

Tugas Jobsheet 5 Pemrograman Web Lanjut (Revisi)



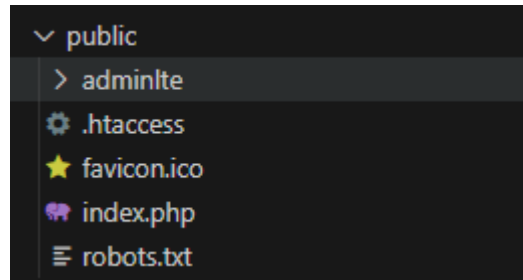
Muhammad Farrel Caesarian
TI-2D/16
2341720053
POLITEKNIK NEGERI MALANG

1. Praktikum 1 - Layouting AdminLTE

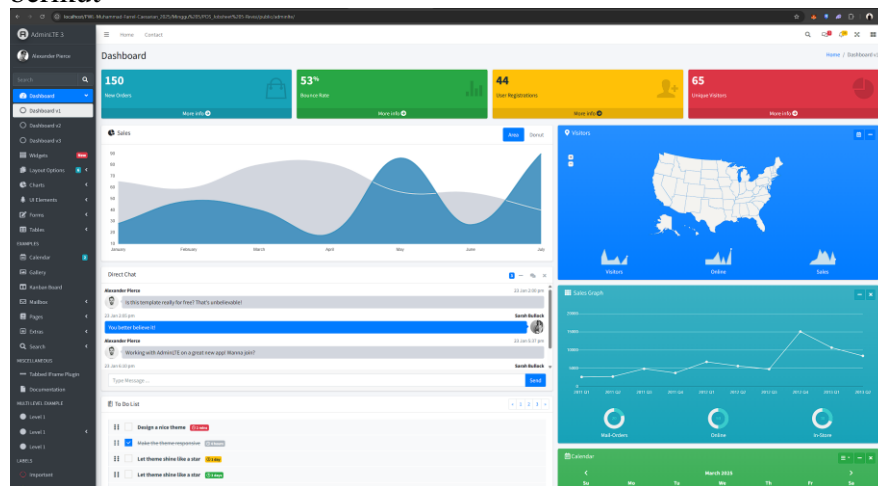
Kita download **AdminLTE v3.2.0** yang rilis pada 8 Feb 2022

▼ Assets 2	
Source code (zip)	Feb 8, 2022
Source code (tar.gz)	Feb 8, 2022

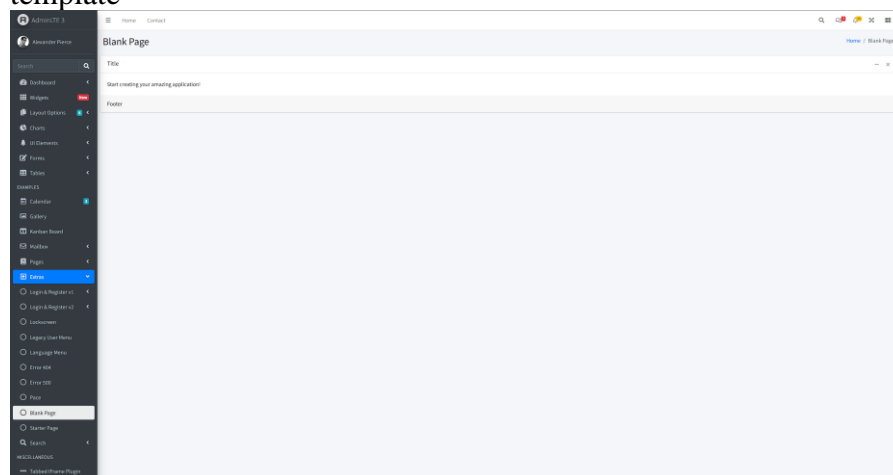
Setelah kita berhasil download, kita ekstrak file yang sudah di download ke folder project PWL_POS/public, kemudian kita **rename** folder cukup menjadi adminlte



