

**LAPORAN PRAKTIKUM**  
**PEMROGRAMAN WEB LANJUT**  
**BLADE VIEW, WEB TEMPLATING (ADMINLTE), DATATABLES**



**Disusun Oleh :**  
Qusnul Diah Mawanti      2341760035

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

**2025**



Jurusan Teknologi Informasi – Politeknik Negeri  
Malang

Jobsheet 5.1 : Blade View, Web Templating(AdminLTE),  
Datatables

Mata Kuliah : Pemrograman Web Lanjut

Kelas : 2A – SIB

Nama : Qusnul Diah Mawanti

No absen : 25

NIM : 2341760035

**Maret 2025**

---

## **PERSIAPAN**

View merupakan tempat bagi kita untuk meletakkan kode-kode HTML. Sehingga dalam laravel tidak menggunakan lagi file berformat .html. Kita masih menggunakan tagging HTML, tapi kita perlu menghandle tampilan dengan lebih canggih. Menampilkan data yang diberikan oleh controller. Untuk itu kita akan menggunakan templating engine, yaitu Blade.

Blade merupakan templating engine bawaan Laravel. Berguna untuk mempermudah dalam menulis kode tampilan. Dan juga memberikan fitur tambahan untuk memanipulasi data di view yang dilempar dari controller. Blade juga memungkinkan penggunaan plain PHP pada kode View.

Sesuai dengan [Studi Kasus PWL.pdf](#).

Jadi project Laravel 10 kita masih sama dengan menggunakan repositori **PWL\_POS**.

*Project PWL\_POS* akan kita gunakan sampai pertemuan 12 nanti, sebagai project yang akan kita pelajari

```
C:\laragon\www>composer create-project "laravel/laravel:^10.0" PWL_POS
Creating a "laravel/laravel:^10.0" project at "./PWL_POS"
Installing laravel/laravel (v10.3.3)
- Installing laravel/laravel (v10.3.3): Extracting archive
Created project in C:\laragon\www\PWL_POS
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 111 installs, 0 updates, 0 removals
- Locking brick/math (0.12.3)
```

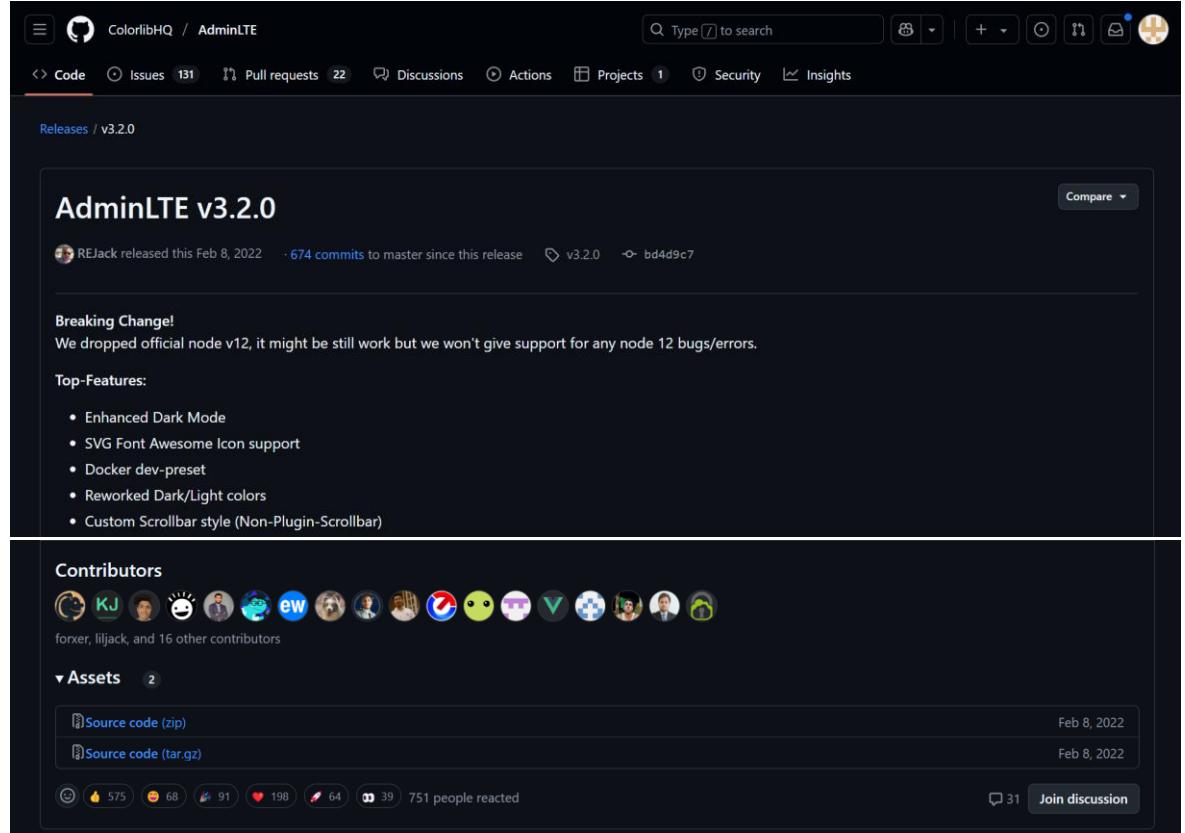
# DASAR TEORI

## 1. Praktikum 1: Layouting AdminLTE

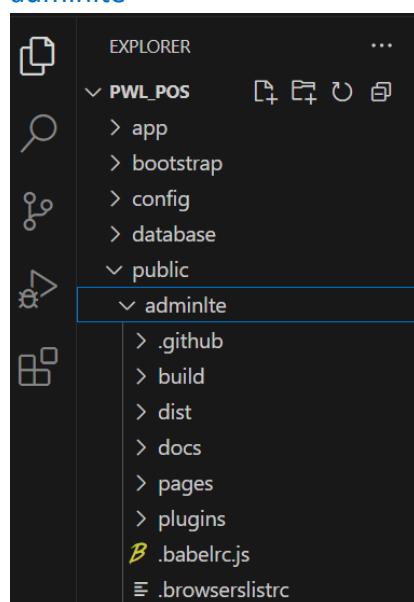
Menggunakan repository dalam tautan berikut:

<https://github.com/ColorlibHQ/AdminLTE/releases/tag/v3.2.0>.

