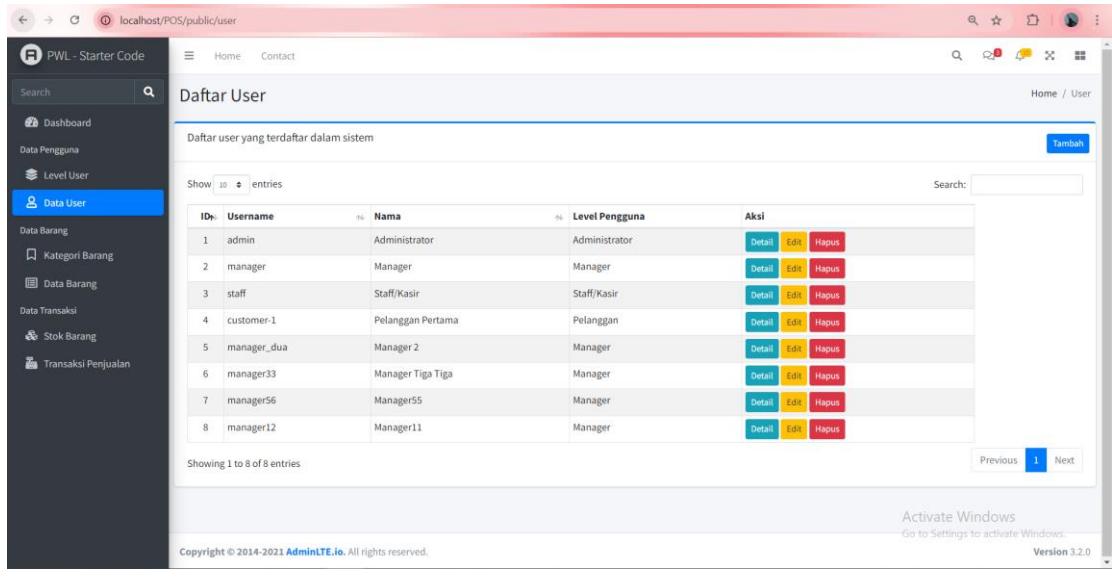
7	Untuk bisa menangkap request data untuk datatable, kita buat fungsi <code>list()</code> pada <code>UserController.php</code> seperti berikut
---	--



```
app > Http > Controllers > UserController.php > PHP > UserController > index()
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\UserModel;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Hash;
8  use Yajra\Datatables\Datatables;
9
10
11  class UserController extends Controller
12  {
13      // Menampilkan halaman awal user
14      public function index(): Factory|View
15      {
16          $breadcrumb = (object) [
17              'title' => 'Daftar User',
18              'list' => ['Home', 'User']
19          ];
20
21          $page = (object) [
22              'title' => 'Daftar user yang terdaftar dalam sistem'
23          ];
24
25          $activeMenu = 'user'; //set menu yang sedang aktif
26
27          return view('user.index', data: ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
28
29      }
30
31      // Ambil data user dalam bentuk json untuk datatables
32      public function list(Request $request): JsonResponse
33      {
34          $users = UserModel::select(columns: 'user_id', 'username', 'nama', 'level_id')->with(relations: 'level');
35          return Datatables::of($users)
36              // menambahkan kolom index / no urut (default nama kolom: DT_RowIndex)
37              ->addIndexColumn()
38              ->addColumn('aksi', function ($user): string { // menambahkan kolom aksi
39                  $btn = '<a href="'.url(path: '/user/' . $user->user_id).'" class="btn btn-info btn-sm">Detail</a> ';
40                  $btn .= '<a href="'.url(path: '/user/' . $user->user_id . '/edit).'" class="btn btn-warning btn-sm">Edit</a> ';
41                  $btn .= '<form class="d-inline-block" method="POST" action="'.url(path: '/user/' . $user->user_id).'">';
42                  $btn .= csrf_field() . method_field(method: 'DELETE') .
43                  '<button type="submit" class="btn btn-danger btn-sm" onclick="return confirm(\"Apakah Anda yakin menghapus data ini?\");">Hapus</button></form>';
44                  return $btn;
45              })
46              ->rawColumns(['aksi']) // memberitahu bahwa kolom aksi adalah html
47              ->make(true);
48      }
49  }
```

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



	<p>Berdasarkan percobaan pertama pada praktikum ini outputnya seperti gambar di atas namun setelah saya cocokan lagi kodenya sudah sesuai namun tetap error begitu, sehingga saya memutuskan untuk mengulang dengan Membuat folde POS.</p>  <table border="1"><thead><tr><th>ID</th><th>Username</th><th>Nama</th><th>Level Pengguna</th><th>Aksi</th></tr></thead><tbody><tr><td>1</td><td>admin</td><td>Administrator</td><td>Administrator</td><td>[Detail] [Edit] [Hapus]</td></tr><tr><td>2</td><td>manager</td><td>Manager</td><td>Manager</td><td>[Detail] [Edit] [Hapus]</td></tr><tr><td>3</td><td>staff</td><td>Staff/Kasir</td><td>Staff/Kasir</td><td>[Detail] [Edit] [Hapus]</td></tr><tr><td>4</td><td>customer-1</td><td>Pelanggan Pertama</td><td>Pelanggan</td><td>[Detail] [Edit] [Hapus]</td></tr><tr><td>5</td><td>manager_dua</td><td>Manager 2</td><td>Manager</td><td>[Detail] [Edit] [Hapus]</td></tr><tr><td>6</td><td>manager33</td><td>Manager Tiga Tiga</td><td>Manager</td><td>[Detail] [Edit] [Hapus]</td></tr><tr><td>7</td><td>manager56</td><td>Manager55</td><td>Manager</td><td>[Detail] [Edit] [Hapus]</td></tr><tr><td>8</td><td>manager12</td><td>Manager11</td><td>Manager</td><td>[Detail] [Edit] [Hapus]</td></tr></tbody></table>	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	manager33	Manager Tiga Tiga	Manager	[Detail] [Edit] [Hapus]	7	manager56	Manager55	Manager	[Detail] [Edit] [Hapus]	8	manager12	Manager11	Manager	[Detail] [Edit] [Hapus]
ID	Username	Nama	Level Pengguna	Aksi																																										
1	admin	Administrator	Administrator	[Detail] [Edit] [Hapus]																																										
2	manager	Manager	Manager	[Detail] [Edit] [Hapus]																																										
3	staff	Staff/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	manager33	Manager Tiga Tiga	Manager	[Detail] [Edit] [Hapus]																																										
7	manager56	Manager55	Manager	[Detail] [Edit] [Hapus]																																										
8	manager12	Manager11	Manager	[Detail] [Edit] [Hapus]																																										
9	Selanjutnya kita modifikasi UserController.php untuk form tambah data user																																													
	<pre>app > Http > Controllers > UserController.php > ... 11 class UserController extends Controller 48 // Menampilkan halaman form tambah user 49 1 reference 0 overrides 50 public function create(): Factory View 51 { 52 \$breadcrumb = (object) [53 'title' => 'Tambah User', 54 'list' => ['Home', 'User', 'Tambah'] 55]; 56 57 \$page = (object) [58 'title' => 'Tambah user baru' 59]; 60 61 \$level = LevelModel::all(); // ambil data level untuk ditampilkan di form 62 \$activeMenu = 'user'; // set menu yang sedang aktif 63 64 return view('user.create', data: ['breadcrumb' => \$breadcrumb, 'page' => \$page, 'level' => \$level, 'activeMenu' => \$activeMenu]); 65 } </pre>																																													
10	Sekarang kita buat form untuk menambah data, kita buat file PWL/resources/views/user/create.blade.php																																													



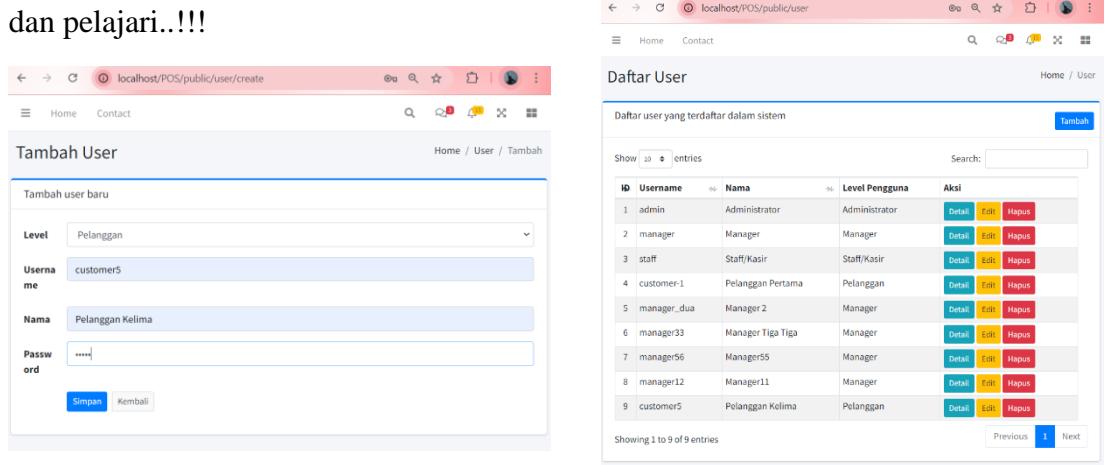
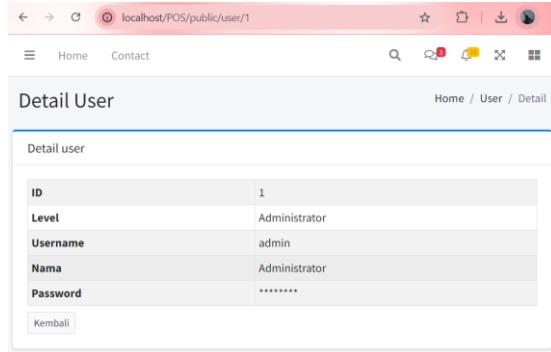
```
resources > views > user > create.blade.php > div.card.card-outline.card-primary > div.card-body > form.form-horizontal > div.form-group.row
1  @extends('layouts.template')
2
3  @section('content')
4      <div class="card card-outline card-primary">
5          <div class="card-header">
6              <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
```

11

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

```
app > Http > Controllers > UserController.php > PHP > UserController > store()
11  class UserController extends Controller
12
13      // Menyimpan data user baru
14      1reference | overrides
15      public function store (Request $request): mixed|RedirectResponse
16  {
17      $request->validate(rules: [
18          // username harus diisi, berupa string, minimal 3 karakter, dan bernilai unik di tabel m_user kolom username
19          'username' => 'required|string|min:3 | unique:m_user,username',
20          'nama' => 'required|string | max: 100', // nama harus diisi, berupa string, dan maksimal 100 karakter
21          'password' => 'required|min:5', // password harus diisi dan minimal 5 karakter
22          'level_id' => 'required|integer' // level_id harus diisi dan berupa angka
23      ]);
24
25      UserModel::create(attributes: [
26          'username' => $request->username,
27          'nama' => $request->nama,
28          'password' => bcrypt(value: $request->password), // password dienkripsi sebelum disimpan
29          'level_id' => $request->level_id
30      ]);
31      return redirect(to: '/user')->with(key: 'success', value: 'Data user berhasil disimpan');
32  }
```



12	Sekarang coba kalian buka form tambah data user dengan klik tombol tambah. Amati dan pelajari..!!!
	 <p>Pada saat mengklik button tambah maka akan terarah ke dalam menu tambah user seperti gambar di kiri yang kemudian jika data user berhasil ditambahkan maka akan kembali ke menu yang menampilkan daftar user dan pada baris terakhir menunjukan data yang baru saja ditambahkan.</p>
13	Selanjutnya, kita masuk pada bagian menampilkan detail data user (klik tombol detail) pada halaman user. Route yang bertugas untuk menangkap request detail adalah
	 <p>Dari hasil tampilan di atas dapat diketahui bahwa detail user yang ditampilkan merupakan berdasarkan data lengkap yang diambil dari database m_user mengenai data setiap user nya.</p>
14	Jadi kita buat/modifikasi fungsi show() pada UserController.php seperti berikut



```
app > Http > Controllers > UserController.php > ...
12 class UserController extends Controller
13
14     // Menampilkan detail user
15     1 reference | 0 overrides
16     public function show(string $id): Factory|View
17     {
18         $user = UserModel::with(relations: 'level')->find(id: $id);
19         $breadcrumb = (object) [
20             'title' => 'Detail User',
21             'list' => ['Home', 'User', 'Detail']
22         ];
23         $page = (object) [
24             'title' => 'Detail user'
25         ];
26         $activeMenu = 'user'; // set menu yang sedang aktif
27         return view(view: 'user.show', data: ['breadcrumb' => $breadcrumb, 'page' => $page, 'user' => $user, 'activeMenu' => $activeMenu]);
28     }
29
30 }
```

15

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

```
resources > views > user > show.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                     <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     @push('css')
43     @endpush
44     @push('js')
45     @endpush
```



16	<p>Sekarang kalian coba untuk melihat detail data user di browser, dan coba untuk mengetikkan id yang salah contoh http://localhost/PWL/public/user/100 amati apa yang terjadi, dan laporan!!!</p>
17	<p>Berdasarkan pemahaman saya dari hasil praktikum di atas menunjukkan detail user pada user id 100 menunjukkan terjadi kesalahan yang mana data yang dicari tidak ditemukan, hal tersebut dikarenakan data user yang ada pada database belum mencapai 100 id sehingga tidak dapat ditemukan. Namun, jika data id yang dicari sudah ada di database seperti id 3 diatas maka tampilannya akan menampilkan detail user mengenai data user yang diambil dari database m_user.</p>
18	<p>Selanjutnya, kita masuk pada bagian untuk memodifikasi data user. Route yang bertugas untuk menangkap request edit adalah</p> <p>Jadi kita buat fungsi edit () dan update () pada UserController.php</p>
19	<p>Selanjutnya, kita buat view untuk melakukan proses edit data user di PWL/resources/views/user/edit.blade.php</p>



```
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">{{ $page->title }}6              <div class="card-tools"></div>  
7          </div>  
8          <div class="card-body">  
9              @empty($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                  <a href="{{ url('user') }}" class="btn btn-sm btn-default mt-2">Kembali</a>  
15              @else  
16                  <form method="POST" action="{{ url('/user/' . $user->user_id) }}" class="form-horizontal">  
17                      @csrf  
18                      {{! method_field('PUT') }}<!-- tambahkan baris ini untuk proses edit yang butuh  
19                          method PUT -->  
20                      <div class="form-group row">  
21                          <label class="col-1 control-label col-form-label">Level</label>  
22                          <div class="col-11">  
23                              <select class="form-control" id="level_id" name="level_id" required>  
24                                  <option value=""> Pilih Level </option>  
25                                  @foreach ($level as $item)  
26                                      <option value="{{ $item->level_id }}>@if ($item->level_id == $user->level_id) selected @endif  
27                                          {{ $item->level_nama }}</option>  
28                                  @endforeach  
29                              </select>  
30                              @error('level_id')  
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">Username</label>  
37                          <div class="col-11">  
38                              <input type="text" class="form-control" id="username" name="username"  
39                                  value="{{ old('username', $user->username) }}" required>  
40                              @error('username')  
41                                  <small class="form-text text-danger">{{ $message }}</small>  
42                              @enderror  
43                          </div>  
44                      </div>  
45                      <div class="form-group row">  
46                          <label class="col-1 control-label col-form-label">Nama</label>  
47                          <div class="col-11">  
48                              <input type="text" class="form-control" id="nama" name="nama"  
49                                  value="{{ old('nama', $user->nama) }}" required>  
50                              @error('nama')  
51                                  <small class="form-text text-danger">{{ $message }}</small>  
52                              @enderror  
53                          </div>  
54                      </div>  
55                      <div class="form-group row">  
56                          <label class="col-1 control-label col-form-label">Password</label>  
57                          <div class="col-11">  
58                              <input type="password" class="form-control" id="password" name="password">  
59                              @error('password')  
60                                  <small class="form-text text-danger">{{ $message }}</small>  
61                              @else  
62                                  <small class="form-text text-muted">Abaikan (jangan diisi) jika tidak ingin  
63                                      mengganti password user.</small>  
64                              @enderror  
65                          </div>  
66                      </div>  
67                      <div class="form-group row">  
68                          <label class="col-1 control-label col-form-label"></label>  
69                          <div class="col-11">  
70                              <button type="submit" class="btn btn-primary btn-sm">Simpan</button>  
71                              <a class="btn btn-sm btn-default ml-1" href="{{ url('user') }}>Kembali</a>  
72                          </div>  
73                      </div>  
74                  </form>  
75              @endempty  
76          </div>  
77      @endsection  
78      @push('css')  
79      @endpush  
80      @push('js')  
81      @endpush
```



20

Sekarang kalian coba untuk mengedit data user di browser, amati, pahami, dan lapor kan!