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



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



Selanjutnya kita klik kanan halaman Blank Page dan klik *view page source*

```
Line wrap
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <title>AdminLTE 3 | Blank Page</title>
7
8 <!-- Google Font: Source Sans Pro -->
9 <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallba
10 <!-- Font Awesome -->
11 <link rel="stylesheet" href="../../plugins/fontawesome-free/css/all.min.css">
12 <!-- Theme style -->
13 <link rel="stylesheet" href="../../dist/css/adminlte.min.css">
14 </head>
15 <body class="hold-transition sidebar-mini">
16 <!-- Site wrapper -->
17 <div class="wrapper">
18 <!-- Navbar -->
19 <nav class="main-header navbar navbar-expand navbar-white navbar-light">
20 <!-- Left navbar links -->
21 <ul class="navbar-nav">
22 <li class="nav-item">
23 <a class="nav-link" data-widget="pushmenu" href="#" role="button"><i class="fas fa-bars"></i></a>
24 </li>
25 <li class="nav-item d-none d-sm-inline-block">
26 <a href="../../index3.html" class="nav-link">Home</a>
27 </li>
28 <li class="nav-item d-none d-sm-inline-block">
29 <a href="#" class="nav-link">Contact</a>
30 </li>
31 </ul>
32
33 <!-- Right navbar links -->
34 <ul class="navbar-nav ml-auto">
35 <!-- Navbar Search -->
36 <li class="nav-item">
37 <a class="nav-link" data-widget="navbar-search" href="#" role="button">
38 <i class="fas fa-search"></i>
39 </a>
40 <div class="navbar-search-block">
41 <form class="form-inline">
42 <div class="input-group input-group-sm">
43 <input class="form-control form-control-navbar" type="search" placeholder="Search" aria-label="Search">
44 <div class="input-group-append">
45 <button class="btn btn-navbar" type="submit">
46 <i class="fas fa-search"></i>
47 </button>
48 <button class="btn btn-navbar" type="button" data-widget="navbar-search">
49 <i class="fas fa-times"></i>
50 </button>
51 </div>
52 </div>
53 </form>
54 </div>
55 </li>
56
57 <!-- Messages Dropdown Menu -->
58 <li class="nav-item dropdown">
59 <a class="nav-link" data-toggle="dropdown" href="#">
60 <i class="far fa-comments"></i>
61 <span class="badge badge-danger navbar-badge">3</span>
```

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)

```
resources > views > layouts > template.blade.php > html > head
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+
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">
```

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

```
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>
```

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

```

14 </head>
15 <body class="hold-transition sidebar-mini">
16 <!-- Site wrapper -->
17 <div class="wrapper">
18     <!-- Navbar -->
19     @include('layouts.header')
20     <!-- /.navbar -->
21
22     <!-- Main Sidebar Container -->
23     <aside class="main-sidebar sidebar-dark-primary elevation-4">
24         <!-- Brand Logo -->
25         <a href="../../index3.html" class="brand-link">
26             
27             <span class="brand-text font-weight-light">AdminLTE 3</span>
28         </a>

```

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

```

25 <a href="{{ url('/') }}" class="brand-link">
26     
27     <span class="brand-text font-weight-light">PWL Starter Code</span>

```

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

```

30 <!-- Sidebar -->
31 @include('layouts.sidebar')
32 <!-- /.sidebar -->

```

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

```

85 <!-- /.content-wrapper -->
86
87 @include('layouts.footer')
88 </div>
89 <!-- ./wrapper -->
90

```

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

```

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

```

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

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

```
37 <!-- Content Header (Page header) -->
38 @include('layouts.breadcrumb')
39
40 <!-- Main content -->
41 <section class="content">
```

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

```
41 <section class="content">
42
43 <!-- Default box -->
44 @yield('content')
45 <!-- /.card -->
46
47 </section>
48 <!-- /.content -->
```

Jangan lupa commit dan push ke github untuk praktikum 1 ini

Praktikum 1 done
FarrelCaesarian • just now

2. **Praktikum 2 - Penerapan Layouting:**

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 class WelcomeController extends Controller
6 {
7     public function index()
8     {
9         $breadcrumb = (object) [
10             'title' => 'Selamat Datang',
11             'list' => ['Home', 'Welcome']
12         ];
13
14         $activeMenu = 'dashboard';
15
16         return view('welcome', ['breadcrumb' => $breadcrumb, 'activeMenu' => $activeMenu]);
17     }
18 }
```

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

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

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

```
resources > views > layouts > breadcrumb.blade.php > section.content-header
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>
```

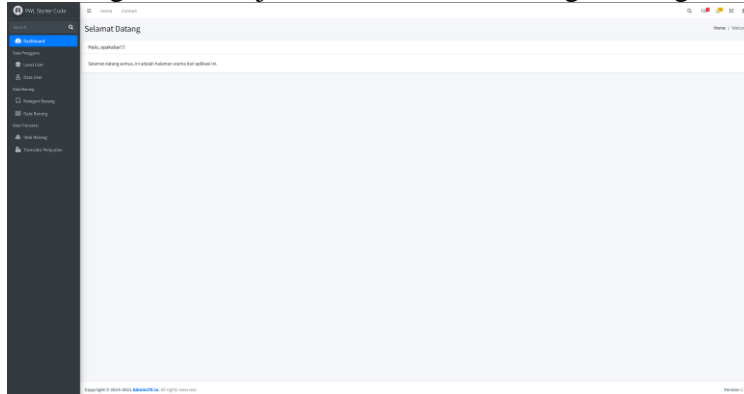
Kita modifikasi file PWL_POS/resources/views/layouts/sidebar.blade.php