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

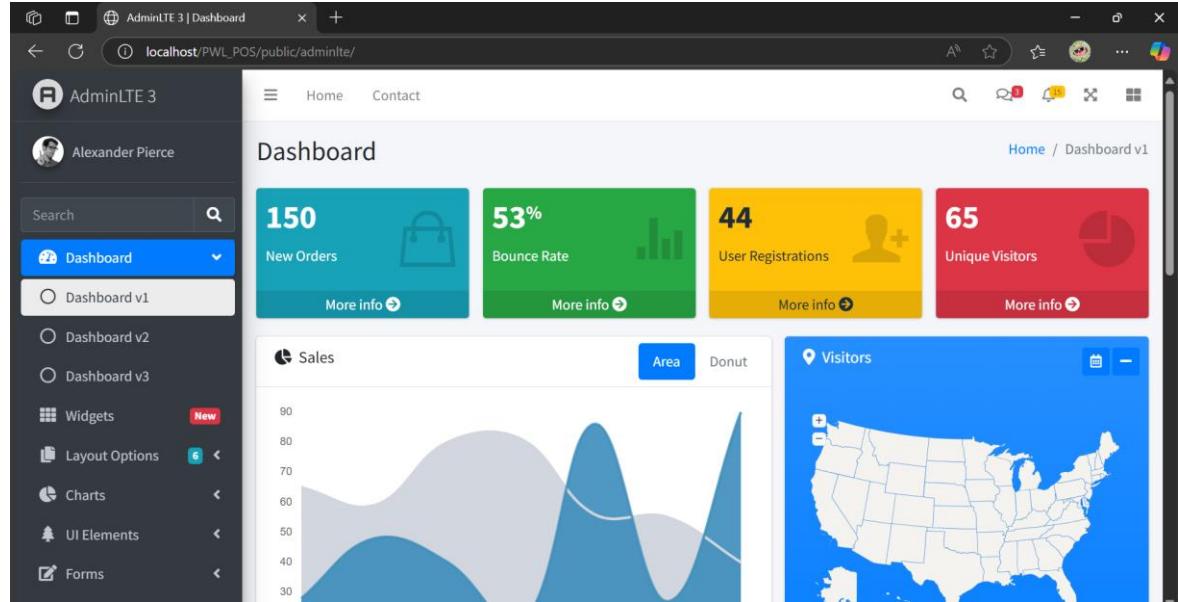


- Setelah kita berhasil download, kita ekstrak file yang sudah di download ke folder project `PWL_POS/public`, kemudian kita **rename** folder cukup menjadi `adminlte`



c. Selanjutnya kita buka di browser dengan alamat

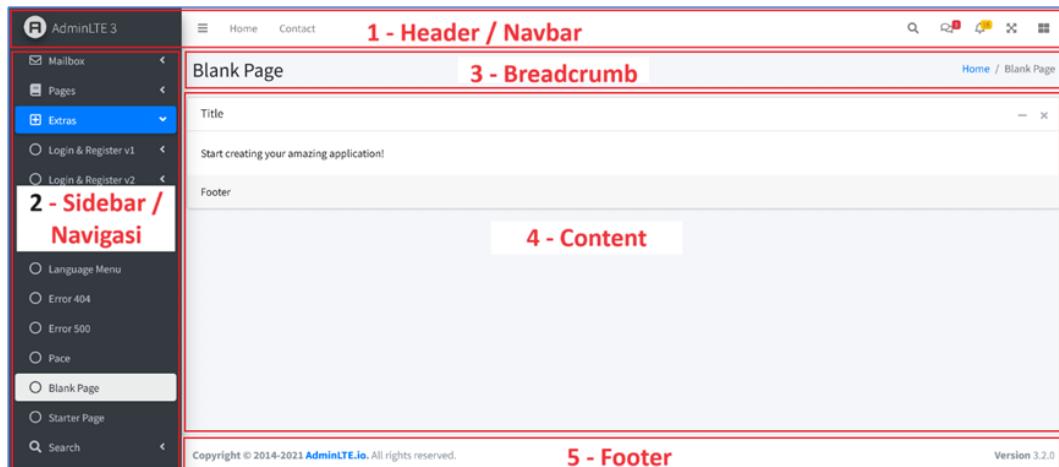
[http://localhost/PWL\\_POS/public/adminlte](http://localhost/PWL_POS/public/adminlte) maka akan muncul tampilan seperti berikut



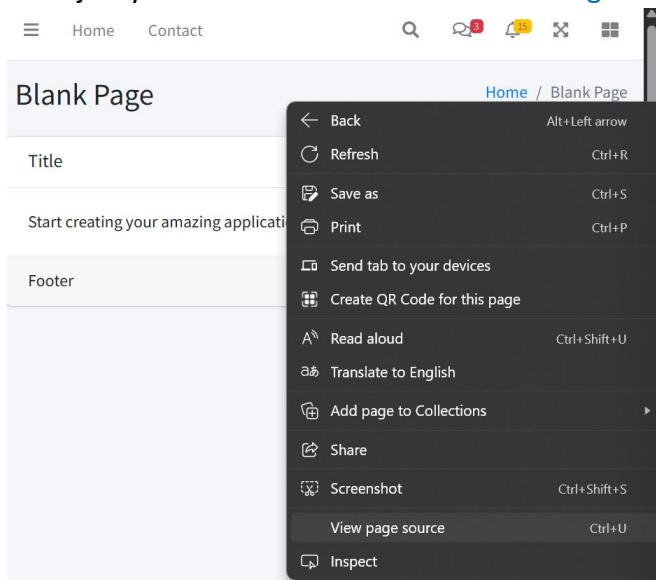
d. Kita klik menu Extras > Blank Page, page inilah yang akan menjadi dasar web template

The screenshot shows a web browser window with the URL [localhost/PWL\\_POS/public/adminlte/pages/examples/blank.html](http://localhost/PWL_POS/public/adminlte/pages/examples/blank.html). The page title is "AdminLTE 3 | Blank Page". On the left, there is a sidebar menu with the following items: Extras (selected), Login & Register v1, Login & Register v2, Lockscreen, Legacy User Menu, Language Menu, Error 404, Error 500, Pace, and Blank Page. The main content area is titled "Blank Page" and contains a "Title" input field, a "Start creating your amazing application!" message, and a "Footer" input field.