The screenshot displays two web pages. The left page is a 'Daftar User' (User List) table with the following data:

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	manager33	Manager Tiga Tiga	Manager	[Detail] [Edit] [Hapus]
7	manager56	Manager55	Manager	[Detail] [Edit] [Hapus]
8	manager12	Manager11	Manager	[Detail] [Edit] [Hapus]
9	customer5	Pelanggan Kelima	Pelanggan	[Detail] [Edit] [Hapus]

The right page is an 'Edit User' form with the following fields:

Level	Pelanggan
Username	Pelanggan5
Nama	Pelanggan Kelima
Password	(Abalkan (jangan diisi) jika tidak ingin mengganti password user.)

Buttons for 'Simpan' (Save) and 'Kembali' (Back) are at the bottom.

Disini saya akan melakukan pengeditan data pada user id 9 yaitu customer5 yang ingin saya ubah nama username nya menjadi Pelanggan5 dengan mengklik button edit pada baris id 9 kemudian klik simpan untuk menyimpan hasil perubahan, maka tampilan daftar user akan menampilkan seluruh data terupdate yang mana dapat dilihat bahwa customer5 sudah berubah username menjadi Pelanggan5.

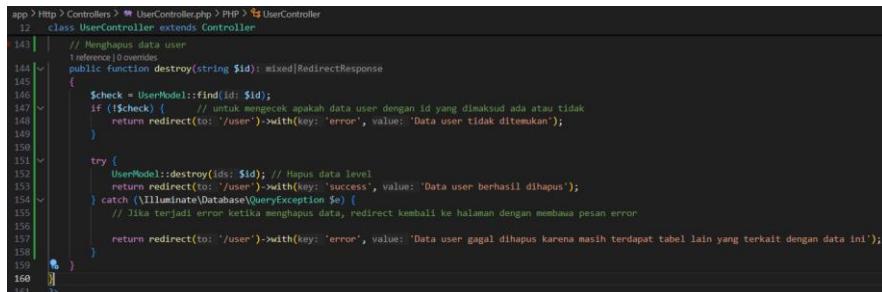
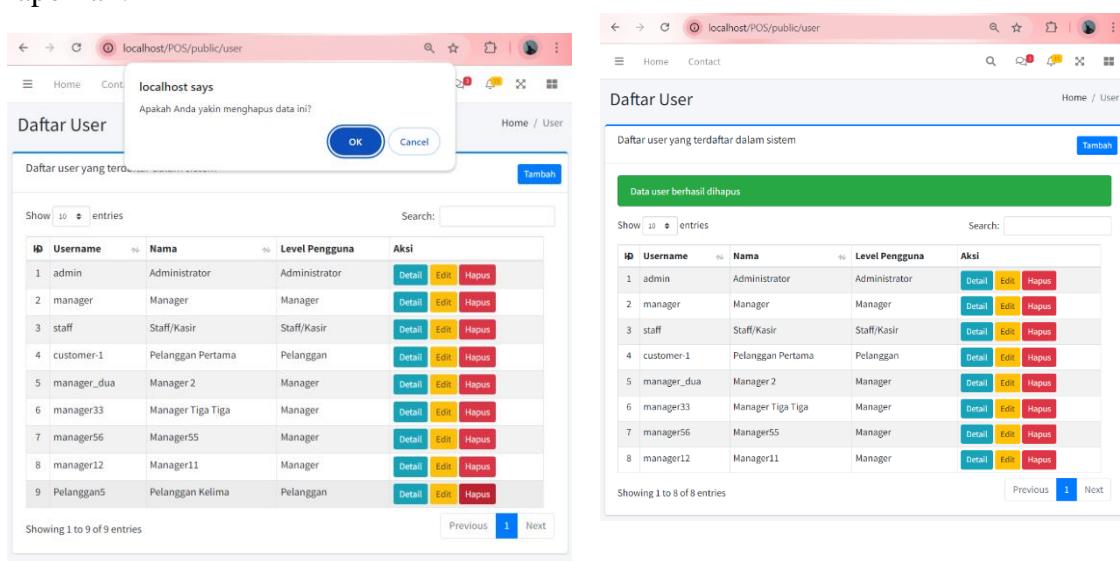
The screenshot shows the 'Daftar User' table again, but now the row for customer5 has been updated. The 'Username' column now contains 'Pelanggan5' instead of 'customer5'. All other data remains the same.

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	manager33	Manager Tiga Tiga	Manager	[Detail] [Edit] [Hapus]
7	manager56	Manager55	Manager	[Detail] [Edit] [Hapus]
8	manager12	Manager11	Manager	[Detail] [Edit] [Hapus]
9	Pelanggan5	Pelanggan Kelima	Pelanggan	[Detail] [Edit] [Hapus]

21

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

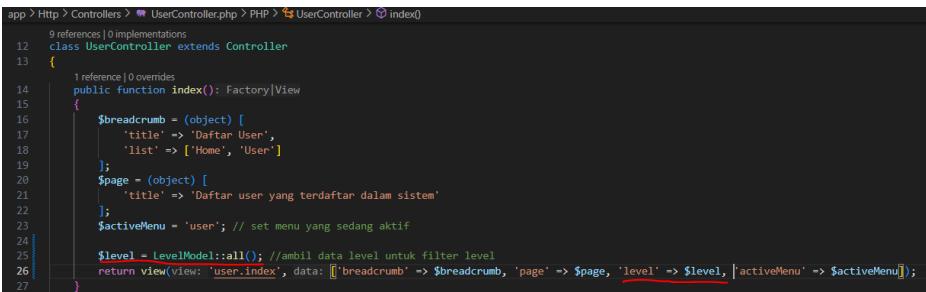
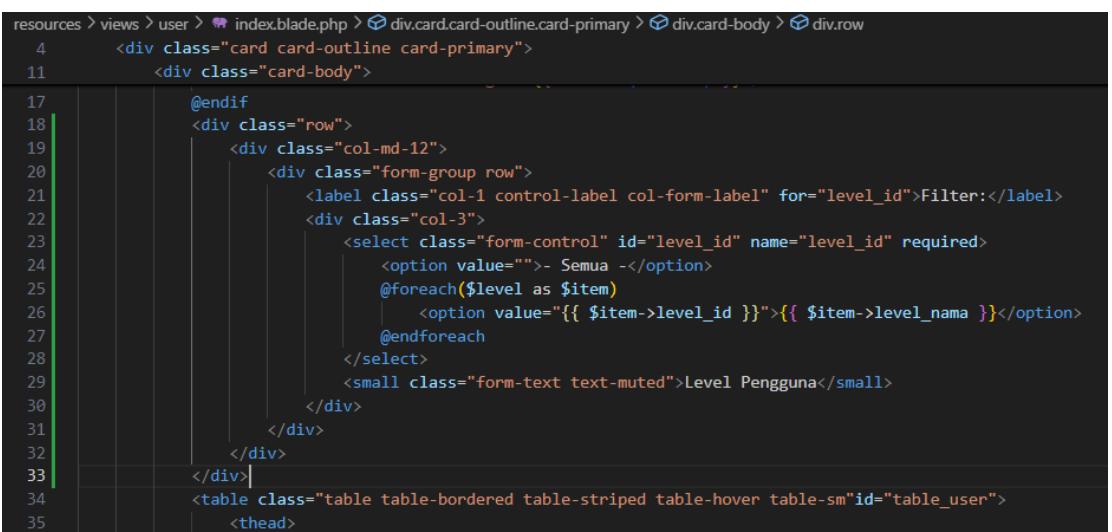


	Route::::delete ('/{id}', 'destroy']);	[UserController::class,
22	Jadi kita buat fungsi destroy() pada UserController.php	
23	Selanjutnya kita modifikasi file PWL/resources/views/user/index.blade.php untuk menambahkan tampilan jika ada pesan error	
24	Kemudian jalankan browser untuk menghapus salah satu data user. Amati dan laporkan!	



	Ketika di klik button hapus pada salah satu user maka akan menampilkan pop up konfirmasi penghapusan, dan jika penghapusan data berhasil akan terdapat keterangan hijau seperti gambar diatas.
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	Kita modifikasi fungsi index() di UserController.php untuk menambahkan data yang ingin dijadikan kategori untuk data <i>filtering</i> 
2	Kemudian kita modifikasi view untuk menampilkan data filtering pada PWL/resources/views/user/index.blade.php 



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
	<pre>resources > views > user > index.blade.php > script > <function> > dataUser > "data" 51 <script> 52 \$(document).ready(function(){ 53 var dataUser = \$('#table_user').DataTable({ 54 // serverSide: true, jika ingin menggunakan server side p 55 serverSide: true, 56 ajax: { 57 "url": "{{ url('user/list') }}", 58 "dataType": "json", 59 "type": "POST" 60 "data": function(d){ 61 d.level_id = \$('#level_id').val(); 62 } 63 }, 64 }); 65 }); 66 </pre>
4	Kemudian kita edit pada bagian akhir script @push('js') untuk menambahkan listener jika data filtering dipilih
	<pre>resources > views > user > index.blade.php > script > <function> > <function> 51 <script> 52 \$(document).ready(function(){ 53 var dataUser = \$('#table_user').DataTable({ 54 "url": "{{ url('user/list') }}", 55 "dataType": "json", 56 "type": "POST" 57 "data": function(d){ 58 d.level_id = \$('#level_id').val(); 59 } 60 }); 61 }); 62 }); 63 </pre>
5	Tahapan akhir adalah memodifikasi fungsi list() pada UserController.php yang digunakan untuk menampilkan data pada datatable
	<pre>app > Http > Controllers > UserController.php > PHP > UserController > list() 12 class UserController extends Controller 13 public function list(Request \$request): JsonResponse 14 { 15 \$users = UserModel::select(columns: 'user_id', 'username', 'nama', 'level_id') 16 ->with(relations: 'level'); 17 18 //Filter data user berdasarkan level_id 19 if (\$request->level_id){ 20 \$users->where(column: 'level_id', operator: \$request->level_id); 21 } 22 23 return DataTables::of(\$users) 24 // menambahkan kolom index / no urut (default nama kolom: DT_RowIndex) 25 ->addIndexColumn() 26 ->addColumn('aksi', function (\$user): string { // menambahkan kolom aksi 27 return \$user->aksi; 28 }); 29 } 30 </pre>



6	Bagian akhir adalah kita coba jalankan di browser dengan akses menu user, maka akan tampil seperti berikut
7	Selamat, sekarang Laravel Starter Code sudah ada filtering dan searching data. Starter Code sudah bisa digunakan dalam membangun sebuah sistem berbasis website.
8	Jangan lupa commit dan push ke github PWL_POS kalian

Soal Praktikum

Implementasikan web layout dan datatables, pada menu berikut ini

- ✓ Tabel m_level
- ✓ Tabel m_kategori
- ✓ Tabel m_supplier
- ✓ Tabel m_barang

A. Tabel m_level

Langkah	Jawaban/Deskripsi



1	<p>Pertama saya melakukan penambahan pada route untuk menentukan tampilan menu apa saja yang akan ada terkait tabel <code>m_level</code>, seperti gambar di bawah ini yang saya samakan dengan tabel <code>m_user</code> dari praktikum di atas.</p> <pre>routes > 🌐 web.php > ... 1 <?php 2 3 4 use App\Http\Controllers\WelcomeController; 5 use Illuminate\Support\Facades\Route; 6 use App\Http\Controllers\UserController; 7 use App\Http\Controllers\LevelController; 8 9 Route::get(uri: '/', action: [WelcomeController::class, 'index']); 10 11 Route::group(attributes: ['prefix' => 'user'], routes: function () void { 12 Route::get(uri: '/', action: [UserController::class, 'index']); // menampilkan halaman awal user 13 Route::post(uri: '/list', action: [UserController::class, 'list']); // menampilkan data user dalam bentuk json untuk datatables 14 Route::get(uri: '/create', action: [UserController::class, 'create']); // menampilkan halaman form tambah user 15 Route::post(uri: '/', action: [UserController::class, 'store']); // menyimpan data user baru 16 Route::get(uri: '/{id}', action: [UserController::class, 'show']); // menampilkan detail user 17 Route::get(uri: '/{id}/edit', action: [UserController::class, 'edit']); // menampilkan halaman form edit user 18 Route::put(uri: '/{id}', action: [UserController::class, 'update']); // menyimpan perubahan data user 19 Route::delete(uri: '/{id}', action: [UserController::class, 'destroy']); // menghapus data user 20 }); 21 22 // Route::get('/', [WelcomeController::class, 'index']); 23 Route::group(attributes: ['prefix' => 'level'], routes: function () void { 24 Route::get(uri: '/', action: [LevelController::class, 'index']); // menampilkan halaman awal level 25 Route::post(uri: '/list', action: [LevelController::class, 'list']); // menampilkan data level dalam bentuk json untuk datatables 26 Route::get(uri: '/create', action: [LevelController::class, 'create']); // menampilkan halaman form tambah level 27 Route::post(uri: '/', action: [LevelController::class, 'store']); // menyimpan data level baru 28 Route::get(uri: '/{id}', action: [LevelController::class, 'show']); // menampilkan detail level 29 Route::get(uri: '/{id}/edit', action: [LevelController::class, 'edit']); // menampilkan halaman form edit level 30 Route::put(uri: '/{id}', action: [LevelController::class, 'update']); // menyimpan perubahan data level 31 Route::delete(uri: '/{id}', action: [LevelController::class, 'destroy']); // menghapus data level 32 }); 33 ?></pre>
2	<p>Kemudian Membuat file Model tentang <code>m_level</code> yang saya beri nama <code>LevelModel</code></p> <pre>app > Models > 🌐 LevelModel.php > PHP Intelephense > 🛡 LevelModel > ⚒ user 1 <?php 2 3 namespace App\Models; 4 5 use Illuminate\Database\Eloquent\Factories\HasFactory; 6 use Illuminate\Database\Eloquent\Model; 7 use Illuminate\Database\Eloquent\Relations\BelongsTo; 8 use Illuminate\Database\Eloquent\Relations\hasMany; 9 10 class LevelModel extends Model 11 { 12 use HasFactory; 13 14 protected \$table = 'm_level'; 15 protected \$primaryKey = 'level_id'; 16 17 public function user():hasMany 18 { 19 return \$this->hasMany(related: UserModel::class, foreignKey: 'user_id', localKey: 'user_id'); 20 } 21 protected \$fillable = ['level_kode', 'level_nama']; 22 }</pre>



3

Lalu membuat view pada

PWL/resources/views/level/index.blade.php seperti berikut

```
resources > views > level > indexblade.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 }}
```