```
resources > views > layouts > sidebar.blade.php > div.sidebar
1  <div class="sidebar">
2      <div class="form-inline mt-2">
3          <div class="input-group" data-widget="sidebar-search">
4              <input class="form-control form-control-sidebar" type="search" placeholder
5              <div class="input-group-append">
6                  <button class="btn btn-sidebar">
7                      <i class="fas fa-search fa-fw"></i>
8                  </button>
9              </div>
10         </div>
11     </div>
12     <nav class="mt-2">
13         <ul class="nav nav-pills nav-sidebar flex-column" data-widget="treeview" rol
14             <li class="nav-item">
15                 <a href="{{ url('/') }}" class="nav-link {{ ($activeMenu == 'dashboard')
16                 <i class="nav-icon fas fa-tachometer-alt"></i>
```

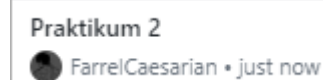
Kita tambahkan kode berikut router web.php

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

Sekarang kita coba jalankan di browser dengan mengetikkan url



Commit Github



3. **Praktikum 3 – Implementasi jQuery Datatable di AdminLTE :**

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

composer require yajra/laravel-datatables:^10.0

Atau

composer require yajra/laravel-datatables-oracle

```
PS C:\laragon\www\PWL-Muhammad-Farrel-Caesarian_2025\Minggu 5\POS_Jobsheet 5-Revisi> composer require yajra/laravel-datatables:^10.0
./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-oracle (v10.0.0)
```

Kita modifikasi route web.php untuk proses CRUD user

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

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

```
app > Http > Controllers > UserController.php > PHP Intelephense > UserController
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\UserModel;
6 use App\Models\UserModel;
7 use Illuminate\Http\Request;
8 use Yajra\DataTables\Facades\DataTables;
9
10 class UserController extends Controller
11 {
12     // Menampilkan halaman awal user
13     public function index()
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', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
27     }
28 }
```

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

```
resources > views > index.blade.php > script > <function> > dataUser
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>
15                    <th>ID</th>
16                    <th>Username</th>
17                    <th>Nama</th>
18                    <th>Level Pengguna</th>
19                    <th>Aksi</th>
20                </tr>
21            </thead>
22        </table>
23    </div>
```

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 > ...
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>A{{ config('app.name', 'PWL Laravel Starter Code') }}</title>
7
8     <meta name="csrf-token" content="{{ csrf_token() }}"> <!-- Untuk mengirimkan token laravel CSRF pada setiap req 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    <!-- Data Tables -->
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 -->
```

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');

    // Filter data 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> ';
            $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 ingin menghapus user ini?')">Hapus</button>
            </form>
            return $btn;
        })
        ->rawColumns(['aksi']) // memberitahu bahwa kolom aksi adalah html
        ->make(true);
}
```


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

ID	Username	Nama	Level/Peran	Aksi
1	admin	Administrator	Administrator	Edit Hapus
2	manager	Manager	Manager	Edit Hapus
3	staff	Staff	Staff	Edit Hapus
4	user1	Manager Perdana	Manager	Edit Hapus
5	manager1	Manager 1	Manager	Edit Hapus
6	manager2	Manager 2	Manager	Edit Hapus
7	manager3	Manager 3	Manager	Edit Hapus
8	manager4	Manager 4	Manager	Edit Hapus
9	manager5	Manager 5	Manager	Edit Hapus
10	manager6	Manager 6	Manager	Edit Hapus

Selanjutnya kita modifikasi UserController.php untuk form tambah data user

```
resources > views > user > 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">{{ $page->title }}</h3>
7 <div class="card-tools"></div>
8 </div>
9 <div class="card-body">
10 <form method="POST" action="{{ url('user') }}" class="form-horizontal">
11 @csrf
12 <div class="form-group row">
13 <label class="col-1 control-label col-form-label">Level</label>
14 <div class="col-11">
15 <select class="form-control" id="level_id" name="level_id" required>
16 <option value="">- Pilih Level -</option>
17 @foreach($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 </form>
26 </div>
27 @endsection
28
```

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

```
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');
}
```

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

The screenshot shows a web form titled 'Tambah User' with fields for 'Level' (a dropdown menu), 'Username' (containing 'prince.cesar_'), 'Nama' (empty), and 'Password' (masked with dots). Below the form is a table with the following data:

ID	Username	Nama	Level Pengguna	Aksi
11	prince.cesar_	Farel Cesaian	Administrator	Detail Edit Hapus

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

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

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

```
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]);
}
```

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

```
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
```

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

The screenshot shows a web page titled 'Detail user'. It contains a red alert box with the following text:

Kesalahan!
Data yang Anda cari tidak ditemukan.

Below the alert box is a button labeled 'Kembali'.

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
```