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



- f. Selanjutnya kita klik kanan halaman [Blank Page](#) dan klik [view page source](#)



- g. 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)

```

EXPLORER ... UserController.php template.blade.php X
PWL_POS
  app
  bootstrap
  config
  database
  public
  resources
    css
    js
    views
      layouts
        template.blade.php
        kategori.blade.php
        level.blade.php
        user_tambah.blade.php
resources > views > layouts > template.blade.php > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="utf-8">
5    <meta name="viewport" content="width=device-width,
6      <title>AdminLTE 3 | Blank Page</title>
7
8
9    <!-- Google Font: Source Sans Pro -->
10   <link rel="stylesheet" href="https://fonts.googleapis.com/
11     <!-- Font Awesome -->
12   <link rel="stylesheet" href="../../plugins/fontawesome-free/
13     <!-- Theme style -->
14   <link rel="stylesheet" href="../../dist/css/adminlte/
15   </head>
16   <body class="hold-transition sidebar-mini">
17     <!-- Site wrapper -->

```

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

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

```
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">
```

Menjadi

```
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>{{ config('app.name', 'PWL Laravel Starter Code') }}</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="{{ asset('adminlte/plugins/fontawesome-free/css/all.min.css') }}">
13   <!-- Theme style -->
14   <link rel="stylesheet" href="{{ asset('adminlte/dist/css/adminlte.min.css') }}">
```

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

```
15  </head>
16  <body class="hold-transition sidebar-mini">
17  <!-- Site wrapper -->
18  <div class="wrapper">
19    <!-- Navbar -->
20    @include('layout.header')
21    <!-- /.navbar -->
22
23    <!-- Main Sidebar Container -->
24    <aside class="main-sidebar sidebar-dark-primary elevation-4">
25      <!-- Brand Logo -->
26      <a href="../../index3.html" class="brand-link">
27        AdminLTE 3</span>
29      </a>
30
31      <!-- Sidebar -->
32      <div class="sidebar">
33        <!-- Sidebar user (optional) -->
```

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

k. Kita modifikasi baris **26** dan **27** pada `template.blade.php`

```
23  <!-- Main Sidebar Container -->
24  <aside class="main-sidebar sidebar-dark-primary elevation-4">
25    <!-- Brand Logo -->
26    <a href="../../index3.html" class="brand-link">
27      AdminLTE 3</span>
29    </a>
```

## Menjadi

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

- I. Selanjutnya kita blok baris **31-693** (baris untuk **element 2-sidebar**), lalu kita **cut**, dan **paste**-kan di file **PWL\_POS/resource/view/layouts/sidebar.blade.php** (buat dulu file sidebar.blade.php jika belum ada). Sehingga tampilan dari file **template.blade.php** menjadi seperti berikut

```
23  <!-- Main Sidebar Container -->
24  <aside class="main-sidebar sidebar-dark-primary elevation-4">
25  <!-- Brand Logo -->
26  <a href="{{ url('/') }}" class="brand-link">
27  |   PWL - Starter Code</span>
29 </a>
30
31  <!-- sidebar -->
32  @include('layouts.sidebar')
33  <!-- ./sidebar -->
34 </aside>
35
36  <!-- Content Wrapper. Contains page content -->
37  <div class="content-wrapper">
38  <!-- Content Header (Page header) -->
39  <section class="content-header">
```

- m. 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

```
83      </section>
84      <!-- /.content -->
85  </div>
86  <!-- /.content-wrapper -->
87
88  @include('layout.footer')
89 </div>
90 <!-- ./wrapper -->
```

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

```
92  <!-- jQuery -->
93  <script src="../../plugins/jquery/jquery.min.js"></script>
94  <!-- Bootstrap 4 -->
95  <script src="../../plugins/bootstrap/js/bootstrap.bundle.min.js"></script>
96  <!-- AdminLTE App -->
97  <script src="../../dist/js/adminlte.min.js"></script>
98  <!-- AdminLTE for demo purposes -->
99  <script src="../../dist/js/demo.js"></script>
100 </body>
101 </html>
```

## Menjadi

```
92  <!-- jQuery -->
93  <script src="{{ asset('adminlte/plugins/jquery/jquery.min.js') }}"></script>
94  <!-- Bootstrap 4 -->
95  <script src="{{ asset('adminlte/plugins/bootstrap/js/bootstrap.bundle.min.js') }}"></script>
96  <!-- AdminLTE App -->
97  <script src="{{ asset('adminlte/dist/js/adminlte.min.js') }}"></script>
98 </body>
99 </html>
100
```

- o. Sekarang masuk pada bagian konten. Konten kita bagi menjadi 2, yaitu elemen untuk breadcrumb dan elemen untuk **content**.
- p. 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

```

36  <!-- Content Wrapper. Contains page content -->
37  <div class="content-wrapper">
38  |   <!-- Content Header (Page header) -->
39  |   @include('layouts.breadcrumb')
40
41  |   <!-- Main content -->
42  |   <section class="content">
43  |       <!-- Default box -->
44  |       <div class="card">
45  |           <div class="card-header">
46  |               <h3 class="card-title">Title</h3>

```

- q. 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.
- r. Untuk **content**, kita akan menghapus baris **42-66** pada file [template.blade.php](#). dan kita ganti dengan kode seperti ini `@yield('content')`
- s. Hasil akhir pada file utama [layouts/template.blade.php](#) adalah seperti berikut

```

16  <body class="hold-transition sidebar-mini">
17  <!-- Site wrapper -->
18  <div class="wrapper">
19  |   <!-- Navbar -->
20  |   @include('layouts.header')
21  |   <!-- /.navbar -->
22
23  |   <!-- Main Sidebar Container -->
24  |   <aside class="main-sidebar sidebar-dark-primary elevation-4">
25  |       <!-- Brand Logo -->
26  |       <a href="{{ url('/') }}" class="brand-link">
27  |           
28  |           <span class="brand-text font-weight-light">PWL - Starter Code</span>
29  |       </a>
30
31  |       <!-- sidebar -->
32  |       @include('layouts.sidebar')
33  |       <!-- /.sidebar -->
34  |   </aside>
35
36  |   <!-- Content Wrapper. Contains page content -->
37  <div class="content-wrapper">
38  |       <!-- Content Header (Page header) -->
39  |       @include('layouts.breadcrumb')
40
41  |       <!-- Main content -->
42  |       <section class="content">
43  |           @yield('content')
44  |       </section>
45  |       <!-- /.content -->
46  |   </div>
47  |   <!-- /.content-wrapper -->
48
49  |   @include('layouts.footer')
50 </div>
51 <!-- /.wrapper -->

```

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

## 2. Praktikum 2: Penerapan Layouting

Sekarang kita akan mencoba melakukan penerapan terhadap layouting yang sudah kita lakukan.

