



Nama: Yonanda Mayla Rusdiaty
Kelas: SIB 2A

JOBSHEET 05

Blade View, Web Templating(AdminLTE), Datatables

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

A. Blade View

Laravel Blade adalah template engine yang digunakan secara default oleh Laravel untuk mempermudah pembuatan tampilan (view) dalam aplikasi web. Blade menawarkan fitur-fitur yang powerful dan mudah digunakan untuk memisahkan logika bisnis dari tampilan antarmuka pengguna (UI). Berikut adalah penjelasan lengkap tentang Laravel Blade View:

1. Apa Itu Blade?

Blade adalah template engine yang disertakan dengan Laravel. Dengan Blade, Anda bisa menulis sintaks HTML yang terintegrasi dengan PHP dengan cara yang lebih bersih dan terstruktur. Meskipun menggunakan Blade, semua file template Blade di Laravel tetap memiliki ekstensi *.blade.php*.

2. Keuntungan Menggunakan Blade



- **Inheritance (Pewarisan Template):** Blade memungkinkan Anda untuk membuat layout dasar yang bisa di-extend oleh halaman lain. Ini mempermudah pembuatan struktur halaman yang konsisten.
- **Komponen:** Anda dapat membuat komponen yang reusable untuk elemen UI yang sering digunakan.
- **Control Structures:** Blade mendukung struktur kontrol seperti if, for, while, dan foreach yang mirip dengan sintaks PHP, namun dengan tampilan yang lebih bersih.
- **Tidak Ada Overhead:** Semua kode Blade dikompilasi menjadi PHP biasa, jadi tidak ada overhead tambahan.

3. Sintaks Dasar Blade

Blade memiliki sintaks yang sangat sederhana dan intuitif. Berikut adalah beberapa contoh penggunaan sintaks Blade:

- **Menampilkan Variabel:**

```
{{ $name }}
```

Blade akan secara otomatis melindungi (escape) output untuk mencegah serangan XSS (Cross-Site Scripting).

- **Struktur Kontrol:**

```
@if($condition)
    <p>Condition is true</p>
@endif
```

```
@foreach($users as $user)
    <li>{{ $user->name }}</li>
@endforeach
```

- **Komentar**

```
-- Ini adalah komentar yang tidak akan terlihat dalam output HTML --}}
```



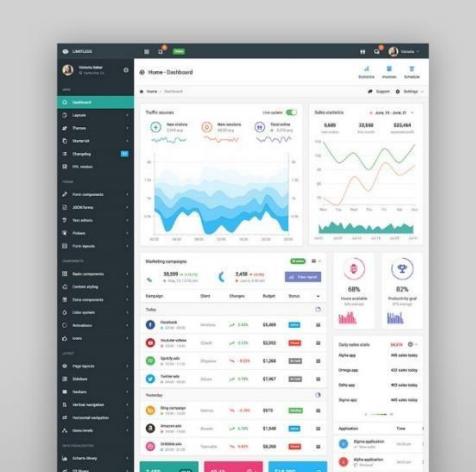
B. Web Template

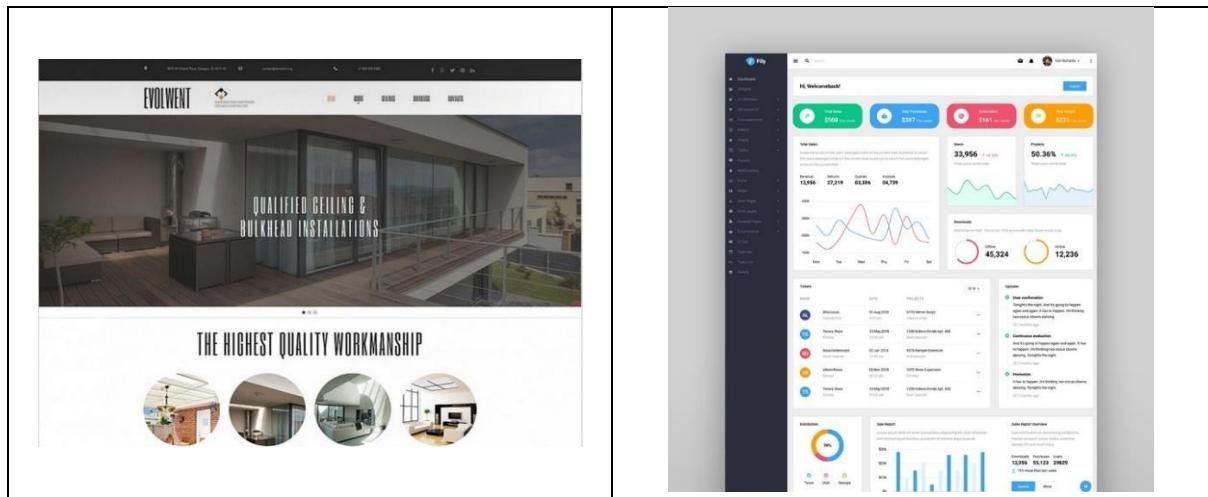
Merupakan suatu file yang sudah memiliki desain tertentu sehingga, kita tinggal memodifikasi bagian-bagian tertentu saja untuk mendapatkan hasil yang sudah bagus. Dengan menggunakan template yang sesuai dengan kebutuhan, kita bisa mengurangi waktu pembuatan suatu website secara drastis.