4	<p>Kemudian membuat file LevelController.php. Dan juga membuat fungsi index() dan list untuk menampilkan halaman awal level yang menampilkan data level.</p>
5	<p>Berikut adalah tampilan awal halaman level ketika di akses.</p>
6	<p>Sekarang membuat form untuk menambah data, maka diperlukan file PWL/resources/views/level/create.blade.php seperti berikut.</p>



```
resources > views > level > create.blade.php > div.card.card-outline.card-primary > div.card-body > form.form-horizontal > div.form-group.row > div.col-11
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              <form method="POST" action="{{ url('level') }}" class="form-horizontal">
10                 @csrf
11                 <div class="form-group row">
12                     <label class="col-1 control-label col-form-label">Kode Level</label>
13                     <div class="col-11">
14                         <input type="text" class="form-control" id="level_kode" name="level_kode"
15                             value="{{ old('level_kode') }}" required>
16                         @error('level_kode')
17                             <small class="form-text text-danger">{{ $message }}</small>
18                         @enderror
19                     </div>
20                 </div>
21                 <div class="form-group row">
22                     <label class="col-1 control-label col-form-label">Nama Level</label>
23                     <div class="col-11">
24                         <input type="text" class="form-control" id="level_nama" name="level_nama"
25                             value="{{ old('level_nama') }}" required>
26                         @error('level_nama')
27                             <small class="form-text text-danger">{{ $message }}</small>
28                         @enderror
29                     </div>
30                 </div>
31                 <div class="form-group row">
32                     <label class="col-1 control-label col-form-label"></label>
33                     <div class="col-11">
34                         <button type="submit" class="btn btn-primary btn-sm">Simpan</button>
35                         <a class="btn btn-sm btn-default ml-1" href="{{ url('level') }}">Kembali</a>
36                     </div>
37                 </div>
38             </form>
39         </div>
40     @endsection
41     @push('css')
42     @endpush
43     @push('js')
44     @endpush
45 
```

Dan melakukan penambahan fungsi `create()` dan `store()` di file `LevelController.php` dengan isi seperti berikut

```
app > Http > Controllers > LevelController.php > PHP Intelephense > LevelController > list
10  class LevelController extends Controller
11
12      // Menampilkan halaman form tambah level
13      public function create(): Factory\View
14      {
15          $breadcrumb = (object) [
16              'title' => 'Tambah Level User',
17              'list' => [ 'Home', 'Level User', 'Tambah' ]
18          ];
19          $page = (object) [
20              'title' => 'Tambah level user baru'
21          ];
22
23          $activeMenu = 'level'; // set menu yang sedang aktif
24          return view(view: 'level.create', data: ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
25      }
26
27      // Menyimpan data level baru
28      public function store(Request $request): mixed|RedirectResponse
29      {
30          $request->validate(rules: [
31              'level_kode' => 'required|string|unique:m_level,level_kode', // level_kode harus unik
32              'level_nama' => 'required|string|max:100' // level_nama harus diisi dan maksimal 100 karakter
33          ]);
34
35          LevelModel::create(attributes: [
36              'level_kode' => $request->level_kode,
37              'level_nama' => $request->level_nama,
38          ]);
39
40          return redirect(to: '/level')->with(key: 'success', value: 'Data level berhasil disimpan');
41      }
42 
```



7

The screenshot shows two windows side-by-side. The left window is titled 'Tambah Level User' and contains a form with fields for 'Kode Level' (set to 'PC') and 'Nama Level' (set to 'Pensi'). The right window is titled 'Daftar Level User' and displays a table of user levels with columns for ID, Kode Level, and Nama Level. The table includes rows for Admin, Manager, Sekolah, Pelajar, and Pensi.

Gambar di atas adalah tampilan halaman penambahan level user yang mana hanya perlu memasukan kode level dan nama level, kemudian klik simpan untuk menyimpan penambahan data maka akan kembali ke halaman list daftar level user yang akan menampilkan data terupdate dan keterangan hijau yang menunjukkan data berhasil diinputkan

8

Selanjutnya membuat form untuk mengedit data, maka diperlukan file PWL/resources/views/level/edit.blade.php seperti berikut.

```
resources > views > level > edit.blade.php > @div.card.card-outline.card-primary > @div.card-body > @form.form-horizontal
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">{{ $page->title }}</h3>
6              <div class="card-tools"></div>
7          </div>
8          <div class="card-body">
9              @empty($level)
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('level') }}" class="btn btn-sm btn-default mt-2">Kembali</a>
15              @else
16                  <form method="POST" action="{{ url('/level/' . $level->level_id) }}" class="form-horizontal">
17                      @csrf
18                      {!! method_field('PUT') !!}
19                      <div class="form-group row">
20                          <label class="col-1 control-label col-form-label">Kode Level</label>
21                          <div class="col-11">
22                              <input type="text" class="form-control" id="level_kode" name="level_kode"
23                                  value="{{ old('level_kode', $level->level_kode) }}" required>
24                                  @error('level_kode')
25                                      <small class="form-text text-danger">{{ $message }}</small>
26                                  @enderror
27                          </div>
28                      </div>
29                      <div class="form-group row">
30                          <label class="col-1 control-label col-form-label">Nama Level</label>
31                          <div class="col-11">
32                              <input type="text" class="form-control" id="level_nama" name="level_nama"
33                                  value="{{ old('level_nama', $level->level_nama) }}" required>
34                                  @error('level_nama')
35                                      <small class="form-text text-danger">{{ $message }}</small>
36                                  @enderror
37                          </div>
38                      </div>
39                      <div class="form-group row">
40                          <label class="col-1 control-label col-form-label"></label>
41                          <div class="col-11">
42                              <button type="submit" class="btn btn-primary btn-sm">Simpan</button>
43                              <a class="btn btn-sm btn-default ml-1" href="{{ url('level') }}>Kembali</a>
44                          </div>
45                      </div>
46                  </form>
47                  @endempty
48              </div>
49          @endsection
50          @push('css')
51          @endpush
52          @push('js')
53          @endpush
54      
```



Dan berikut adalah tampilan halaman edit level user.

ID	Kode Level	Nama Level	Aksi
1	ADM	Administrator	[Detail] [Edit] [Hapus]
2	MNG	Manager	[Detail] [Edit] [Hapus]
3	STF	Staff/Kasir	[Detail] [Edit] [Hapus]
4	CUS	Pelanggan	[Detail] [Edit] [Hapus]
5	PIC	Person in Charge	[Detail] [Edit] [Hapus]

9 Untuk menampilkan detail data level user maka diperlukan file PWL/resources/views/level/show.blade.php seperti berikut. Dan diperlukan penambahan fungsi show() pada file LevelController.php.

```
resources > views > level > 🌐 show.blade.php > ⚒ div.card.card-outline.card-primary > ⚒ div.card-body > ⚒ table.table-bordered.table-striped.table-hover.table-sm > ⚒ tr
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">{{ $page->title }}</h3>
6    </div>
7    <div class="card-body">
8      @empty($level)
9        <div class="alert alert-danger alert-dismissible">
10          <h5><i class="fas fa-ban"></i> Kesalahan!</h5>
11          Data yang anda cari tidak ditemukan.
12        </div>
13      @else
14        <table class="table table-bordered table-striped table-hover table-sm">
15          <thead>
16            <tr>
17              <th>ID Level</th>
18              <td>{{ $level->level_id }}</td>
19            </tr>
20          </thead>
21          <tbody>
22            <tr>
23              <th>Kode Level</th>
24              <td>{{ $level->level_kode }}</td>
25            </tr>
26            <tr>
27              <th>Nama Level</th>
28              <td>{{ $level->level_nama }}</td>
29            </tr>
30          </tbody>
31        </table>
32        @endempty
33        <a href="{{ url('level') }}" class="btn btn-sm btn-default mt-2">Kembali</a>
34      </div>
35    </div>
36  @endsection
37  @push('css')
38  @endpush
39  @push('js')
40  @endpush
41
```

```
app > Http > Controllers > 🌐 LevelController.php > PHP Intelephense > 🛡 LevelController > ⚒ list
10  class LevelController extends Controller
11
12  // Menampilkan detail level
13  1 reference | 0 overrides
14  public function show(string $id): Factory|mixed|RedirectResponse|View
15  {
16    $level = LevelModel::find(id: $id);
17
18    if (!$level) {
19      return redirect(to: '/level')->with(key: 'error', value: 'Data level tidak ditemukan');
20    }
21
22    $breadcrumb = (object) [
23      'title' => 'Detail Level',
24      'list' => ['Home', 'Level', 'Detail']
25    ];
26    $page = (object) [
27      'title' => 'Detail level';
28    ];
29    $activeMenu = 'level'; // set menu yang sedang aktif
30    return view(view: 'level.show', data: ['breadcrumb' => $breadcrumb, 'page' => $page, 'level' => $level, 'activeMenu' => $activeMenu]);
31  }
```

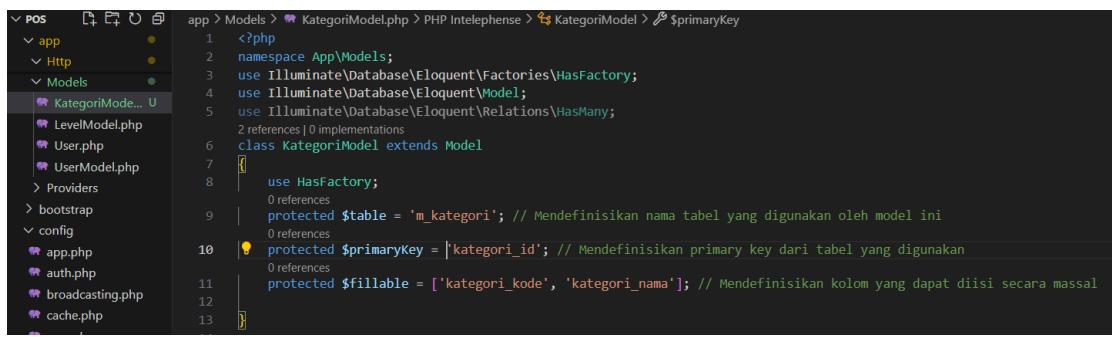


	<p>Berikut adalah tampilan halaman detail level user yang mengambil data dari database m_level.</p>
10	<p>Untuk melakukan penghapusan data maka diperlukan penambahan fungsi <code>destroy(string \$id)</code> pada file <code>LevelController.php</code>.</p> <pre>app > Http > Controllers > LevelController.php > PHP Intelephense > LevelController > edit 10 < class LevelController extends Controller 148 149 // Menghapus data level 150 1 reference 0 overrides 151 public function destroy(string \$id): mixed RedirectResponse 152 { 153 \$level = LevelModel::find(id: \$id); 154 155 if (!\$level) { 156 return redirect(to: '/level')->with(key: 'error', value: 'Data level tidak ditemukan'); 157 } 158 159 try { 160 \$level->delete(); // Hapus data level 161 return redirect(to: '/level')->with(key: 'success', value: 'Data level berhasil dihapus'); 162 } catch (\Illuminate\Database\QueryException \$e) { 163 return redirect(to: '/level')->with(key: 'error', value: 'Data level gagal dihapus karena masih terkait dengan data lain'); 164 } 165 } </pre>
	<p>Yang mana tampilan halaman nya seperti gambar diatas, yaitu ketika salah satu data dihapus maka akan terdapat pop up konfirmasi penghapusan dan keterangan yang menunjukan berhasil tidaknya data tersebut dihapus.</p>

B. Tabel m_kategori

Langkah	Jawaban/Deskripsi



1	Pertama saya melakukan penambahan pada route untuk menentukan tampilan menu apa saja yang akan ada terkait tabel m_kategori, seperti gambar di bawah ini yang saya samakan dengan menu lainnya dari praktikum di atas.
	
2	Kemudian Membuat file Model tentang m_kategori yang saya beri nama KategoriModel
	
3	Lalu membuat view pada PWL/resources/views/level/index.blade.php seperti berikut



```
resources > views > kategori > index.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">{{ $page->title }}</h3>
6              <div class="card-tools">
7                  <a class="btn btn-sm btn-primary st-1" href="{{ url('kategori/create') }}>Tambah</a>
8              </div>
9          </div>
10         <div class="card-body">
11             @if (session('success'))
12                 <div class="alert alert-success">{{ session('success') }}</div>
13             @endif
14             @if (session('error'))
15                 <div class="alert alert-danger">{{ session('error') }}</div>
16             @endif
17             <div class="row">
18                 <div class="col-md-12">
19                     <div class="form-group row">
20                         <label class="col-1 control-label col-form-label" for="kategori_id">Filter:</label>
21                         <div class="col-3">
22                             <select class="form-control" id="kategori_id" name="kategori_id" required>
23                                 <option value=""> Semua </option>
24                                 @foreach($kategori as $item)
25                                     <option value="{{ $item->kategori_id }}>{{ $item->kategori_nama }}</option>
26                                 @endforeach
27                             </select>
28                             <small class="form-text text-muted">Kategori Produk</small>
29                         </div>
30                     </div>
31                 </div>
32                 <table class="table table-bordered table-striped table-hover table-sm" id="table_kategori">
33                     <thead>
34                         <tr>
35                             <th>ID</th>
36                             <th>Kode Kategori</th>
37                             <th>Nama Kategori</th>
38                             <th>Aksi</th>
39                         </tr>
40                     </thead>
41                     <tbody>
42                         <tr>
43                             <td></td>
44                         </tr>
45                     </tbody>
46                 </table>
47             </div>
48         </div>
49     @endsection
50     @push('css')
51     @endpush
52     @push('js')
53         <script>
54             $(document).ready(function() {
55                 var datakategori = $('#table_kategori').DataTable({
56                     serverSide: true,
57                     ajax: [
58                         {
59                             "url": "{{ url('kategori/list') }}",
60                             "datatype": "json",
61                             "type": "POST",
62                             "data": function(d){
63                                 d.kategori_id = $('#kategori_id').val();
64                             }
65                         },
66                     ],
67                     columns: [
68                         {
69                             data: "DT_RowIndex",
70                             className: "text-center",
71                             orderable: false,
72                             searchable: false
73                         },
74                         {
75                             data: "kategori_kode",
76                             className: "",
77                             orderable: true,
78                             searchable: true
79                         },
80                         {
81                             data: "kategori_nama",
82                             className: "",
83                             orderable: true,
84                             searchable: true
85                         },
86                         {
87                             data: "aksi",
88                             className: "",
89                             orderable: false,
90                             searchable: false
91                         }
92                     ],
93                     });
94                 $('#kategori_id').on('change', function(){
95                     datakategori.ajax.reload();
96                 });
97             });
98         </script>
99     @endpush
100    </div>
```



4

Kemudian membuat file KategoriController.php. Dan juga membuat fungsi index() dan list() untuk menampilkan halaman awal kategori yang menampilkan data kategori.

```
app > Http > Controllers > KategoriController.php > PHP Intelephense > KategoriController > list
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\KategoriModel;
6 use Illuminate\Http\Request;
7 use Yajra\Datatables\Datatables;
8
9 references | 0 implementations
10 class KategoriController extends Controller
11 {
12     // Menampilkan halaman daftar kategori
13     1 reference | 0 overrides
14     public function index(): Factory|View
15     {
16         $breadcrumb = (object) [
17             'title' => 'Daftar Kategori',
18             'list' => ['Home', 'Kategori']
19         ];
20         $page = (object) [
21             'title' => 'Daftar kategori yang terdaftar dalam sistem'
22         ];
23         $activeMenu = 'kategori'; // set menu yang sedang aktif
24
25         $kategori = KategoriModel::all(); // ambil data kategori untuk filter kategori
26
27         return view('kategori.index', data: ['breadcrumb' => $breadcrumb, 'page' => $page, 'kategori' => $kategori, 'activeMenu' => $activeMenu]);
28     }
29
30     // Ambil data kategori dalam bentuk JSON untuk datatables
31     1 reference | 0 overrides
32     public function list(Request $request): JsonResponse
33     {
34         $kategoris = KategoriModel::select(columns: 'kategori_id', 'kategori_kode', 'kategori_nama');
35
36         // Filter data kategori berdasarkan kategori_id
37         if ($request->kategori_id) {
38             $kategoris->where(column: 'kategori_id', operator: $request->kategori_id);
39         }
40
41         return Datatables::of($kategoris)
42             ->addIndexColumn()
43             ->addColumn('aksi', function ($kategori): string { // menambahkan kolom aksi
44                 $btn = '<a href="'. url(path: "/kategori/" . $kategori->kategori_id) . '" class="btn btn-info btn-sm">Detail</a> ';
45                 $btn .= '<a href="'. url(path: "/kategori/" . $kategori->kategori_id . '/edit') . '" class="btn btn-warning btn-sm">Edit</a> ';
46                 $btn .= '<form class="d-inline-block" method="POST" action="'. url(path: "/kategori/" . $kategori->kategori_id . '/'). '>' .
47                     . csrf_field() .
48                     . '<button type="submit" class="btn btn-danger btn-sm" onclick="return confirm(\'Apakah Anda yakin menghapus data ini?\');">Hapus</button></form>';
49                 return $btn;
50             })
51             ->rawColumns(['aksi']) // memberitahu bahwa kolom aksi adalah HTML
52             ->make(true);
53     }
}
```