- Kita buat file controller dengan nama `WelcomeController.php`

```
C:\laragon\www\PWL_POS>php artisan make:controller WelcomeController
INFO Controller [C:\laragon\www\PWL_POS\app\Http\Controllers\WelcomeController.php] created successfully.

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

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

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

- Kita modifikasi file `PWL_POS/resources/views/layouts/breadcrumb.blade.php`

```
resources > views > layouts > breadcrumb.blade.php > ...
1  <section class="content-header">
2      <div class="container-fluid">
3          <div class="row mb-2">
4              <div class="col-sm-6"><h1>{{ $breadcrumb->title }}</h1></div>
5              <div class="col-sm-6">
6                  <ol class="breadcrumb float-sm-right">
7                      @foreach($breadcrumb->list as $key => $value)
8                          @if($key == count($breadcrumb->list) - 1)
9                              <li class="breadcrumb-item active">{{ $value }}</li>
10                         @else
11                             <li class="breadcrumb-item">{{ $value }}</li>
12                         @endif
13                     @endforeach
14                 </ol>
15             </div>
16         </div>
17     </div>
18 </section>
```

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

```
1  <div class="sidebar">
2    <!-- SidebarSearch Form -->
3    <div class="form-inline mt-2">
4      <div class="input-group" data-widget="sidebar-search">
5        <input class="form-control form-control-sidebar" type="search"
6        placeholder="Search" aria-label="Search">
7        <div class="input-group-append">
8          <button class="btn btn-sidebar">
9            <i class="fas fa-search fa-fw"></i>
10         </button>
11       </div>
12     </div>
13   </div>
14   <!-- Sidebar Menu -->
15   <nav class="mt-2">
16     <ul class="nav nav-pills nav-sidebar flex-column" data-widget="treeview"
17     role="menu" data-accordion="false">
18       <li class="nav-item">
19         <a href="{{ url('/') }}" class="nav-link {{ ($activeMenu == 'dashboard')?
20 'active' : '' }} ">
21           <i class="nav-icon fas fa-tachometer-alt"></i>
22           <p>Dashboard</p>
23         </a>
24       </li>
25       <li class="nav-header">Data Pengguna</li>
26       <li class="nav-item">
27         <a href="{{ url('/level') }}" class="nav-link {{ ($activeMenu == 'level')?
28 'active' : '' }} ">
29           <i class="nav-icon fas fa-layer-group"></i>
30           <p>Level User</p>
31         </a>
32       </li>
33       <li class="nav-item">
34         <a href="{{ url('/user') }}" class="nav-link {{ ($activeMenu == 'user')?
35 'active' : '' }} ">
36           <i class="nav-icon far fa-user"></i>
37           <p>Data User</p>
38         </a>
39       </li>
40       <li class="nav-header">Data Barang</li>
41       <li class="nav-item">
42         <a href="{{ url('/kategori') }}" class="nav-link {{ ($activeMenu ==
43 'kategori')? 'active' : '' }} ">
44           <i class="nav-icon far fa-bookmark"></i>
45           <p>Kategori Barang</p>
46         </a>
47       </li>
48       <li class="nav-item">
49         <a href="{{ url('/barang') }}" class="nav-link {{ ($activeMenu ==
```

```

50   'barang')? 'active' : '' }} ">
51   |   <i class="nav-icon far fa-list-alt"></i>
52   |   <p>Data Barang</p>
53   |   </a>
54   |   </li>
55   |   <li class="nav-header">Data Transaksi</li>
56   |   <li class="nav-item">
57   |   |   <a href="{{ url('/stok') }}" class="nav-link {{ ($activeMenu == 'stok')?
58   |   'active' : '' }} ">
59   |   |   |   <i class="nav-icon fas fa-cubes"></i>
60   |   |   |   <p>Stok Barang</p>
61   |   |   </a>
62   |   |   </li>
63   |   |   <li class="nav-item">
64   |   |   |   <a href="{{ url('/barang') }}" class="nav-link {{ ($activeMenu ==
65   |   |   'penjualan')? 'active' : '' }} ">
66   |   |   |   |   <i class="nav-icon fas fa-cash-register"></i>
67   |   |   |   |   <p>Transaksi Penjualan</p>
68   |   |   |   </a>
69   |   |   |   </li>
70   |   |   </ul>
71   |   </nav>
72   |   </div>

```

- e. Kita tambahkan kode berikut router `web.php`

```

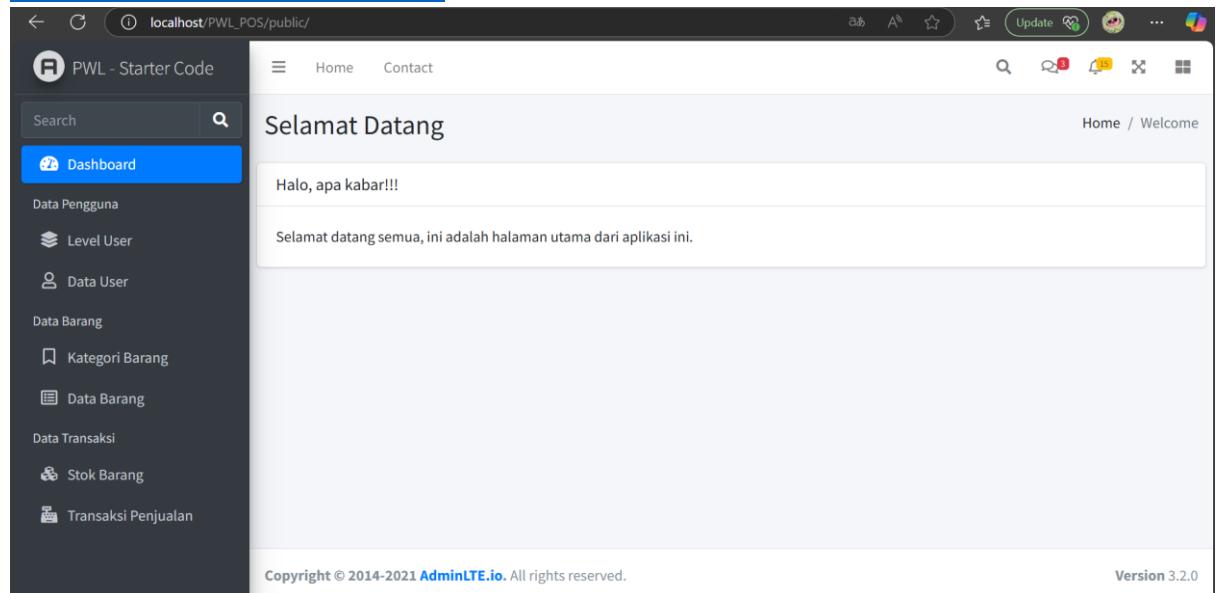
use App\Http\Controllers\WelcomeController;