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

```
public function edit(string $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,
    ]);

    // Menyimpan perubahan data user
    public function update(Request $request, string $id)
    {
        $request->validate([
            'username' => 'required|string|min:3|unique:m_user,
            // username harus diisi, berupa string, minimal 3 k
            // dan bernilai unik di tabel user kolom username k
            'nama' => 'required|string|max:100',
            // nama harus diisi, berupa string, dan maksimal 10
            'password' => 'nullable|min:5',
            // password bisa diisi (minimal 5 karakter) dan bis
            'level_id' => 'required|integer'
            // level_id harus diisi dan berupa angka
        ]);

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

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

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

```
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-dismissable">
12         <h5><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-success">Balik
16 </div>
```

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

Edit User

Edit user

Level

Administrator

Username

admin

Nama

Tes Edit

Password

Abaikan (jangan diisi) jika tidak

Simpan

Kembali

ID	Username	Nama	Level Pengguna
1	admin	Tes Edit	Administrator

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'])`

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

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

    try {
        UserModel::destroy($id); // Hapus data user
        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');
    }
}
```

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

Show 10 of 11 entities

Search

ID	Username	Nama	Level Pengguna	Aksi
1	admin	Tes Edit	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	manager22	Manager Dua Dua	Manager	Detail Edit Hapus
7	manager33	Manager Tiga Tiga	Manager	Detail Edit Hapus
8	manager56	Manager55	Manager	Detail Edit Hapus
9	manager12	Manager11	Manager	Detail Edit Hapus
10	manager11	Manager11	Manager	Detail Edit Hapus

Showing 1 to 10 of 11 entries

Previous 1 2 Next

Data user berhasil dihapus

Filter:

Semua

 jenis pengguna

Show 11 of 11 entities

Search

ID	Username	Nama	Level Pengguna	Aksi
1	admin	Tes Edit	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	manager22	Manager Dua Dua	Manager	Detail Edit Hapus
7	manager33	Manager Tiga Tiga	Manager	Detail Edit Hapus
8	manager56	Manager55	Manager	Detail Edit Hapus
9	manager12	Manager11	Manager	Detail Edit Hapus
10	prince.casuar...	Princi Casuarian	Administrator	Detail Edit Hapus

Showing 1 to 10 of 11 entries

Previous 1 2 Next

Jangan lupa commit dan push ke github PWL_POS kalian

Praktikum 3 Done

 FarrelCaesarian • just now

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

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

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

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

```
{{-- Filtering --}}  
<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">  
    <thead>  
        <tr><th>ID</th><th>Username</th><th>Nama</th><th>Level Pengguna</th><th>Aksi</th></tr>  
    </thead>  
</table>
```

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();  
                }  
            },  
            columns: [  
                {  
                    data: 'ID',  
                    orderable: true,  
                    search: ''  
                },  
                {  
                    data: 'Username',  
                    orderable: true,  
                    search: ''  
                },  
                {  
                    data: 'Nama',  
                    orderable: true,  
                    search: ''  
                },  
                {  
                    data: 'Level Pengguna',  
                    orderable: true,  
                    search: ''  
                },  
                {  
                    data: 'Aksi',  
                    orderable: false,  
                    search: ''  
                }  
            ]  
        });  
        $('#level_id').on('change', function() {  
            dataUser.ajax.reload();  
        });  
    });  
</script>
```

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 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> ';
            $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 d
            return $btn;
        })
        ->rawColumns(['aksi']) // memberitahu bahwa kolom aksi adalah html
        ->make(true);
}
```

Hasil:

Daftar user yang terdaftar dalam sistem

Filter:

Show: entries Search:

ID	Username	Nama	Level Pengguna	Aksi
1	admin	Tes Edit	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	manager22	Manager Dua Dua	Manager	Detail Edit Hapus
7	manager33	Manager Tiga Tiga	Manager	Detail Edit Hapus
8	manager56	Manager55	Manager	Detail Edit Hapus
9	manager12	Manager11	Manager	Detail Edit Hapus
10	prince caesar...	Farrel Caesarian	Administrator	Detail Edit Hapus

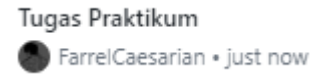
Showing 1 to 10 of 10 entries Previous **1** Next

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

Jangan lupa commit dan push ke github PWL_POS kalian

Praktikum 4

 FarrelCaesarian • just now

5.	G. Tugas Implementasikan web layout dan datatables, pada menu berikut ini ✓ Tabel m_level ✓ Tabel m_kategori ✓ Tabel m_supplier ✓ Tabel m_barang
6.	Jawaban tugas ada di GitHub saya <div>  </div> https://github.com/Quincy17/PWL-Muhammad-Farrel-Caesarian_2025/tree/main/Minggu%205/POS_Jobsheet%205-Revisi