5

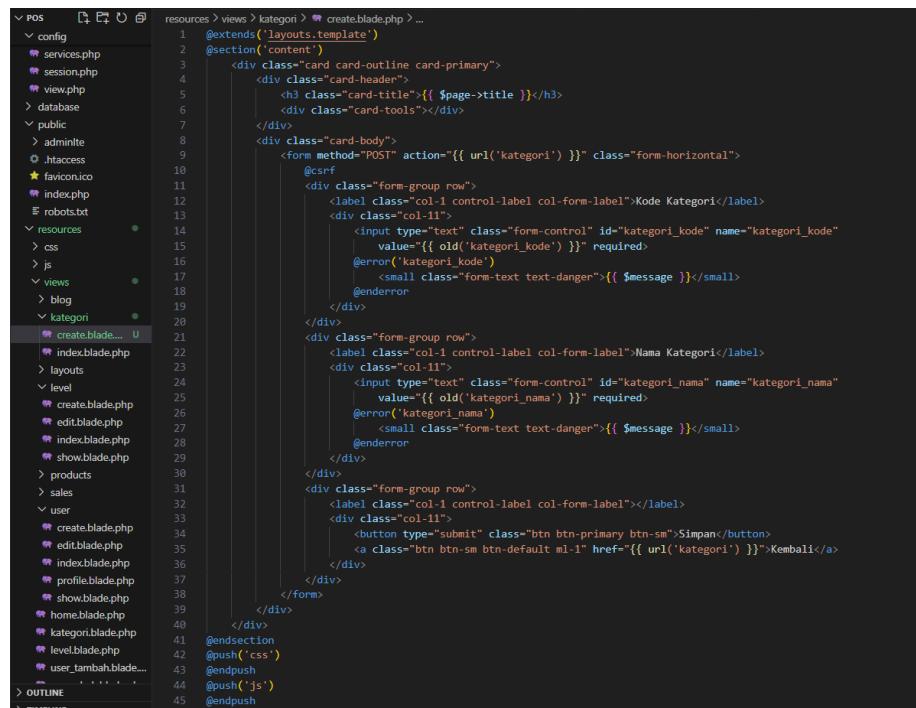
Berikut adalah tampilan awal halaman kategori ketika di akses.

ID	Kode Kategori	Nama Kategori	Aksi
1	fnb	Food and Beverage	Detail Edit Hapus
2	bnh	Beauty and Health	Detail Edit Hapus
3	bnk	Baby and Kid	Detail Edit Hapus
4	hnc	Home and Care	Detail Edit Hapus
5	elk	Eletronik	Detail Edit Hapus



6

Sekarang membuat form untuk menambah data, maka diperlukan file PWL/resources/views/kategori/create.blade.php seperti berikut.

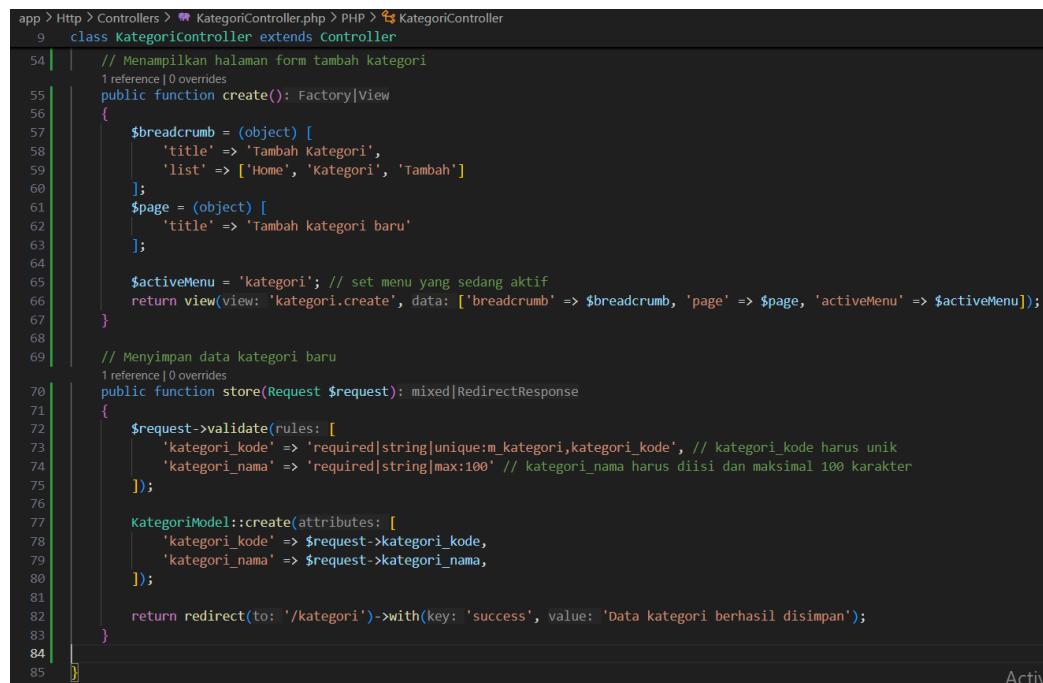


```
<?php

@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('kategori') }}" class="form-horizontal">
                @csrf
                <div class="form-group row">
                    <label class="col-1 control-label col-form-label">Kode Kategori</label>
                    <div class="col-11">
                        <input type="text" class="form-control" id="kategori_kode" name="kategori_kode" value="{{ old('kategori_kode') }}" required>
                        @error('kategori_kode')
                            <small class="form-text text-danger">{{ $message }}</small>
                        @enderror
                    </div>
                </div>
                <div class="form-group row">
                    <label class="col-1 control-label col-form-label">Nama Kategori</label>
                    <div class="col-11">
                        <input type="text" class="form-control" id="kategori_nama" name="kategori_nama" value="{{ old('kategori_nama') }}" required>
                        @error('kategori_nama')
                            <small class="form-text text-danger">{{ $message }}</small>
                        @enderror
                    </div>
                </div>
                <div class="form-group row">
                    <label class="col-1 control-label col-form-label"></label>
                    <div class="col-11">
                        <button type="submit" class="btn btn-primary btn-sm">Simpan</button>
                        <a class="btn btn-sm btn-default ml-1" href="{{ url('kategori') }}>Kembali</a>
                    </div>
                </div>
            </form>
        </div>
    </div>
    @endsection
    @push('css')
    @endpush
    @push('js')
    @endpush

```

Dan melakukan penambahan fungsi create() dan store() di file KategoriController.php dengan isi seperti berikut. Serta fungsi edit() dan update() untuk mengedit dan menyimpan perubahan data.



```
class KategoriController extends Controller
{
    // Menampilkan halaman form tambah kategori
    public function create(): Factory|View
    {
        $breadcrumb = (object) [
            'title' => 'Tambah Kategori',
            'list' => ['Home', 'Kategori', 'Tambah']
        ];
        $page = (object) [
            'title' => 'Tambah kategori baru'
        ];

        $activeMenu = 'kategori'; // set menu yang sedang aktif
        return view('kategori.create', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
    }

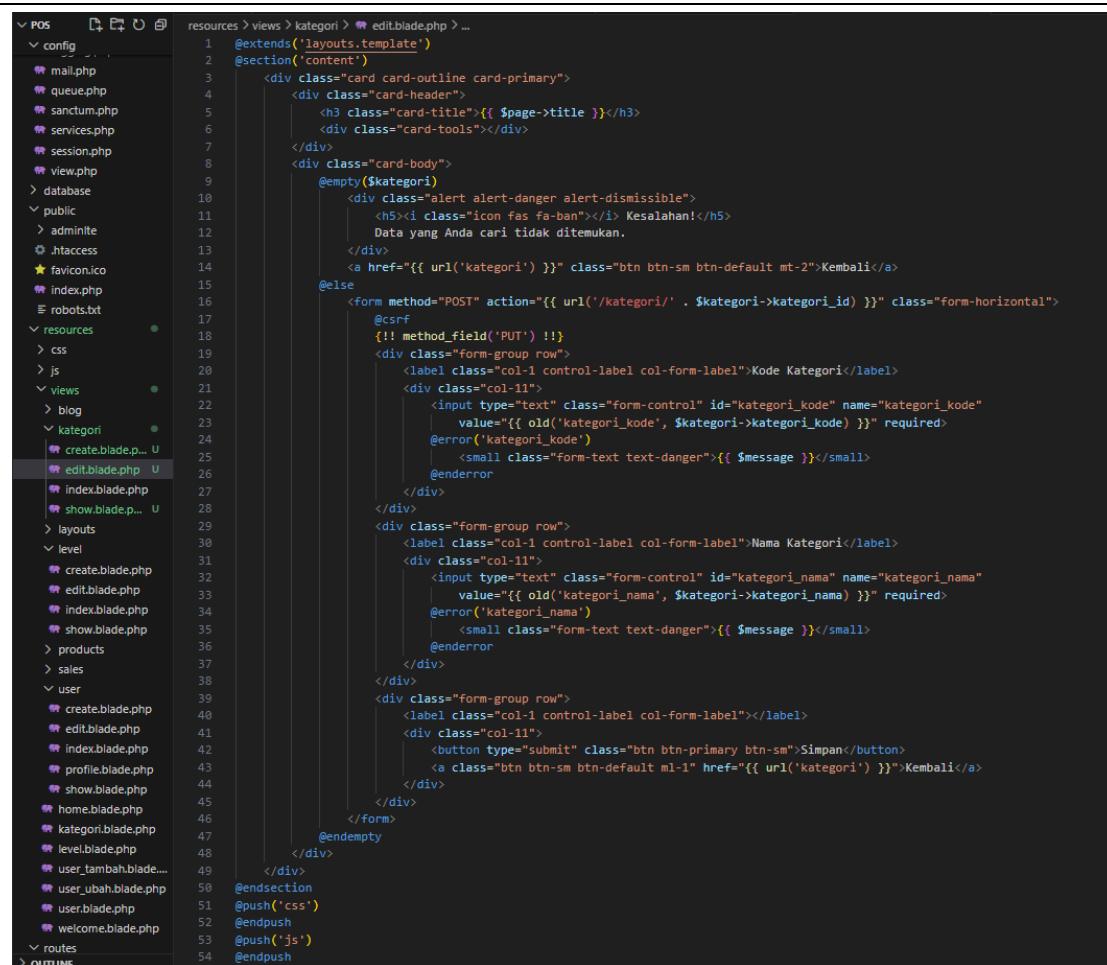
    // Menyimpan data kategori baru
    public function store(Request $request): mixed|RedirectResponse
    {
        $request->validate(rules: [
            'kategori_kode' => 'required|string|unique:kategori,kategori_kode', // kategori_kode harus unik
            'kategori_nama' => 'required|string|max:100' // kategori_nama harus diisi dan maksimal 100 karakter
        ]);

        KategoriModel::create(attributes: [
            'kategori_kode' => $request->kategori_kode,
            'kategori_nama' => $request->kategori_nama,
        ]);

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



	<pre>app > Http > Controllers > 🗃 KategoriController.php > PHP > 🗃 KategoriController > ⚙ update() 9 class KategoriController extends Controller 105 // Menampilkan halaman form edit kategori 106 1 reference overrides 107 public function edit(string \$id): Factory mixed RedirectResponse View 108 { 109 \$kategori = KategoriModel::find(id: \$id); 110 111 if (!\$kategori) { 112 return redirect(to: '/kategori')->with(key: 'error', value: 'Data kategori tidak ditemukan'); 113 } 114 115 \$breadcrumb = (object) [116 'title' => 'Edit Kategori', 117 'list' => ['Home', 'Kategori', 'Edit'] 118]; 119 120 \$page = (object) [121 'title' => 'Edit kategori' 122]; 123 124 \$activeMenu = 'kategori'; // set menu yang sedang aktif 125 return view(view: 'kategori.edit', data: ['breadcrumb' => \$breadcrumb, 'page' => \$page, 'kategori' => \$kategori, 'activeMenu' => \$activeMenu]); 126 } 127 128 // Menyimpan perubahan data kategori 129 1 reference overrides 130 public function update(Request \$request, string \$id): mixed RedirectResponse 131 { 132 \$request->validate(rules: [133 'kategori.kode' => 'required string unique:m_kategori,kategori.kode,' . \$id . ',kategori_id', 134 'kategori.nama' => 'required string max:100' 135]); 136 137 \$kategori = KategoriModel::find(id: \$id); 138 139 if (!\$kategori) { 140 return redirect(to: '/kategori')->with(key: 'error', value: 'Data kategori tidak ditemukan'); 141 } 142 143 \$kategori->update(attributes: [144 'kategori.kode' => \$request->kategori.kode, 145 'kategori.nama' => \$request->kategori.nama 146]); 147 148 return redirect(to: '/kategori')->with(key: 'success', value: 'Data kategori berhasil diubah'); 149 } </pre>
7	<p>The screenshot shows two windows side-by-side. On the left, a modal window titled 'Tambah Kategori' (Add Category) is displayed. It has two input fields: 'Kode Kategori' with the value 'sport' and 'Nama Kategori' with the value 'spor'. Below the inputs are two buttons: 'Simpan' (Save) and 'Batal' (Cancel). On the right, a larger window titled 'Daftar Kategori' (Category List) shows a table of categories. The table has columns for 'ID', 'Kode Kategori', 'Nama Kategori', and 'Aksi'. There are six rows, each with a green status indicator. The last row, which corresponds to the category added in the modal, has a status of 'Berhasil' (Success).</p>
8	<p>Gambar di atas adalah tampilan halaman penambahan Kategori yang mana hanya perlu memasukan kode kategori dan nama kategori, kemudian klik simpan untuk menyimpan penambahan data maka akan kembali ke halaman list daftar kategori yang akan menampilkan data terupdate dan keterangan hijau yang menunjukan data berhasil diinputkan</p> <p>Selanjutnya membuat form untuk mengedit data, maka diperlukan file PWL/resources/views/kategori/edit.blade.php seperti berikut.</p>

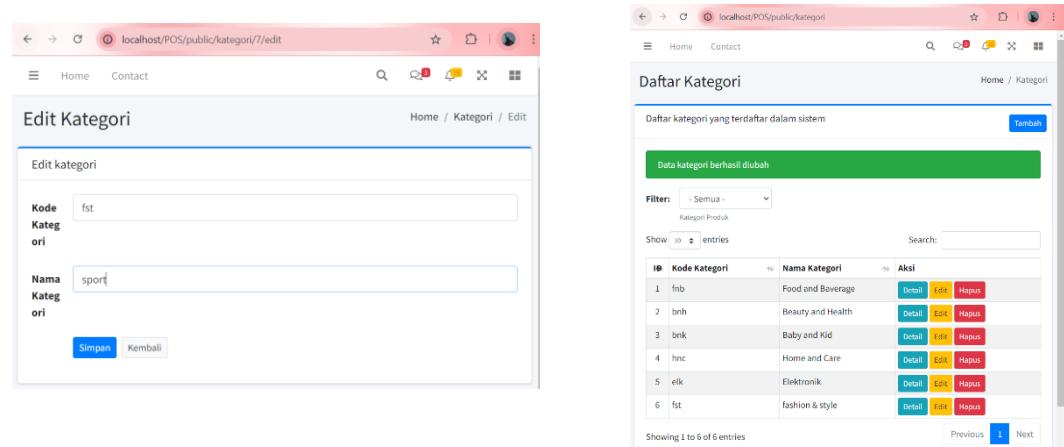