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

```

- f. Sekarang kita coba jalankan di browser dengan mengetikkan url

[http://localhost/PWL\\_POS/public](http://localhost/PWL_POS/public)



- g. Jangan lupa commit dan push ke github PWL\_POS kalian

### 3. Praktikum 3: Implementasi jQuery Datatable di AdminLTE

- Kita modifikasi proses CRUD pada tabel `m_user` pada praktikum ini
- Kita gunakan library Yajra-datable dengan mengetikkan perintah pada CMD  
`composer require yajra/laravel-datatables:^10.0` atau  
`composer require yajra/laravel-datatables-oracle`

```
C:\laragon\www\PWL_POS>composer require yajra/laravel-datatables:^10.0
The "10.0" constraint for "yajra/laravel-datatables" appears too strict
and will likely not match what you want. See https://getcomposer.org/c
onstraints
./composer.json has been updated
Running composer update yajra/laravel-datatables
Loading composer repositories with package information
Updating dependencies
Lock file operations: 7 installs, 0 updates, 0 removals
  - Locking league/fractal (0.20.2)
  - Locking yajra/laravel-datatables (v10.0.0)
```

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

```
5  use App\Http\Controllers\WelcomeController;
6  use Illuminate\Support\Facades\Route;
7  use App\Http\Controllers\UserController;
8
9  Route::get('/', [WelcomeController::class, 'index']);
10
11 Route::group(['prefix' => 'user'], function () {
12     Route::get('/', [UserController::class, 'index']);           // menampilkan halaman awal user
13     Route::post('/list', [UserController::class, 'list']);        // menampilkan data user dalam bentuk JSON untuk datatables
14     Route::get('/create', [UserController::class, 'create']);    // menampilkan halaman form tambah user
15     Route::post('/', [UserController::class, 'store']);          // menyimpan data user baru
16     Route::get('/{id}', [UserController::class, 'show']);         // menampilkan detail user
17     Route::get('/{id}/edit', [UserController::class, 'edit']);    // menampilkan halaman form edit user
18     Route::put('/{id}', [UserController::class, 'update']);      // menyimpan perubahan data user
19     Route::delete('/{id}', [UserController::class, 'destroy']);   // menghapus data user
20 });
});
```

- 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  namespace App\Http\Controllers;
3
4  use App\Models\LevelModel;
5  use App\Models\UserModel;
6  use Illuminate\Http\Request;
7  use Yajra\DataTables\Facades\DataTables;
8
9  class UserController extends Controller
10 {
11     // Menampilkan halaman awal user
12     public function index()
13     {
14         $breadcrumb = (object) [
15             'title' => 'Daftar User',
16             'list'  => ['Home', 'User']
17         ];
18
19         $page = (object) [
20             'title' => 'Daftar user yang terdaftar dalam sistem'
21         ];
22
23         $activeMenu = 'user'; // set menu yang sedang aktif
24
25         return view('user.index', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
26     }
27 }
```

- 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"
13 id="table_user">
14              <thead>
15                  <tr><th>ID</th><th>Username</th><th>>Nama</th><th>Level
16 Pengguna</th><th>Aksi</th></tr>
17              </thead>
18          </table>
19      </div>
20  </div>
21  @endsection
22
23  @push('css')
24  @endpush
25
26  @push('js')
27      <script>
28          $(document).ready(function() {
29              var dataUser = $('#table_user').DataTable({
30                  // serverSide: true, jika ingin menggunakan server side processing
31                  serverSide: true,
32                  ajax: {
33                      "url": "{{ url('user/list') }}",
34                      "dataType": "json",
35                      "type": "POST"
36                  },
37                  columns: [
38                      {
39                          // nomor urut dari laravel datatable addIndexColumn()
40                          data: "DT_RowIndex",
41                          className: "text-center",
42                          orderable: false,
43                          searchable: false
44                      },{
45                          data: "username",
46                          className: "",
47                          // orderable: true, jika ingin kolom ini bisa diurutkan
48                          orderable: true,
49                          // searchable: true, jika ingin kolom ini bisa dicari
50                          searchable: true
51                      },{
52                          data: "nama",
53                          className: "",
54                          orderable: true,
55                          searchable: true
56                      },{
57                          // mengambil data level hasil dari ORM berelasi
58                          data: "level.level_nama",
59                          className: "",
60                          orderable: false,
61                          searchable: false
62                      },{
63                          data: "aksi",
64                          className: "",
65                          orderable: false,
66                          searchable: false
67                      }
68                  ]
69              });
70          });
71      </script>
72  @endpush

```

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

```

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:300,400,400i,700&display=fallback">
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     <!-- jQuery -->
25     <script src="{{ asset('adminlte/plugins/jquery/jquery.min.js') }}></script>
26     <!-- Bootstrap 4 -->
27     <script src="{{ asset('adminlte/plugins/bootstrap/js/bootstrap.bundle.min.js') }}></script>
28     <!-- DataTables & Plugins -->
29     <script src="{{ asset('adminlte/plugins/datatables/jquery.dataTables.min.js') }}></script>
30     <script src="{{ asset('adminlte/plugins/datatables-bs4/js/dataTables.bootstrap4.min.js') }}></script>
31     <script src="{{ asset('adminlte/plugins/datatables-responsive/js/dataTables.responsive.min.js') }}></script>
32     <script src="{{ asset('adminlte/plugins/datatables-responsive/js/responsive.bootstrap4.min.js') }}></script>
33     <script src="{{ asset('adminlte/plugins/datatables-buttons/js/dataTables.buttons.min.js') }}></script>
34     <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.bootstrap4.min.js') }}></script>
35     <script src="{{ asset('adminlte/plugins/jstable/jstable.min.js') }}></script>
36     <script src="{{ asset('adminlte/plugins/pdfmake/pdfmake.min.js') }}></script>
37     <script src="{{ asset('adminlte/plugins/pdfmake/vfs_fonts.js') }}></script>
38     <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.html5.min.js') }}></script>
39     <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.print.min.js') }}></script>
40     <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.colVis.min.js') }}></script>
41     <!-- AdminLTE App -->
42     <script src="{{ asset('adminlte/dist/js/adminlte.min.js') }}></script>
43     <script>
44         // Untuk mengirimkan token Laravel CSRF pada setiap request ajax
45         $.ajaxSetup({headers: { 'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content') }});
46     </script>
47     @stack('js') <!-- Digunakan untuk memanggil custom js dari perintah push('js') pada masing-masing view -->
48 </body>
49 </html>

```

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

```

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

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

```

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

ID	Username	Nama	Level Pengguna	Aksi
1	admin	Administrator	Administrator	[Detail, Edit, Hapus]
2	manager	Manager	Manager	[Detail, Edit, Hapus]
3	staf	Staf/Kasir	Staf/Kasir	[Detail, Edit, Hapus]
4	customer-1	Pelanggan Pertama	Pelanggan	[Detail, Edit, Hapus]
5	manager_dua	Manager 2	Manager	[Detail, Edit, Hapus]
6	manager_tiga	Manager 3	Manager	[Detail, Edit, Hapus]
7	manager9	Manager 9	Manager	[Detail, Edit, Hapus]
8	manager22	Manager Dua Dua	Manager	[Detail, Edit, Hapus]
9	manager33	Manager Tiga Tiga	Manager	[Detail, Edit, Hapus]
10	manager56	Manager55	Manager	[Detail, Edit, Hapus]

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

```
// Menampilkan halaman form tambah user
public function create()
{
    $breadcrumb = (object) [
        'title' => 'Tambah User',
        'list'  => ['Home', 'User', 'Tambah']
    ];

    $page = (object) [
        'title' => 'Tambah user baru'
    ];

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

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

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

```

resources > views > user > 📄 create.blade.php > ...
1  @extends('layouts.template')
2
3  @section('content')
4      <div class="card card-outline card-primary">
5          <div class="card-header">
6              <h3 class="card-title">{{ $page->title }}</h3>
7              <div class="card-tools"></div>
8          </div>
9          <div class="card-body">
10             <form method="POST" action="{{ url('user') }}" class="form-horizontal">
11                 @csrf
12                 <div class="form-group row">
13                     <label class="col-1 control-label col-form-label">Level</label>
14                     <div class="col-11">
15                         <select class="form-control" id="level_id" name="level_id" required>
16                             <option value="">- Pilih Level -</option>
17                             @foreach($level as $item)
18                                 <option value="{{ $item->level_id }}>{{ $item->level_nama }}</option>
19                             @endforeach
20                         </select>
21                         @error('level_id')
22                             <small class="form-text text-danger">{{ $message }}</small>
23                         @enderror
24                     </div>
25                 </div>
26                 <div class="form-group row">
27                     <label class="col-1 control-label col-form-label">Username</label>
28                     <div class="col-11">
29                         <input type="text" class="form-control" id="username" name="username" value="{{
30                             old('username') }}" required>
31                         @error('username')
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</label>
38                     <div class="col-11">
39                         <input type="text" class="form-control" id="nama" name="nama" value="{{
40                             old('nama') }}" required>
41                         @error('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">Password</label>
48                     <div class="col-11">
49                         <input type="password" class="form-control" id="password" name="password"
50 required>
51                         @error('password')
52                             <small class="form-text text-danger">{{ $message }}</small>
53                         @enderror
54                     </div>
55                 </div>
56                 <div class="form-group row">
57                     <label class="col-1 control-label col-form-label"></label>
58                     <div class="col-11">
59                         <button type="submit" class="btn btn-primary btn-sm">Simpan</button>
60                         <a class="btn btn-sm btn-default ml-1" href="{{ url('user') }}>Kembali</a>
61                     </div>
62                 </div>
63             </form>
64         </div>
65     </div>
66     @endsection
67     @push('css')
68     @endpush
69     @push('js')
70     @endpush

```

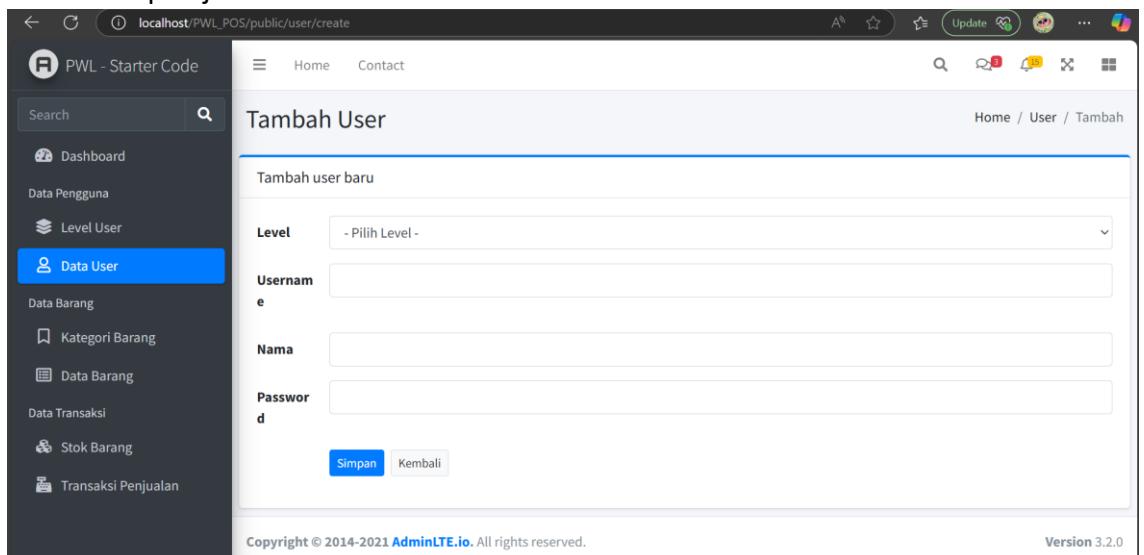
- k. Kemudian untuk bisa menng-*handle* data yang akan disimpan ke database, kita buat fungsi **store()** di **UserController.php**

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

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

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

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



- m. Selanjutnya, kita masuk pada bagian menampilkan detail data user (klik tombol **Detail**) pada halaman user. Route yang bertugas untuk menangkap request detail adalah