```
<?php

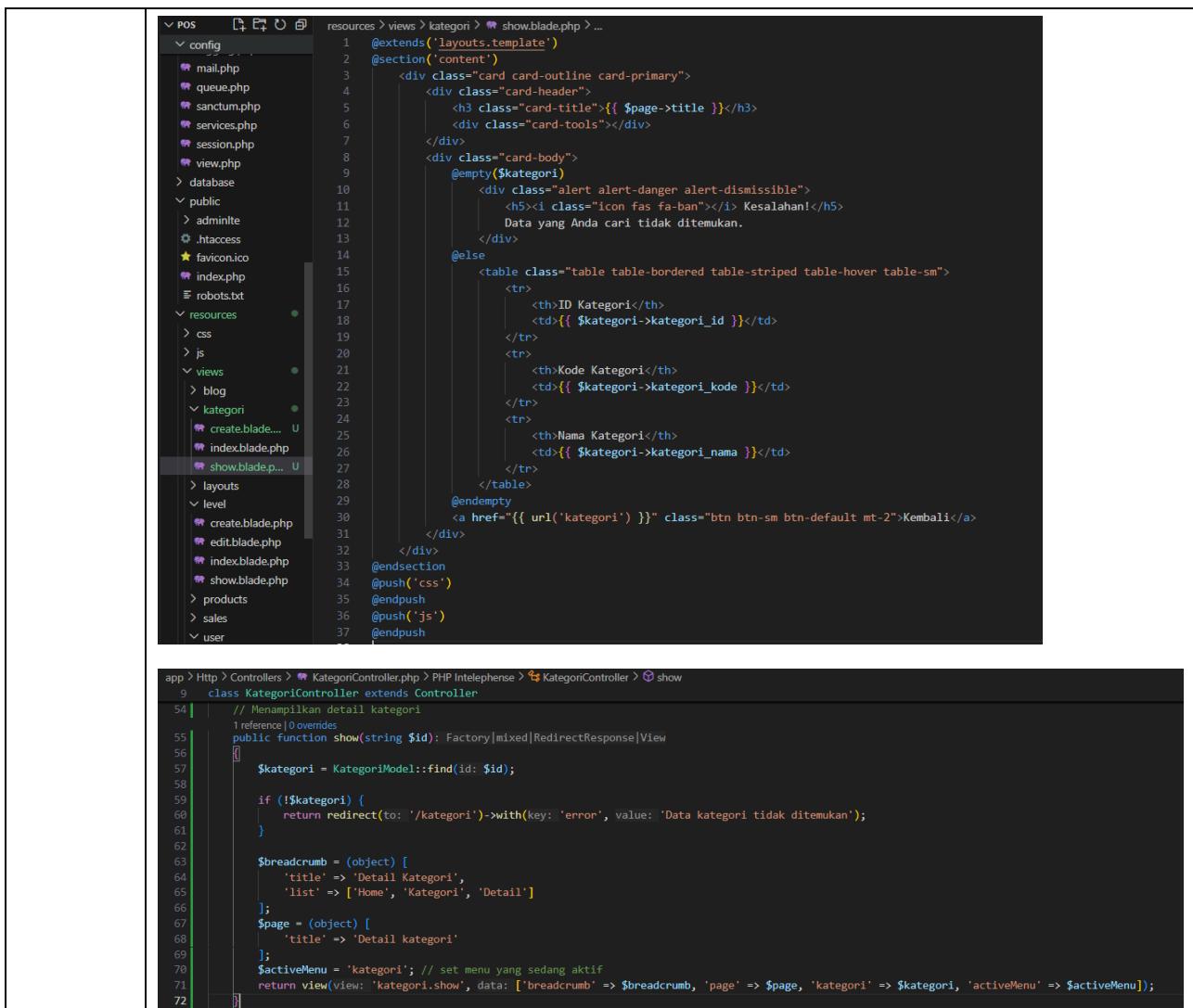
@extends('layouts.template')

@section('content')
    <div class="card card-outline card-primary">
        <div class="card-header">
            <h3 class="card-title">{{ $page->title }}</h3>
            <div class="card-tools"></div>
        </div>
        <div class="card-body">
            @empty($kategori)
                <div class="alert alert-danger alert-dismissible">
                    <h5><i class="icon fas fa-ban"></i> Kesalahan!</h5>
                    Data yang Anda cari tidak ditemukan.
                </div>
                <a href="{{ url('kategori') }}" class="btn btn-sm btn-default mt-2">Kembali</a>
            @else
                <form method="POST" action="{{ url('/kategori/' . $kategori->kategori_id) }}" class="form-horizontal">
                    @csrf
                    {!! method_field('PUT') !!}
                    <div class="form-group row">
                        <label class="col-1 control-label col-form-label">Kode Kategori</label>
                        <div class="col-11">
                            <input type="text" class="form-control" id="kategori_kode" name="kategori_kode" value="{{ old('kategori_kode', $kategori->kategori_kode) }}" required>
                            @error('kategori_kode')
                                <small class="form-text text-danger">{{ $message }}</small>
                            @enderror
                        </div>
                    </div>
                    <div class="form-group row">
                        <label class="col-1 control-label col-form-label">Nama Kategori</label>
                        <div class="col-11">
                            <input type="text" class="form-control" id="kategori_nama" name="kategori_nama" value="{{ old('kategori_nama', $kategori->kategori_nama) }}" required>
                            @error('kategori_nama')
                                <small class="form-text text-danger">{{ $message }}</small>
                            @enderror
                        </div>
                    </div>
                    <div class="form-group row">
                        <label class="col-1 control-label col-form-label"></label>
                        <div class="col-11">
                            <button type="submit" class="btn btn-primary btn-sm">Simpan</button>
                            <a class="btn btn-sm btn-default ml-1" href="{{ url('kategori') }}">Kembali</a>
                        </div>
                    </div>
                </form>
            @endempty
        </div>
    @endsection
    @push('css')
        @endpush
    @push('js')
        @endpush
</?php
```

Dan berikut adalah tampilan halaman edit kategori.



9	Untuk menampilkan detail data level user maka diperlukan file PWL/resources/views/kategori/show.blade.php seperti berikut. Dan diperlukan penambahan fungsi show() pada file KategoriController.php .
---	---

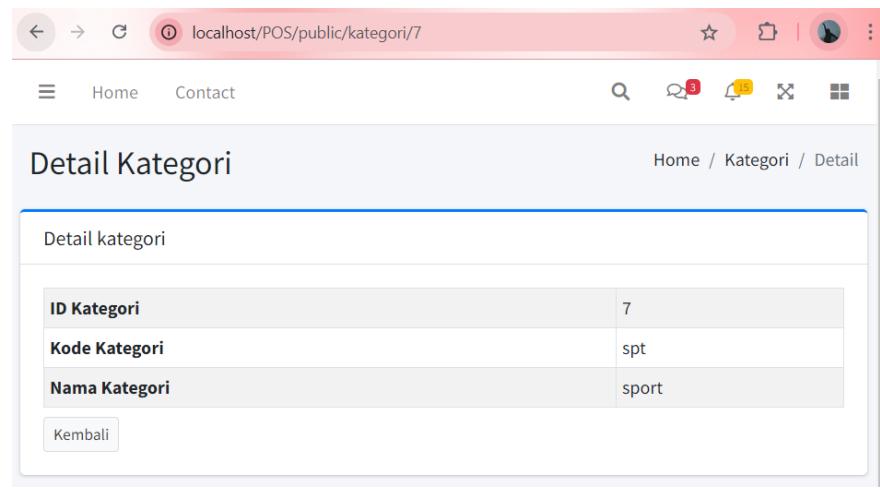


The screenshot shows two code editor panes. The top pane displays a Blade template file (`resources/views/kategori/show.blade.php`) with PHP code. The bottom pane displays the corresponding PHP controller code (`app/Http/Controllers/KategoriController.php`). Both files are related to displaying category details.

```
resources > views > kategori > show.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">{{ $page->title }}</h3>  
6             <div class="card-tools"></div>  
7         </div>  
8         <div class="card-body">  
9             @empty($kategori)  
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             @else  
15                 <table class="table table-bordered table-striped table-hover table-sm">  
16                     <tr>  
17                         <th>ID Kategori</th>  
18                         <td>{{ $kategori->kategori_id }}</td>  
19                     </tr>  
20                     <tr>  
21                         <th>Kode Kategori</th>  
22                         <td>{{ $kategori->kategori_kode }}</td>  
23                     </tr>  
24                     <tr>  
25                         <th>Nama Kategori</th>  
26                         <td>{{ $kategori->kategori_nama }}</td>  
27                     </tr>  
28                 </table>  
29             @endempty  
30             <a href="{{ url('kategori') }}" class="btn btn-sm btn-default mt-2">Kembali</a>  
31         </div>  
32     </div>  
33     @endsection  
34     @push('css')  
35     @endpush  
36     @push('js')  
37     @endpush
```

```
app > Http > Controllers > KategoriController.php > PHP Intelephense > KategoriController > show  
9 class KategoriController extends Controller  
41     // Menampilkan detail kategori  
42     1 reference | Overrides  
43     public function show(string $id): Factory|mixed|RedirectResponse|View  
44     {  
45         $kategori = KategoriModel::find(id: $id);  
46  
47         if (!$kategori) {  
48             return redirect(to: '/kategori')->with(key: 'error', value: 'Data kategori tidak ditemukan');  
49         }  
50  
51         $breadcrumb = (object) [  
52             'title' => 'Detail Kategori',  
53             'list' => [ 'Home', 'Kategori', 'Detail' ]  
54         ];  
55         $page = (object) [  
56             'title' => 'Detail kategori'  
57         ];  
58         $activeMenu = 'kategori'; // set menu yang sedang aktif  
59         return view(view: 'kategori.show', data: [ 'breadcrumb' => $breadcrumb, 'page' => $page, 'kategori' => $kategori, 'activeMenu' => $activeMenu ]);  
60     }
```

Berikut adalah tampilan halaman detail kategori barang yang mengambil data dari database m_kategori.

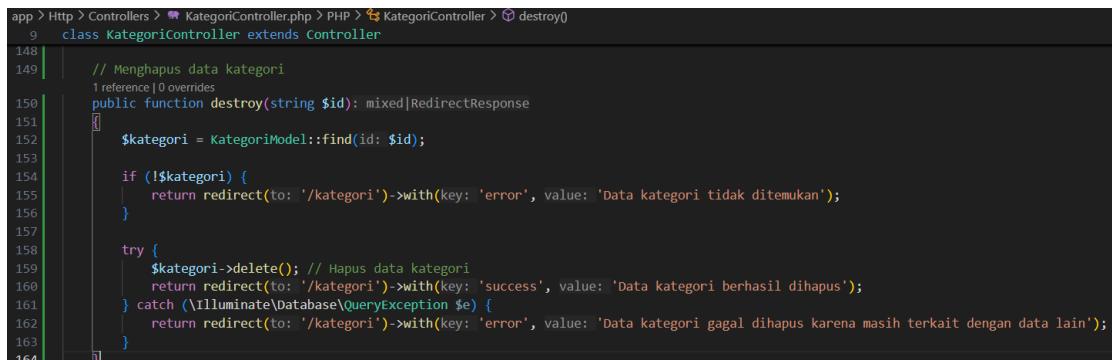
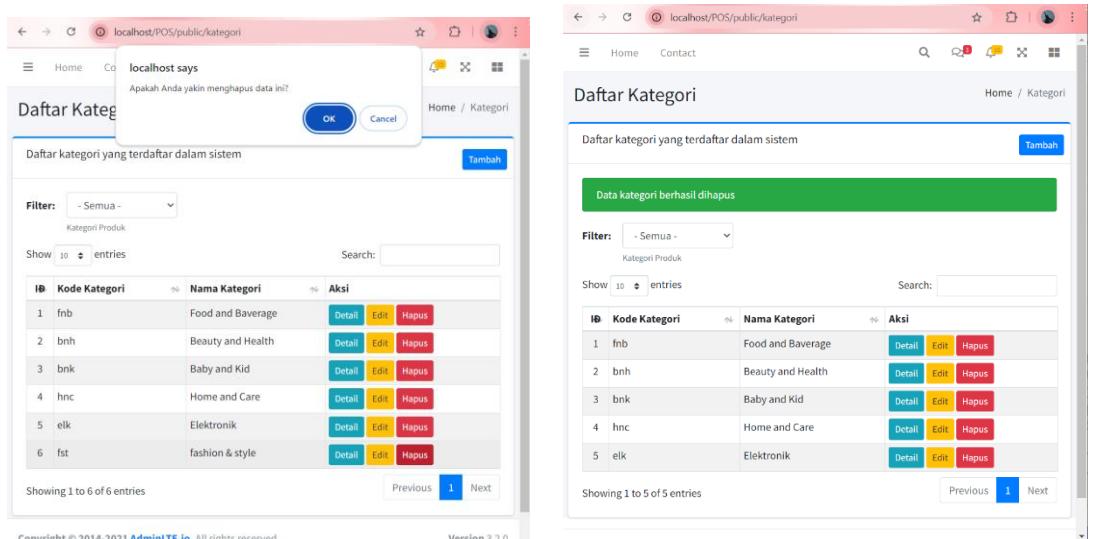


The screenshot shows a web browser window displaying a category detail page. The URL is `localhost/POS/public/kategori/7`. The page title is "Detail Kategori". The content area shows a table with the following data:

ID Kategori	7
Kode Kategori	spt
Nama Kategori	sport

Below the table is a "Kembali" button.



10	<p>Untuk melakukan penghapusan data maka diperlukan penambahan fungsi <code>destroy(string \$id)</code> pada file <code>KategoriController.php</code>.</p>  <p>The screenshot shows the <code>KategoriController.php</code> file with the <code>destroy()</code> method implemented. The code handles the deletion of a category based on its ID, displaying success or error messages to the user.</p>  <p>The first screenshot shows a modal dialog box with the text "Apakah Anda yakin menghapus data ini?" (Are you sure you want to delete the data?). The second screenshot shows the same table after one row has been deleted, with a green success message at the top: "Data kategori berhasil dihapus".</p> <p>Yang mana tampilan halaman nya seperti gambar diatas, yaitu ketika salah satu data dihapus maka akan terdapat pop up konfirmasi penghapusan dan keterangan yang menunjukan berhasil tidaknya data tersebut dihapus.</p>
----	--

C. Tabel m_supplier

Langkah	Jawaban/Deskripsi
1	Pertama saya melakukan penambahan pada route untuk menentukan tampilan menu apa saja yang akan ada terkait tabel <code>m_supplier</code> , seperti gambar di bawah ini yang saya samakan dengan menu lainnya dari praktikum di atas.



	<pre>routes > 🌐 web.php > ... 48 // TABEL SUPPLIER 49 Route::group(attributes: ['prefix' => 'stok'], routes: function () void { 50 Route::get(uri: '/', action: [SupplierController::class, 'index']); // menampilkan halaman awal supplier 51 Route::post(uri: '/list', action: [SupplierController::class, 'list']); // menampilkan data supplier dalam bentuk json untuk datatables 52 Route::get(uri: '/create', action: [SupplierController::class, 'create']); // menampilkan halaman form tambah supplier 53 Route::post(uri: '/', action: [SupplierController::class, 'store']); // menyimpan data supplier baru 54 Route::get(uri: '{id}', action: [SupplierController::class, 'show']); // menampilkan detail supplier 55 Route::get(uri: '{id}/edit', action: [SupplierController::class, 'edit']); // menampilkan halaman form edit supplier 56 Route::put(uri: '{id}', action: [SupplierController::class, 'update']); // menyimpan perubahan data supplier 57 Route::delete(uri: '{id}', action: [SupplierController::class, 'destroy']); // menghapus data supplier 58 }); 59 }); 60 </pre>
2	Kemudian Membuat file Model tentang m_supplier yang saya beri nama SupplierModel
3	Dilanjutkan membuat file SupplierController.php. Dan juga membuat fungsi index() dan list() untuk menampilkan halaman awal menu stok yang menampilkan data supplier. Serta membuat file stok.blade.php yang mana disini saya isi untuk data supplier.



```
resources > views > stok.blade.php > ...
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>Data Supplier</title>
5  </head>
6  <body>
7  |   <h1>Data Supplier</h1>
8  |   <table border="1" cellpadding="2" cellspacing="0">
9  |       <tr>
10 |           <th>ID Supplier</th>
11 |           <th>Kode Supplier</th>
12 |           <th>Nama Supplier</th>
13 |           <th>Alamat Supplier</th>
14 |       </tr>
15 |       @foreach ($data as $supplier)
16 |       <tr>
17 |           <td>{{ $supplier->supplier_id }}</td>
18 |           <td>{{ $supplier->supplier_kode }}</td>
19 |           <td>{{ $supplier->supplier_nama }}</td>
20 |           <td>{{ $supplier->supplier_alamat }}</td>
21 |       </tr>
22 |       @endforeach
23 |   </table>
24 </body>
25 </html>
```

4	Lalu membuat view pada PWL/resources/views/level/index.blade.php seperti berikut
---	--



```
bootstrap > view > site > indexIndex.php > @extends > @<--> <!--> <!-->
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">{{ $page->title }}</h3>
6              <div> class="card-tools">
7                  <a class="btn btn-sm btn-primary m-1" href="{{ url('site/create') }}> Tambah </a>
8              </div>
9          </div>
10         <div> class="card-body">
11             @if (session('success'))
12                 <div> class="alert alert-success">{{ session('success') }}</div>
13             @endif
14             @if (session('error'))
15                 <div> class="alert alert-danger">{{ session('error') }}</div>
16             @endif
17         </div>
18         <div> class="row">
19             <div> class="col-md-12">
20                 <div> class="form-group row">
21                     <label> class="col-1 control-label col-form-label" form="supplier_id">Filter:</label>
22                     <div> class="col-3">
23                         <select> class="form-control" id="supplier_id" name="supplier_id" required>
24                             option value=""> Semua --</option>
25                             @foreach($supplier as $Supplier)
26                                 option value="{{ $Supplier->supplier_id }}>{{ $Supplier->supplier_name }}</option>
27                             @endforeach
28                         </select>
29                         <small> class="form-text text-muted">Nama Supplier</small>
30                     </div>
31                 </div>
32             </div>
33         </div>
34         <table> class="table table-bordered table-striped table-hover table-sm" id="table_supplier">
35             <thead>
36                 <tr>
37                     <th> ID </th>
38                     <th> Kode Supplier </th>
39                     <th> Nama Supplier </th>
40                     <th> Alamat Supplier </th>
41                     <th> Aksi </th>
42                 </tr>
43             </thead>
44             </table>
45         </div>
46     </div>
47     <script>
48         $(`#supplier_id`)>
49         <script>
50             $(document).ready(function() {
51                 var dataSupplier = $('#table_supplier').DataTable({
52                     serverSide: true,
53                     ajax: {
54                         url: "{{ url('site/list') }}",
55                         dataType: "json",
56                         type: "POST",
57                         data: function(d) {
58                             d.supplier_id = $('#supplier_id').val();
59                         }
60                     },
61                     columns: [
62                         {
63                             data: "DT_RowIndex",
64                             className: "text-center",
65                             orderable: false,
66                             searchable: false
67                         },
68                         {
69                             data: "supplier_code",
70                             className: "",
71                             orderable: true,
72                             searchable: true
73                         },
74                         {
75                             data: "supplier_name",
76                             className: "",
77                             orderable: true,
78                             searchable: true
79                         },
80                         {
81                             data: "supplier_alamat",
82                             className: "",
83                             orderable: true,
84                             searchable: true
85                         },
86                         {
87                             data: "aksi",
88                             className: "",
89                             orderable: false,
90                             searchable: false
91                         }
92                     ]
93                 });
94                 $('#supplier_id').on('change', function(){
95                     dataSupplier.ajax.reload();
96                 });
97             });
98         </script>
99     </script>
100 </body>
```

5

Berikut adalah tampilan awal halaman supplier ketika di akses.



- 6 Sekarang membuat form untuk menambah data, maka diperlukan file PWL/resources/views/stok/create.blade.php seperti berikut.

```
resources > views > stok > create.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>
7          <div class="card-body">
8              <form method="POST" action="{{ url('stok') }}" class="form-horizontal">
9                  <csrf>
10                 <div class="form-group row">
11                     <label class="col-1 control-label col-form-label>Kode Supplier</label>
12                     <div class="col-11">
13                         <input type="text" class="form-control" id="supplier_kode" name="supplier_kode"
14                             value="{{ old('supplier_kode') }}" required>
15                         @error('supplier_kode')
16                             <small class="form-text text-danger">{{ $message }}</small>
17                         @enderror
18                     </div>
19                 </div>
20                 <div class="form-group row">
21                     <label class="col-1 control-label col-form-label>Nama Supplier</label>
22                     <div class="col-11">
23                         <input type="text" class="form-control" id="supplier_nama" name="supplier_nama"
24                             value="{{ old('supplier_nama') }}" required>
25                         @error('supplier_nama')
26                             <small class="form-text text-danger">{{ $message }}</small>
27                         @enderror
28                     </div>
29                 </div>
30                 <div class="form-group row">
31                     <label class="col-1 control-label col-form-label>Alamat Supplier</label>
32                     <div class="col-11">
33                         <input type="text" class="form-control" id="supplier_alamat" name="supplier_alamat"
34                             value="{{ old('supplier_alamat') }}" required>
35                         @error('supplier_alamat')
36                             <small class="form-text text-danger">{{ $message }}</small>
37                         @enderror
38                     </div>
39                 </div>
40                 <div class="form-group row">
41                     <label class="col-1 control-label col-form-label></label>
42                     <div class="col-11">
43                         <button type="submit" class="btn btn-primary btn-sm">Simpan</button>
44                         <a class="btn btn-sm btn-default ml-1" href="{{ url('stok') }}>Kembali</a>
45                     </div>
46                 </div>
47             </form>
48         </div>
49     </div>
50     @endsection
51     @endpush
52     @push('css')
53     @push('js')
54     @endpush
55     @endpush
```

Dan melakukan penambahan fungsi create() dan store() di file KategoriController.php dengan isi seperti berikut. Serta fungsi edit() dan update() untuk mengedit dan menyimpan perubahan data.



```
app > Http > Controllers > 🗂 SupplierController.php > PHP > 📄 SupplierController > ⚑ store()
10   class SupplierController extends Controller
11
12   // Menampilkan halaman form tambah supplier
13   1 reference | 0 overrides
14   public function create(): Factory|View
15   {
16       $breadcrumb = (object) [
17           'title' => 'Tambah Supplier',
18           'list' => ['Home', 'Supplier', 'Tambah']
19       ];
20       $page = (object) [
21           'title' => 'Tambah supplier baru'
22       ];
23
24       $activeMenu = 'supplier'; // set menu yang sedang aktif
25       return view(view: '_stok.create', data: ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
26   }
27
28   // Menyimpan data supplier baru
29   1 reference | 0 overrides
30   public function store(Request $request): mixed|RedirectResponse
31   {
32       $request->validate(rules: [
33           'supplier_kode' => 'required|string|unique:m_supplier,supplier_kode', // supplier_kode harus unik
34           'supplier_nama' => 'required|string|max:100', // supplier_nama harus diisi dan maksimal 100 karakter
35           'supplier_alamat' => 'nullable|string|max:255' // supplier_alamat opsional
36       ]);
37
38       StokModel::create(attributes: [
39           'supplier_kode' => $request->supplier_kode,
40           'supplier_nama' => $request->supplier_nama,
41           'supplier_alamat' => $request->supplier_alamat,
42       ]);
43
44       return redirect(to: '/stok')->with(key: 'success', value: 'Data supplier berhasil disimpan');
45   }
46
47   ?>
```



```
app > Http > Controllers > 🗂 SupplierController.php > PHP > 📄 SupplierController > ⚑ update()
10   class SupplierController extends Controller
11
12   // Menampilkan halaman form edit supplier
13   1 reference | 0 overrides
14   public function edit(string $id): Factory|mixed|RedirectResponse|View
15   {
16       $supplier = StokModel::find(id: $id);
17
18       if (!$supplier) {
19           return redirect(to: '/stok')->with(key: 'error', value: 'Data supplier tidak ditemukan');
20       }
21
22       $breadcrumb = (object) [
23           'title' => 'Edit Supplier',
24           'list' => ['Home', 'Supplier', 'Edit']
25       ];
26
27       $page = (object) [
28           'title' => 'Edit supplier'
29       ];
30
31       $activeMenu = 'supplier'; // set menu yang sedang aktif
32       return view(view: '_stok.edit', data: ['breadcrumb' => $breadcrumb, 'page' => $page, 'supplier' => $supplier, 'activeMenu' => $activeMenu]);
33   }
34
35   // Menyimpan perubahan data supplier
36   1 reference | 0 overrides
37   public function update(Request $request, string $id): mixed|RedirectResponse
38   {
39       $request->validate(rules: [
40           'supplier_kode' => 'required|string|unique:m_supplier,supplier_kode,' . $id . ',supplier_id',
41           'supplier_nama' => 'required|string|max:100',
42           'supplier_alamat' => 'nullable|string|max:255'
43       ]);
44
45       $supplier = StokModel::find(id: $id);
46
47       if (!$supplier) {
48           return redirect(to: '/stok')->with(key: 'error', value: 'Data supplier tidak ditemukan');
49       }
50
51       $supplier->update(attributes: [
52           'supplier_kode' => $request->supplier_kode,
53           'supplier_nama' => $request->supplier_nama,
54           'supplier_alamat' => $request->supplier_alamat
55       ]);
56
57       return redirect(to: '/stok')->with(key: 'success', value: 'Data supplier berhasil diubah');
58   }
59
60   ?>
```



7	<p>The screenshot shows two browser windows. The left window displays the 'Tambah Supplier' (Add Supplier) form with fields for 'Kode Supplier' (kode supplier), 'Nama Supplier' (Supplier Name), and 'Alamat Supplier' (Supplier Address). The right window shows the 'Daftar Supplier' (Supplier List) page with a table of suppliers, each with 'Detail', 'Edit', and 'Hapus' (Delete) buttons.</p> <table border="1"><thead><tr><th>ID</th><th>Kode Supplier</th><th>Nama Supplier</th><th>Alamat Supplier</th><th>Aksi</th></tr></thead><tbody><tr><td>1</td><td>yol</td><td>Yolanda Supplier</td><td>Jln Hamidrusdi No.102</td><td>[Detail] [Edit] [Hapus]</td></tr><tr><td>2</td><td>snd</td><td>Sendi Supplier</td><td>Jln Buring No.54</td><td>[Detail] [Edit] [Hapus]</td></tr><tr><td>3</td><td>ily</td><td>Liya Supplier</td><td>Jln Tuban no 65</td><td>[Detail] [Edit] [Hapus]</td></tr><tr><td>4</td><td>nels</td><td>Nela Supplier</td><td>Jln jerusalem No.4</td><td>[Detail] [Edit] [Hapus]</td></tr></tbody></table>	ID	Kode Supplier	Nama Supplier	Alamat Supplier	Aksi	1	yol	Yolanda Supplier	Jln Hamidrusdi No.102	[Detail] [Edit] [Hapus]	2	snd	Sendi Supplier	Jln Buring No.54	[Detail] [Edit] [Hapus]	3	ily	Liya Supplier	Jln Tuban no 65	[Detail] [Edit] [Hapus]	4	nels	Nela Supplier	Jln jerusalem No.4	[Detail] [Edit] [Hapus]
ID	Kode Supplier	Nama Supplier	Alamat Supplier	Aksi																						
1	yol	Yolanda Supplier	Jln Hamidrusdi No.102	[Detail] [Edit] [Hapus]																						
2	snd	Sendi Supplier	Jln Buring No.54	[Detail] [Edit] [Hapus]																						
3	ily	Liya Supplier	Jln Tuban no 65	[Detail] [Edit] [Hapus]																						
4	nels	Nela Supplier	Jln jerusalem No.4	[Detail] [Edit] [Hapus]																						
8	<p>Selanjutnya membuat form untuk mengedit data, maka diperlukan file PWL/resources/views/kategori/edit.blade.php seperti berikut.</p> <pre>resources > views > stok > edit.blade.php > div.card.card-outline.card-primary > div.card-body 1 #extends('layouts.template') 2 @section('content') 3 <div class="card card-outline card-primary"> 4 <div class="card-header"> 5 <h3>{{ \$page_title }}</h3> 6 <div class="card-tools"></div> 7 </div> 8 <div class="card-body"> 9 @empty(\$supplier) 10 <div class="alert alert-danger alert-dismissible"> 11 <h5> Kesalahan!</h5> 12 Data yang Anda cari tidak ditemukan. 13 </div> 14 Kembali 15 @else 16 <form method="POST" action="{{ url('stok') . \$supplier->supplier_id }} " class="form-horizontal"> 17 @csrf 18 @if(\$supplier->supplier_id != null) 19 <div class="form-group row"> 20 <label class="col-1 control-label col-form-label">Kode Supplier</label> 21 <div class="col-11"> 22 <input type="text" class="form-control" id="supplier_kode" name="supplier_kode" value="{{ old('supplier_kode', \$supplier->supplier_kode) }}" required> 23 @error('supplier_kode') 24 <small class="form-text text-danger" style="color: red;">{{ \$message }}{{ \$message }}{{ \$message }}</pre>																									



Dan berikut adalah tampilan halaman edit kategori.

- 9 Untuk menampilkan detail data level user maka diperlukan file PWL/resources/views/stok/show.blade.php seperti berikut. Dan diperlukan penambahan fungsi show() pada file SupplierController.php.

```
resources > views > stok > 🗂 show.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">{ $page->title }</h3>
6              <div class="card-tools"></div>
7          </div>
8          <div class="card-body">
9              @empty($supplier)
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 Supplier</th>
18                         <td>{{ $supplier->supplier_id }}</td>
19                     </tr>
20                     <tr>
21                         <th>Kode Supplier</th>
22                         <td>{{ $supplier->supplier_kode }}</td>
23                     </tr>
24                     <tr>
25                         <th>Nama Supplier</th>
26                         <td>{{ $supplier->supplier_nama }}</td>
27                     </tr>
28                     <tr>
29                         <th>Alamat Supplier</th>
30                         <td>{{ $supplier->supplier_alamat }}</td>
31                     </tr>
32                 </table>
33             @endempty
34             <a href="{{ url('stok') }}" class="btn btn-sm btn-default mt-2">Kembali</a>
35         </div>
36     </div>
37     @endsection
38     @push('css')
39     @endpush
40     @push('js')
41     @endpush
42     
```



```
app > Http > Controllers > SupplierController.php > PHP > SupplierController > show()
10 class SupplierController extends Controller
11 {
12     // Menampilkan detail supplier
13     1 reference|0 overrides
14     public function show(string $id): Factory|mixed|RedirectResponse|View
15     {
16         $supplier = StokModel::find(id: $id);
17
18         if (!$supplier) {
19             return redirect(to: '/stok')->with(key: 'error', value: 'Data supplier tidak ditemukan');
20         }
21
22         $breadcrumb = (object) [
23             'title' => 'Detail supplier',
24             'list' => ['Home', 'Supplier', 'Detail']
25         ];
26         $page = (object) [
27             'title' => 'Detail supplier'
28         ];
29         $activeMenu = 'stok'; // set menu yang sedang aktif
30         return view(view: 'stok.show', data: [$breadcrumb, $page => $page, 'supplier' => $supplier, 'activeMenu' => $activeMenu]);
31     }
32 }
```

Berikut adalah tampilan halaman detail supplier yang mengambil data dari database m_supplier.

localhost/POS/public/stok/1

Home Contact

Detail Supplier

Home / Supplier / Detail

Detail supplier	
ID Supplier	1
Kode Supplier	yol
Nama Supplier	Yolanda Supplier
Alamat Supplier	Jln Hamidrusdi No.102

Kembali

- 10 Untuk melakukan penghapusan data maka diperlukan penambahan fungsi `destroy(string $id)` pada file `KategoriController.php`.