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

- n. Jadi kita buat/modifikasi fungsi **show()** pada **UserController.php** seperti berikut

```
// Menampilkan detail user
public function show(string $id)
{
    $user = UserModel::with('level')->find($id);

    $breadcrumb = (object) [
        'title' => 'Detail User',
        'list'  => ['Home', 'User', 'Detail']
    ];

    $page = (object) [
        'title' => 'Detail user'
    ];

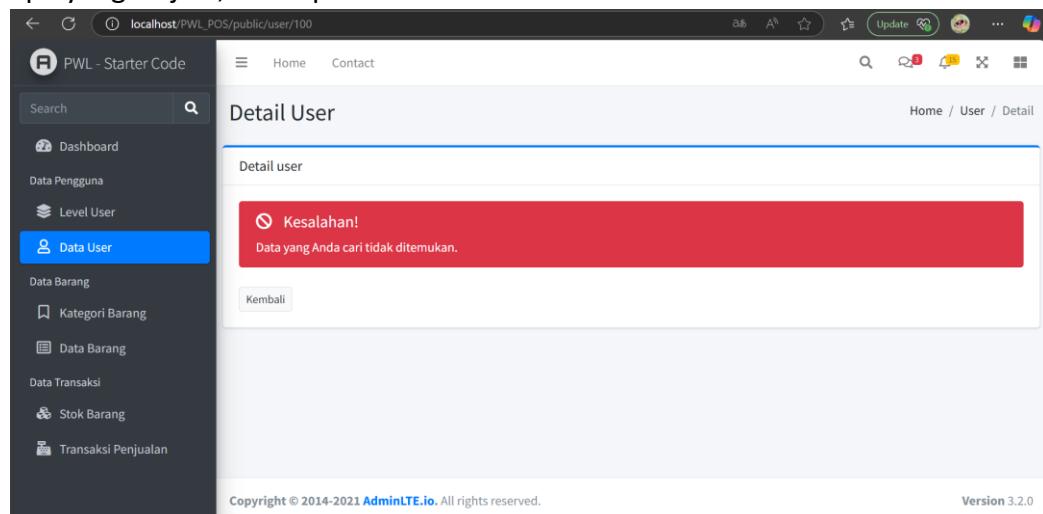
    $activeMenu = 'user'; // set menu yang sedang aktif

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

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

```
1  @extends('layouts.template')
2
3  @section('content')
4      <div class="card card-outline card-primary">
5          <div class="card-header">
6              <h3 class="card-title">{{ $page->title }}</h3>
7              <div class="card-tools"></div>
8          </div>
9          <div class="card-body">
10             @empty($user)
11                 <div class="alert alert-danger alert-dismissible">
12                     <h5><i class="icon fas fa-ban"></i> Kesalahan!</h5>
13                     Data yang Anda cari tidak ditemukan.
14                 </div>
15             @else
16                 <table class="table table-bordered table-striped table-hover table-sm">
17                     <tr>
18                         <th>ID</th>
19                         <td>{{ $user->user_id }}</td>
20                     </tr>
21                     <tr>
22                         <th>Level</th>
23                         <td>{{ $user->level->level_nama }}</td>
24                     </tr>
25                     <tr>
26                         <th>Username</th>
27                         <td>{{ $user->username }}</td>
28                     </tr>
29                     <tr>
30                         <th>Nama</th>
31                         <td>{{ $user->nama }}</td>
32                     </tr>
33                     <tr>
34                         <th>Password</th>
35                         <td>*****</td>
36                     </tr>
37                 </table>
38             @endempty
39             <a href="{{ url('user') }}" class="btn btn-sm btn-default mt-2">Kembali</a>
40         </div>
41     </div>
42     @endsection
43
44     @push('css')
45     @endpush
46
47     @push('js')
48     @endpush
```

- 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!!!



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

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

- r. Jadi kita buat fungsi **edit()**

```
// Menampilkan halaman form edit user
public function edit(string $id)
{
    $user = UserModel::find($id);
    $level = LevelModel::all();

    $breadcrumb = (object) [
        'title' => 'Edit User',
        'list'  => ['Home', 'User', 'Edit']
    ];

    $page = (object) [
        'title' => 'Edit user'
    ];

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

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

dan **update()** pada [UserController.php](#)

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

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

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

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

```

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

```

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

Username berhasil diubah

10	manager55	Manager55	Manager	<a href="#">Detail</a>	<a href="#">Edit</a>	<a href="#">Hapus</a>
----	-----------	-----------	---------	------------------------	----------------------	-----------------------

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

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

```
Route::delete('/{id}', [UserController::class, 'destroy']); // menghapus data user
```

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

```
// Menghapus data user
public function destroy(string $id)
{
    $check = UserModel::find($id);
    if (!$check) { // untuk mengecek apakah data user dengan id yang dimaksud ada atau tidak
        return redirect('/user')->with('error', 'Data user tidak ditemukan');
    }

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

        return redirect('/user')->with('success', 'Data user berhasil dihapus');
    } catch (\Illuminate\Database\QueryException $e) {

        // Jika terjadi error ketika menghapus data, redirect kembali ke halaman dengan membawa pesan error
        return redirect('/user')->with('error', 'Data user gagal dihapus karena masih terdapat tabel lain yang terkait dengan data ini');
    }
}
```

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

```
11     <div class="card-body">
12         <table class="table table-bordered table-striped table-hover table-sm"
13             id="table_user">
14             <thead>
15                 <tr><th>ID</th><th>Username</th><th>Nama</th><th>Level
16 Pengguna</th><th>Aksi</th></tr>
17             </thead>
18             </table>
19     </div>
20 </div>
```

Menjadi

```

11 |         <div class="card-body">
12 |             @if (session('success'))
13 |                 <div class="alert alert-success">{{ session('success') }}</div>
14 |             @endif
15 |             @if (session('error'))
16 |                 <div class="alert alert-danger">{{ session('error') }}</div>
17 |             @endif
18 |             <table class="table table-bordered table-striped table-hover table-sm" id="table_user">
19 |                 <thead>
20 |                     <tr><th>ID</th><th>Username</th><th>Nama</th><th>Level Pengguna</th><th>Aksi</th></tr>
21 |                 </thead>
22 |                 </table>
23 |             </div>

```

- x. Kemudian jalankan browser untuk menghapus salah satu data user. Amati dan lapor kan!