```
app > Http > Controllers > SupplierController.php > PHP > SupplierController > destroy()
10 class SupplierController extends Controller
11 {
12     // Menghapus data supplier
13     1 reference|0 overrides
14     public function destroy(string $id): mixed|RedirectResponse
15     {
16         $supplier = StokModel::find(id: $id);
17
18         if (!$supplier) {
19             return redirect(to: '/stok')->with(key: 'error', value: 'Data supplier tidak ditemukan');
20         }
21
22         try {
23             $supplier->delete(); // Hapus data supplier
24             return redirect(to: '/stok')->with(key: 'success', value: 'Data supplier berhasil dihapus');
25         } catch (\Illuminate\Database\QueryException $e) {
26             return redirect(to: '/stok')->with(key: 'error', value: 'Data supplier gagal dihapus karena masih terkait dengan data lain');
27         }
28     }
29 }
```



Yang mana tampilan halaman nya seperti gambar diatas, yaitu ketika salah satu data dihapus maka akan terdapat pop up konfirmasi penghapusan dan keterangan yang menunjukan berhasil tidaknya data tersebut dihapus.

D. Tabel m_barang

Langkah	Jawaban/Deskripsi
1	Pertama saya melakukan penambahan pada route untuk menentukan tampilan menu apa saja yang akan ada terkait tabel m_barang, seperti gambar di bawah ini yang saya samakan dengan menu lainnya dari praktikum di atas. <pre>routes > web.php > ... 62 // TABEL BARANG 63 Route::group(attributes: ['prefix' => 'barang'], routes: function () { 64 Route::get('/index', action: [BarangController::class, 'index']); // menampilkan halaman awal supplier 65 Route::post('/list', action: [BarangController::class, 'list']); // menampilkan data supplier dalam bentuk json untuk datatables 66 Route::get('/create', action: [BarangController::class, 'create']); // menampilkan halaman form tambah supplier 67 Route::post('/store', action: [BarangController::class, 'store']); // menyimpan data supplier baru 68 Route::get('{id}', action: [BarangController::class, 'show']); // menampilkan detail supplier 69 Route::get('{id}/edit', action: [BarangController::class, 'edit']); // menampilkan halaman form edit supplier 70 Route::put('{id}', action: [BarangController::class, 'update']); // menyimpan perubahan data supplier 71 Route::delete('{id}', action: [BarangController::class, 'destroy']); // menghapus data supplier 72 }); 73 }</pre>
2	Kemudian Membuat file Model tentang m_barang yang saya beri nama BarangModel



	<pre>app > Models > BarangModel.php > ... 1 <?php 2 namespace App\Models; 3 use Illuminate\Database\Eloquent\Factories\HasFactory; 4 use Illuminate\Database\Eloquent\Model; 5 class BarangModel extends Model 6 { 7 use HasFactory; 8 protected \$table = 'm_barang'; // Mendefinisikan nama tabel yang digunakan oleh model ini 9 protected \$primaryKey = 'barang_kode'; // Mendefinisikan primary key dari tabel yang digunakan 10 protected \$fillable = ['barang_kode', 'kategori_id', 'barang_nama', 'harga_beli', 'harga_jual']; // Mendefinisikan kolom yang dapat diisi secara massal 11 }</pre>
3	Dilanjutkan membuat file BarangController.php. Dan juga membuat fungsi index() dan list() untuk menampilkan halaman awal barang yang menampilkan data barang. Serta membuat file barang.blade.php yang mana disini saya isi untuk data barang.
	<pre>app > Http > Controllers > BarangController.php > PHP Intelephense > BarangController > index 1 <?php 2 namespace App\Http\Controllers; 3 use App\Models\BarangModel; 4 use Illuminate\Http\Request; 5 use Yajra\Datatables\Facades\Datatables; 6 class BarangController extends Controller 7 { 8 // Menampilkan halaman daftar barang 9 public function index(): Factory\View 10 (11 \$breadcrumb = (object) [12 'title' => 'Daftar Barang', 13 'list' => ['Home', 'Barang'] 14]; 15 \$page = (object) [16 'title' => 'Daftar barang yang terdaftar dalam sistem' 17]; 18 \$activeMenu = 'barang'; // set menu yang sedang aktif 19 \$barang = BarangModel::all(); // ambil data barang untuk filter 20 return view('barang.index', data: ['breadcrumb' => \$breadcrumb, 'page' => \$page, 'barang' => \$barang, 'activeMenu' => \$activeMenu]); 21 } 22 // Ambil data barang dalam bentuk JSON untuk datatables 23 public function list(Request \$request): JsonResponse 24 (25 \$barangs = BarangModel::select(columns: 'barang_id', 'kategori_id', 'barang_kode', 'barang_nama', 'harga_beli', 'harga_jual'); 26 // Filter data barang berdasarkan barang_id 27 if (\$request->barang_id) { 28 \$barangs->where(column: 'barang_id', operator: \$request->barang_id); 29 } 30 return Datatables::of(\$barangs) 31 ->addIndexColumn() 32 ->addColumn('aksi', function (\$barang): string { // menambahkan kolom aksi 33 \$btn = 'barang_id) . '" class="btn btn-info btn-sm">Detail '; 34 \$btn .= 'barang_id) . '" class="btn btn-warning btn-sm">Edit '; 35 \$btn .= '<form class="d-inline-block" method="POST" action="'; 36 url(path: '/barang/' . \$barang->barang_id) . '>' . csrf_field() . method_field(method: "DELETE") . '</form> '; 37 \$button = '<button type="submit" class="btn btn-danger btn-sm" onclick="return confirm(\'Apakah Anda yakin menghapus data ini?\')>Hapus</button>'; 38 return \$btn; 39 }) 40 ->rawColumn(['aksi']) // memberitahu bahwa kolom aksi adalah HTML 41 ->make(true); 42 } 43 }</pre> <p style="text-align: right;">Activate Window Go to Settings to...</p> <pre>resources > views > barang.blade.php > ... 1 <!DOCTYPE html> 2 <html> 3 <head> 4 <title>Data Barang</title> 5 </head> 6 <body> 7 <h1>Data Barang</h1> 8 <table border="1" cellpadding="2" cellspacing="0"> 9 <tr> 10 <th>ID Barang</th> 11 <th>Kode Barang</th> 12 <th>Nama Barang</th> 13 <th>Harga Beli</th> 14 <th>Harga Jual</th> 15 <th>Kode Kategori</th> 16 </tr> 17 @foreach (\$data as \$barang) 18 <tr> 19 <td>{{ \$barang->barang_id }}</td> 20 <td>{{ \$barang->barang_kode }}</td> 21 <td>{{ \$barang->barang_nama }}</td> 22 <td>{{ number_format(\$barang->harga_beli, 2, ',', '.') }}</td> 23 <td>{{ number_format(\$barang->harga_jual, 2, ',', '.') }}</td> 24 <td>{{ \$barang->kategori_kode }}</td> 25 </tr> 26 @endforeach 27 </table> 28 </body> 29 </html></pre>



4

Lalu membuat view pada PWL/resources/views/level/index.blade.php seperti berikut

```
resources > views > barang > index.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 }}{{ session('success') }}{{ session('error') }}Barang Produk</small>
29        </div>
30      </div>
31    </div>
32    <table class="table table-bordered table-striped table-hover table-sm" id="table_barang">
33      <thead>
34        <tr>
35          <th>ID</th>
36          <th>Kode Barang</th>
37          <th>Nama Barang</th>
38          <th>Harga Beli</th>
39          <th>Harga Jual</th>
40          <th>Kode Kategori</th> <!-- Kolom untuk Kode Kategori -->
41          <th>Aksi</th>
42        </tr>
43      </thead>
44      <tbody>
45        <!-- Data akan diisi menggunakan DataTables -->
46      </tbody>
47    </table>
48  </div>
49 </div>
50 <script>
51 $(document).ready(function() {
52   var databarang = $('#table_barang').DataTable({
53     serverSide: true,
54     ajax: {
55       url: "{{ url('barang/list') }}",
56       dataType: "json",
57       type: "POST",
58       data: function(d) {
59         d.barang_id = $('#barang_id').val();
60       }
61     },
62     columns: [
63       {
64         data: "DT_RowIndex",
65         className: "text-center",
66         orderable: false,
67         searchable: false
68       },
69       {
70         data: "barang_kode",
71         className: "text-center",
72         orderable: true,
73         searchable: true
74       },
75       {
76         data: "barang_nama",
77         className: "",
78         orderable: true,
79         searchable: true
80       },
81       {
82         data: "harga_beli",
83         className: "text-right",
84         orderable: true,
85         searchable: true
86       },
87       {
88         data: "harga_jual",
89         className: "text-right",
90         orderable: true,
91         searchable: true
92       },
93       {
94         data: "kategori_id", // Kolom untuk Kode Kategori
95         className: "",
96         orderable: true,
97         searchable: true
98       },
99       {
100         data: "aksi",
101         className: "",
102         orderable: false,
103         searchable: false
104       }
105     ],
106     $('#barang_id').on('change', function(){
107       databarang.ajax.reload();
108     });
109   });
110 });
111 </script>
112 </div>
113 </body>
114 </html>
```



5

Berikut adalah tampilan awal halaman barang ketika di akses.

Daftar Barang						
Daftar barang yang terdaftar dalam sistem						
Filter: - Semua -						
Show 10 entries						
ID	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kode Kategori	Aksi
1	0	Nasi	10000	12000	1	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
2	0	Ayam	8000	10000	1	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
3	0	Sawi	5000	7000	1	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
4	0	Promag	8000	10000	2	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
5	0	Sunscreen	12000	15000	2	<button>Detail</button> <button>Edit</button> <button>Hapus</button>

Daftar Barang						
Daftar barang yang terdaftar dalam sistem						
Filter: - Semua -						
Show 10 entries						
ID	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kode Kategori	Aksi
1	0	Nasi	10000	12000	1	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
2	0	Ayam	8000	10000	1	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
3	0	Sawi	5000	7000	1	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
4	0	Promag	8000	10000	2	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
5	0	Sunscreen	12000	15000	2	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
6	0	Dulcolax	8000	10000	2	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
7	0	Minyak Telon	15000	19000	3	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
8	0	Popok Bayi	8000	20000	3	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
9	0	Bedak Bayi	8000	10000	3	<button>Detail</button> <button>Edit</button> <button>Hapus</button>

6

Sekarang membuat form untuk menambah data, maka diperlukan file PWL/resources/views/barang/create.blade.php seperti berikut.



```
resources > views > barang > create.blade.php > div.card.card-outline.card-primary > div.card-body > form.form-horizontal > div.form-group.row > div.col-11
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">{{ $page->title }}</h3>
7              <div class="card-tools"></div>
8          </div>
9          <div class="card-body">
10             <form method="POST" action="{{ url('barang') }}" class="form-horizontal">
11                 @csrf
12                 <div class="form-group row">
13                     <label class="col-1 control-label col-form-label">kategori</label>
14                     <div class="col-11">
15                         <select class="form-control" id="kategori_id" name="kategori_id" required>
16                             <option value=""> Pilih kategori </option>
17                             @foreach ($kategori as $item)
18                                 <option value="{{ $item->kategori_id }}>{{ $item->kategori_nama }}</option>
19                             @endforeach
20                         </select>
21                         @error('kategori_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">Kode barang</label>
28                     <div class="col-11">
29                         <input type="text" class="form-control" id="barang_kode" name="barang_kode"
30                         value="{{ old('barang_kode') }}" required>
31                         @error('barang_kode')
32                             <small class="form-text text-danger">{{ $message }}</small>
33                         @enderror
34                     </div>
35                 </div>
36                 <div class="form-group row">
37                     <label class="col-1 control-label col-form-label">Nama barang</label>
38                     <div class="col-11">
39                         <input type="text" class="form-control" id="baranag_nama" name="barang_nama" value="{{ old('barang_nama') }}"
40                         required>
41                         @error('barang_nama')
42                             <small class="form-text text-danger">{{ $message }}</small>
43                         @enderror
44                     </div>
45                 </div>
46                 <div class="form-group row">
47                     <label class="col-1 control-label col-form-label">Harga jual</label>
48                     <div class="col-11">
49                         <input type="number" class="form-control" id="harga_jual" name="harga_jual" required>
50                         @error('harga_jual')
51                             <small class="form-text text-danger">{{ $message }}</small>
52                         @enderror
53                     </div>
54                 </div>
55                 <div class="form-group row">
56                     <label class="col-1 control-label col-form-label">Harga beli</label>
57                     <div class="col-11">
58                         <input type="number" class="form-control" id="harga_beli" name="harga_beli" required>
59                         @error('harga_beli')
60                             <small class="form-text text-danger">{{ $message }}</small>
61                         @enderror
62                     </div>
63                 </div>
64                 <div class="form-group row">
65                     <label class="col-1 control-label col-form-label"></label>
66                     <div class="col-11">
67                         <button type="submit" class="btn btn-primary btn-sm">Simpan</button>
68                         <a class="btn btn-sm btn-default ml-1" href="{{ url('barang') }}>Kembali</a>
69                     </div>
70                 </div>
71             </form>
72         </div>
73     </div>
74     @endsection
75     @push('css')
76     @endpush
77     @push('js')
78     @endpush
```

Dan melakukan penambahan fungsi `create()` dan `store()` di file `KategoriController.php` dengan isi seperti berikut. Serta fungsi `edit()` dan `update()` untuk mengedit dan menyimpan perubahan data.



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141
Telp. (0341) 404424 - 404425 Fax. (0341) 404420

Telp. (0341) 404424 – 404425, Fax (0341) 404420
<http://iti.polinama.ac.id>

<https://jti.polinema.ac.id>

```
app > Http > Controllers > BarangController.php > PHP > BarangController > [object]
6   class BarangController extends Controller
65  // Menampilkan halaman form tambah barang
66  | reference|0 overrides
67  public function create(): Factory|View
68  {
69    $breadcrumb = (object) [
70      'title' => 'Tambah Barang',
71      'list' => ['Home', 'Barang', 'Tambah']
72    ];
73    $page = (object) [
74      'title' => 'Tambah barang baru'
75    ];
76
77    $activeMenu = 'barang'; // set menu yang sedang aktif
78    return view(view: 'barang.create', data: ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
79
80  }
81
82  // Menyimpan data barang baru
83  | reference|0 overrides
84  public function store(Request $request): mixed\RedirectResponse
85  {
86    $request->validate(rules: [
87      'barang_kode' => 'required|numeric|unique:m_barang,barang_kode', // barang_kode harus unik dan numeric
88      'kategori_id' => 'required|numeric|max:20', // kategori_id harus diisi, numeric, dan maksimal 20 karakter
89      'barang_nama' => 'required|string|max:100', // barang_nama harus diisi dan maksimal 100 karakter
90      'harga_beli' => 'required|numeric', // harga_beli harus diisi dan berupa angka
91      'harga_jual' => 'required|numeric', // harga_jual harus diisi dan berupa angka
92    ]);
93
94    BarangModel::create(attributes: [
95      'barang_kode' => $request->barang_kode,
96      'kategori_id' => $request->kategori_id, // tambahkan kategori_id
97      'barang_nama' => $request->barang_nama,
98      'harga_beli' => $request->harga_beli,
99      'harga_jual' => $request->harga_jual
100    ]);
101
102    return redirect(to: '/barang')->with(key: 'success', value: 'Data barang berhasil disimpan');
103
104  }
105
106  class BarangController extends Controller
107  // Menampilkan halaman form edit barang
108  | reference|0 overrides
109  public function edit(string $id): Factory|mixed\RedirectResponse|View
110  {
111    $barang = BarangModel::find(id: $id);
112
113    if (!$barang) {
114      return redirect(to: '/barang')->with(key: 'error', value: 'Data barang tidak ditemukan');
115    }
116
117    $breadcrumb = (object) [
118      'title' => 'Edit Barang',
119      'list' => ['Home', 'Barang', 'Edit']
120    ];
121
122    $page = (object) [
123      'title' => 'Edit barang'
124    ];
125
126    $activeMenu = 'barang'; // set menu yang sedang aktif
127    return view(view: 'barang.edit', data: ['breadcrumb' => $breadcrumb, 'page' => $page, 'barang' => $barang, 'activeMenu' => $activeMenu]);
128  }
129
130  // Menyimpan perubahan data barang
131  | reference|0 overrides
132  public function update(Request $request, string $id): mixed\RedirectResponse
133  {
134    $request->validate(rules: [
135      'barang_kode' => 'required|numeric|unique:m_barang,barang_kode,' . $id . ',barang_id', // barang_kode harus unik dan numeric
136      'barang_nama' => 'required|string|max:100',
137      'harga_beli' => 'required|numeric',
138      'harga_jual' => 'required|numeric',
139      'kategori_id' => 'required|numeric|max:20' // tambahkan validasi kategori_id
140    ]);
141
142    $barang = BarangModel::find(id: $id);
143
144    if (!$barang) {
145      return redirect(to: '/barang')->with(key: 'error', value: 'Data barang tidak ditemukan');
146    }
147
148    $barang->update(attributes: [
149      'barang_kode' => $request->barang_kode,
150      'barang_nama' => $request->barang_nama,
151      'harga_beli' => $request->harga_beli,
152      'harga_jual' => $request->harga_jual,
153      'kategori_id' => $request->kategori_id // tambahkan kategori_id
154    ]);
155
156    return redirect(to: '/barang')->with(key: 'success', value: 'Data barang berhasil diubah');
157  }
```



The screenshot displays two windows of the PWL-Starter Code application. The left window shows the 'Tambah Barang' (Add Item) form with fields for ID Barang, Kode Barang, Nama Barang, Harga Beli, and Harga Jual. The right window shows the 'Daftar Barang' (Item List) page displaying a table of items with columns for ID Barang, Kode Barang, Nama Barang, Harga Beli, Harga Jual, and Kategori.

ID Barang	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori
11	Pc	DELL	8000	8200	1
12	Tablet	SAMSUNG	8500	8800	2
13	Laptop	DELL	8200	8500	3
14	Smartphone	SAMSUNG	8000	8200	1
15	Keyboard	LOGITECH	8000	8200	3
16	Mouse	LOGITECH	8000	8200	3

Gambar di atas adalah tampilan halaman penambahan Barang yang mana hanya perlu memasukan id barang, kode kategori, kode barang, nama barang, harga beli dan harga jual.kemudian klik simpan untuk menyimpan penambahan data maka akan kembali ke halaman list daftar barang yang akan menampilkan data terupdate dan keterangan hijau yang menunjukan data berhasil diinputkan

- 8 Selanjutnya membuat form untuk mengedit data, maka diperlukan file PWL/resources/views/kategori/edit.blade.php seperti berikut.

```
1 <extends name="layout" href="#" autoWidth="true">
2 <action name="listarTemplate">
3 <action name="Content">
4 <div class="card card-primary">
5   <div class="card-header">
6     <h3 class="card-title" style="text-align: center;">{{ $page.title }}EditData yang Anda cari tidak ditemukan.<a href="{{ url('barang') }}" class="btn btn-sm btn-default ml-2">Kembali</a>Kode Barang</label>
26     <div class="col-12">
27       <input type="text" class="form-control" id="barang_id" name="barang_id" value="{{ old('barang_id', $barang->barang_id) }}" required readonly="required" style="width: 100%; height: 35px; border: 1px solid #ccc; border-radius: 5px; padding: 5px; margin-bottom: 10px;">
28     </div>
29   </div>
30   <div class="form-group row">
31     <label class="col-12 control-label col-form-label" style="text-align: center;">Kode Kategori</label>
32     <div class="col-12">
33       <input type="text" class="form-control" id="kategori_id" name="kategori_id" value="{{ old('kategori_id', $barang->kategori_id) }}" required style="width: 100%; height: 35px; border: 1px solid #ccc; border-radius: 5px; padding: 5px; margin-bottom: 10px;">
34     </div>
35   </div>
36   <div class="form-group row">
37     <label class="col-12 control-label col-form-label" style="text-align: center;">Nama Barang</label>
38     <div class="col-12">
39       <input type="text" class="form-control" id="barang_nama" name="barang_nama" value="{{ old('barang_nama', $barang->barang_nama) }}" required style="width: 100%; height: 35px; border: 1px solid #ccc; border-radius: 5px; padding: 5px; margin-bottom: 10px;">
40     </div>
41   </div>
42   <div class="form-group row">
43     <label class="col-12 control-label col-form-label" style="text-align: center;">Harga Bell</label>
44     <div class="col-12">
45       <input type="text" class="form-control" id="harga_bell" name="harga_bell" value="{{ old('harga_bell', $barang->harga_bell) }}" required style="width: 100%; height: 35px; border: 1px solid #ccc; border-radius: 5px; padding: 5px; margin-bottom: 10px;">
46     </div>
47   </div>
48   <div class="form-group row">
49     <label class="col-12 control-label col-form-label" style="text-align: center;">Harga Jual</label>
50     <div class="col-12">
51       <input type="text" class="form-control" id="harga_jual" name="harga_jual" value="{{ old('harga_jual', $barang->harga_jual) }}" required style="width: 100%; height: 35px; border: 1px solid #ccc; border-radius: 5px; padding: 5px; margin-bottom: 10px;">
52     </div>
53   </div>
54   <div class="form-group row">
55     <div style="text-align: center; margin-top: 20px; margin-bottom: 10px;">
56       <button type="submit" class="btn btn-primary">Simpan</button>
57       <button type="button" class="btn btn-default ml-2" href="{{ url('barang') }}">Kembali</button>
58     </div>
59   </div>
60 </form>
61 </div>
62 </body>
63 </html>
64 </body>
65 </html>
66 </body>
67 </html>
68 </body>
69 </body>
70 </body>
71 </body>
72 </body>
73 </body>
74 </body>
75 </body>
76 </body>
77 </body>
78 </body>
79 </body>
80 </body>
81 </body>
82 </body>
83 </body>
84 </body>
85 </body>
86 </body>
87 </body>
88 </body>
89 </body>
90 </body>
91 </body>
92 </body>
93 </body>
94 </body>
```



Dan berikut adalah tampilan halaman edit kategori.

Edit data barang

kategori: Food and Beverage

Kode barang: 17

Nama barang: babycorn

Harga jual: 3500

Harga beli: 5000

Simpan Kembali

Daftar Barang

Data barang berhasil diubah

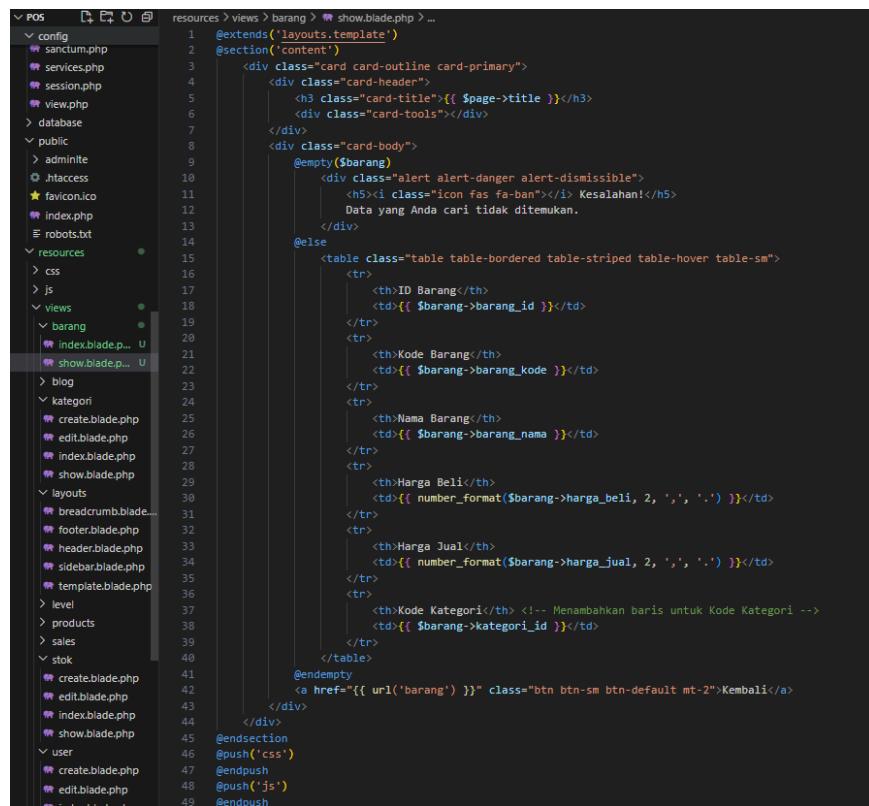
ID	Nama barang	Kode barang	Harga jual	Harga beli	Kategori barang	Aksi
1	Nasi	nas	Rp 10.000	Rp 12.000	Food and Beverage	[Detail] [Edit] [Hapus]
2	Ayam	aym	Rp 8.000	Rp 10.000	Food and Beverage	[Detail] [Edit] [Hapus]
3	Sawi	saw	Rp 5.000	Rp 7.000	Food and Beverage	[Detail] [Edit] [Hapus]
4	Promag	pmg	Rp 8.000	Rp 10.000	Beauty and Health	[Detail] [Edit] [Hapus]
5	Sunscreen	scr	Rp 12.000	Rp 15.000	Beauty and Health	[Detail] [Edit] [Hapus]
6	Dulcolax	dlx	Rp 8.000	Rp 10.000	Beauty and Health	[Detail] [Edit] [Hapus]
7	Minyak Telon	myt	Rp 15.000	Rp 19.000	Baby and Kid	[Detail] [Edit] [Hapus]
8	Popok Bayi	pop	Rp 8.000	Rp 20.000	Baby and Kid	[Detail] [Edit] [Hapus]
9	Bedak Bayi	bdk	Rp 8.000	Rp 10.000	Baby and Kid	[Detail] [Edit] [Hapus]
10	Kursi Plastik	ksr	Rp 20.000	Rp 22.000	Home and Care	[Detail] [Edit] [Hapus]
11	Pel	pel	Rp 8.000	Rp 10.000	Home and Care	[Detail] [Edit] [Hapus]
12	Sapu	sap	Rp 8.000	Rp 10.000	Home and Care	[Detail] [Edit] [Hapus]
13	Lampu	lmp	Rp 8.000	Rp 10.000	Elektronik	[Detail] [Edit] [Hapus]
14	Stop Kontak	stk	Rp 8.000	Rp 10.000	Elektronik	[Detail] [Edit] [Hapus]
15	Kabel	tbl	Rp 8.000	Rp 10.000	Elektronik	[Detail] [Edit] [Hapus]
16	Jagung	20	Rp 5.000	Rp 7.000	Food and Beverage	[Detail] [Edit] [Hapus]
17	babycorn	172	Rp 5.000	Rp 3.500	Food and Beverage	[Detail] [Edit] [Hapus]

Showing 1 to 17 of 17 entries

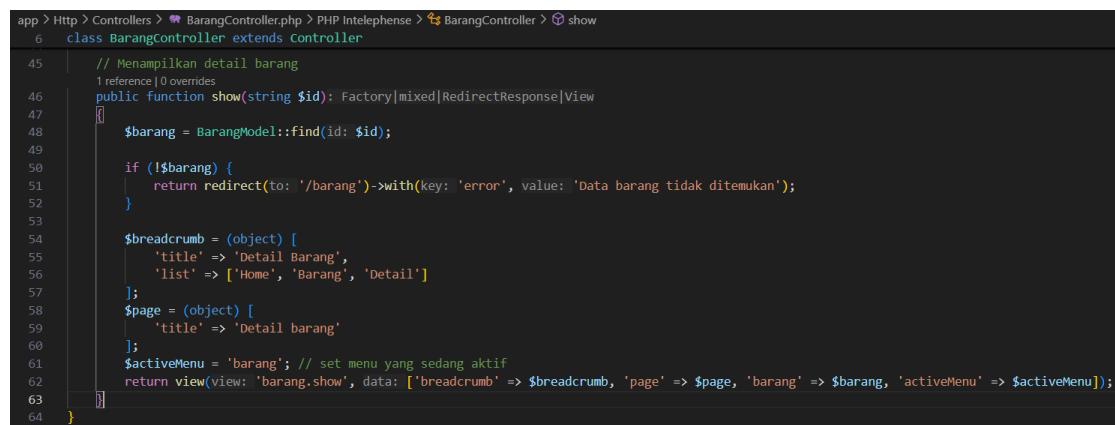


9

Untuk menampilkan detail data level user maka diperlukan file PWL/resources/views/barang/show.blade.php seperti berikut. Dan diperlukan penambahan fungsi show() pada file BarangController.php.



```
resources > views > barang > show.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">{{ $page->title }}</h3>
6             <div class="card-tools"></div>
7         </div>
8         <div class="card-body">
9             @empty($barang)
10                 <div class="alert alert-danger alert-dismissible">
11                     <h5><i class="fas fa-ban"></i> Kesalahan!</h5>
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 Barang</th>
18                         <td>{{ $barang->barang_id }}</td>
19                     </tr>
20                     <tr>
21                         <th>Kode Barang</th>
22                         <td>{{ $barang->barang_kode }}</td>
23                     </tr>
24                     <tr>
25                         <th>Nama Barang</th>
26                         <td>{{ $barang->barang_nama }}</td>
27                     </tr>
28                     <tr>
29                         <th>Harga Beli</th>
30                         <td>{{ number_format($barang->harga_beli, 2, ',', '.') }}</td>
31                     </tr>
32                     <tr>
33                         <th>Harga Jual</th>
34                         <td>{{ number_format($barang->harga_jual, 2, ',', '.') }}</td>
35                     </tr>
36                     <tr>
37                         <th>Kode Kategori</th> <!-- Menambahkan baris untuk Kode Kategori -->
38                         <td>{{ $barang->kategori_id }}</td>
39                     </tr>
40                 </table>
41             @endif
42             <a href="{{ url('barang') }}" class="btn btn-sm btn-default mt-2">Kembali</a>
43         </div>
44     </div>
45     @endsection
46     @push('css')
47     @endpush
48     @push('js')
49     @endpush
50 
```



```
app > Http > Controllers > BarangController.php > PHP Intelephense > BarangController > show
6 class BarangController extends Controller
45     // Menampilkan detail barang
46     public function show(string $id): Factory|mixed|RedirectResponse|View
47     {
48         $barang = BarangModel::find(id: $id);
49
50         if (!$barang) {
51             return redirect(to: '/barang')->with(key: 'error', value: 'Data barang tidak ditemukan');
52         }
53
54         $breadcrumb = (object) [
55             'title' => 'Detail Barang',
56             'list' => ['Home', 'Barang', 'Detail']
57         ];
58         $page = (object) [
59             'title' => 'Detail barang'
60         ];
61         $activeMenu = 'barang'; // set menu yang sedang aktif
62         return view(view: 'barang.show', data: ['breadcrumb' => $breadcrumb, 'page' => $page, 'barang' => $barang, 'activeMenu' => $activeMenu]);
63     }
64 }
```

Berikut adalah tampilan halaman detail barang yang mengambil data dari database m_barang.



10	Untuk melakukan penghapusan data maka diperlukan penambahan fungsi <code>destroy(string \$id)</code> pada file <code>BarangController.php</code> .
	<pre>app > Http > Controllers > BarangController.php > ... 10 class BarangController extends Controller 11 { 12 ... 13 public function destroy(string \$barang_id): mixed RedirectResponse{ 14 \$check = barangmodel::find(\$barang_id); 15 if(!\$check){ 16 return redirect(to: '/barang')->with(key: 'error',value: 'Data user tidak ditemukan'); 17 } 18 19 try{ 20 barangmodel::destroy(\$barang_id); 21 return redirect(to: '/barang')->with(key: 'success', value: 'Data user berhasil dihapus'); 22 } catch (\Illuminate\Database\QueryException \$e){ 23 return redirect(to: '/barang')->with(key: 'error',value: 'Data user gagal dihapus karena masih terdapat tabel lain yang terkait dengan data ini'); 24 } 25 } 26 }</pre>