ID	Username	Nama	Level Pengguna	Aksi
1	admin	Administrator	Administrator	[Detail] [Edit] [Hapus]
2	manager	Manager	Manager	[Detail] [Edit] [Hapus]
3	staf	Staf/Kasir	Staf/Kasir	[Detail] [Edit] [Hapus]
4	customer-1	Pelanggan Pertama	Pelanggan	[Detail] [Edit] [Hapus]
5	manager_dua	Manager 2	Manager	[Detail] [Edit] [Hapus]
6	manager_tiga	Manager 3	Manager	[Detail] [Edit] [Hapus]
7	manager9	Manager 9	Manager	[Detail] [Edit] [Hapus]
8	manager22	Manager Dua Dua	Manager	[Detail] [Edit] [Hapus]
9	manager33	Manager Tiga Tiga	Manager	[Detail] [Edit] [Hapus]
10	manager12	Manager11	Manager	[Detail] [Edit] [Hapus]

Data manager12 berhasil dihapus

ID	Username	Nama	Level Pengguna	Aksi
1	admin	Administrator	Administrator	[Detail] [Edit] [Hapus]
2	manager	Manager	Manager	[Detail] [Edit] [Hapus]
3	staf	Staf/Kasir	Staf/Kasir	[Detail] [Edit] [Hapus]
4	customer-1	Pelanggan Pertama	Pelanggan	[Detail] [Edit] [Hapus]
5	manager_dua	Manager 2	Manager	[Detail] [Edit] [Hapus]
6	manager_tiga	Manager 3	Manager	[Detail] [Edit] [Hapus]
7	manager9	Manager 9	Manager	[Detail] [Edit] [Hapus]
8	manager22	Manager Dua Dua	Manager	[Detail] [Edit] [Hapus]
9	manager33	Manager Tiga Tiga	Manager	[Detail] [Edit] [Hapus]

- y. Selamat, kalian sudah membuat Laravel Starter Code untuk proses CRUD dengan menggunakan template AdminLTE dan plugin jQuery Datatables.  
z. Jangan lupa commit dan push ke github PWL\_POS kalian

## 4. Praktikum 4: Implementasi *Filtering* Datatables

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

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

```
@if ($session('error'))
    <div class="alert alert-danger">{{ $session('error') }}</div>
@endif
<div class="row">
    <div class="col-md-12">
        <div class="form-group row">
            <label class="col-1 control-label col-form-label">Filter:</label>
            <div class="col-3">
                <select class="form-control" id="level_id" name="level_id" required>
                    <option value="">- Semua -</option>
                    @foreach($level as $item)
                        <option value="{{ $item->level_id }}>{{ $item->level_nama }}</option>
                    @endforeach
                </select>
                <small class="form-text text-muted">Level Pengguna</small>
            </div>
        </div>
    </div>
</div>
<table class="table table-bordered table-striped table-hover table-sm" id="table_user">
```

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

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

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

```

        $('#level_id').on('change', function() {
            dataUser.ajax.reload();
        });
    });
</script>
@endpush

```

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

```

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

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

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

```

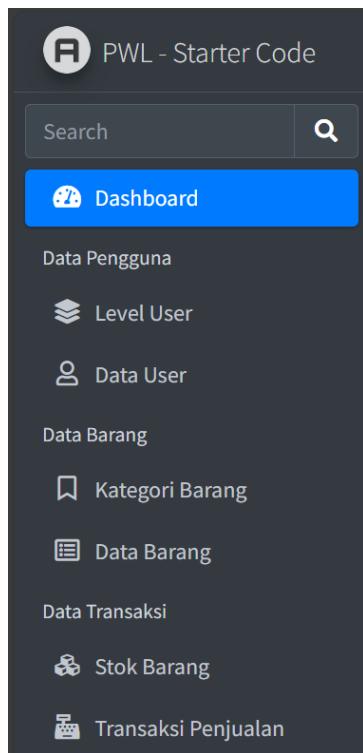
- f. Bagian akhir adalah kita coba jalankan di browser dengan akses menu user, maka akan tampil seperti berikut

ID	Username	Nama	Level Pengguna	Aksi
1	admin	Administrator	Administrator	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
2	manager	Manager	Manager	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
3	staf	Staf/Kasir	Staf/Kasir	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
4	customer-1	Pelanggan Pertama	Pelanggan	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
5	manager_dua	Manager 2	Manager	<button>Detail</button> <button>Edit</button> <button>Hapus</button>

- g. Selamat, sekarang Laravel Starter Code sudah ada filtering dan searching data. Starter Code sudah bisa digunakan dalam membangun sebuah sistem berbasis website.
- h. Jangan lupa commit dan push ke github PWL\_POS kalian

## TUGAS

Implementasikan web layout dan datatables, pada menu berikut ini



✓ Tabel m\_level

ID	Level Kode	Level Nama	aksi
1	ADM	Administrator	[Detail] [Edit] [Hapus]
2	MNG	Manager	[Detail] [Edit] [Hapus]
3	STF	Staf/Kasir	[Detail] [Edit] [Hapus]
4	PLG	Pelanggan	[Detail] [Edit] [Hapus]

✓ Tabel m\_kategori

localhost/PWL\_POS/public/kategori

**Daftar Kategori**

Daftar kategori yang terdaftar dalam sistem

Show 10 entries Search:

ID	Kode Kategori	Nama Kategori	Aksi
1	ELK	Elektronik	[Detail] [Edit] [Hapus]
2	FAS	Fashion	[Detail] [Edit] [Hapus]
3	MKN	Makanan	[Detail] [Edit] [Hapus]
4	ATK	Alat Tulis	[Detail] [Edit] [Hapus]
5	SPRT	Sport	[Detail] [Edit] [Hapus]
6	CML	Cemilan	[Detail] [Edit] [Hapus]
7	MNR	Minuman Ringan	[Detail] [Edit] [Hapus]

✓ Tabel m\_supplier

localhost/PWL\_POS/public/supplier

**Daftar Supplier**

Daftar supplier yang terdaftar dalam sistem

Show 10 entries Search:

ID	Kode Supplier	Nama Supplier	Alamat Supplier	Aksi
1	SP1	Toko Elektronik Jaya	JL. Merdeka No. 45, Bandung	[Detail] [Edit] [Hapus]
2	SP2	Sentral Komputer	JL. Ahmad Yani No. 10, Semarang	[Detail] [Edit] [Hapus]
3	SP3	Toko Grosir Makmur	JL. Raya Sudirman No. 99, Yogyakarta	[Detail] [Edit] [Hapus]

Showing 1 to 3 of 3 entries Previous 1 Next

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

Version 3.2.0

✓ Tabel m\_barang

localhost/PWL\_POS/public/barang

**Data Barang**

Daftar barang yang terdaftar dalam sistem

Tambah

Filter: - Semua -

Kategori Barang

Show 10 entries Search:

ID	Kode Barang	Nama Barang	Nama Kategori	Harga Beli	Harga Jual	Aksi
1	HP001	Handphone	Elektronik	3000000	3500000	[Detail] [Edit] [Hapus]
2	SW001	Sweater	Fashion	150000	200000	[Detail] [Edit] [Hapus]
3	JNS001	Jeans	Fashion	200000	250000	[Detail] [Edit] [Hapus]
4	MSG001	Mie Instan	Makanan	2500	3500	[Detail] [Edit] [Hapus]
5	SUS001	Susu Kotak	Makanan	5000	7000	[Detail] [Edit] [Hapus]