Secara umum kita bisa menggunakan template website yang sudah ada di internet untuk mempermudah kita dalam membuat website. Ada banyak sekali template yang ada di internet dan kita ambil satu contoh template untuk halaman administrator, seperti **AdminLTE** template. **AdminLTE** adalah salah satu template yang sering digunakan oleh web developer sebagai template backend/admin pada proyek yang sering dikerjakan. Template ini dibuat menggunakan framework bootstrap yang merupakan framework CSS yang paling banyak digunakan di kalangan web desainer

Template AdminLTE dapat di download di <https://adminlte.io>

Contoh Web template

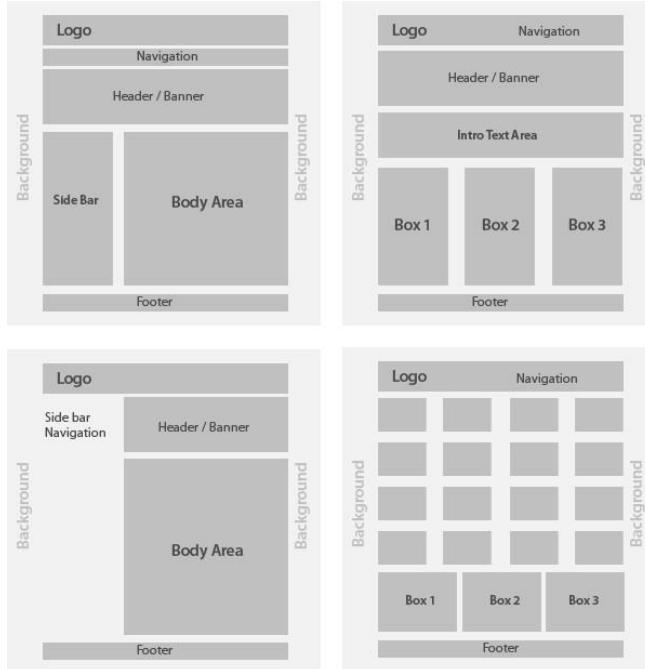
Frontend Template	Backend Template
	



A. Web Layout

Merupakan sekumpulan elemen atau variabel desain yang terkait dengan media tertentu untuk mendukung konsep pembuatan website. Pada dasarnya tujuan penggunaan layout website adalah untuk menggabungkan elemen yang digunakan untuk membangun website. Selain elemen di dalamnya, layout website juga harus memuat beberapa elemen, mulai dari header, navigasi, sidebar, hingga footer. Untuk memahami perbedaan elemen sebuah layout website, berikut penjelasan singkatnya

- **Header** → Pada bagian header, layout designer dapat mengisi bagian ini dengan logo website, navigasi situs, ikon media sosial, dan menu pencarian.
- **Navigasi** → Navigasi dapat dipahami sebagai panduan. Di website, navigasi dapat berkisar dari menu yang muncul di bagian atas halaman web hingga menu pendukung lainnya yang biasa ditemukan di bawah situs.
- **Body/Konten** → Selain itu, terdapat elemen body/konten yang biasanya diisi dengan informasi produk, fitur produk, dan deskripsi produk yang dijual.
- **Sidebar** → elemen ini merupakan elemen yang berada di samping kanan/kiri sebuah website. Element ini kadang sudah tidak dipakai, terutama pada Web Frontend. Namun, sidebar masih digunakan di banyak halaman artikel, yang kemudian akan diisi dengan informasi produk, produk terpopuler, dan navigasi tambahan.
- **Footer** → Elemen terakhir adalah footer. Footer merupakan elemen yang berada dibawah sendiri sebagai menu pendukung atau sebagai identitas dari sebuah website.



C. Layouting AdminLTE

Pada kali ini kita akan melakukan *layouting* template AdminLTE. Tujuan dari *layouting* ini adalah untuk memecah bagian-bagian dari template adminLTE menjadi berbagai elemen. Hal ini dapat mempermudah kita dalam membuat website karena kita bisa menggunakan berulang-ulang elemen yang sudah kita buat nantinya.

Pada layouting kali ini, kita membutuhkan **Blade Template Engine** dari Laravel untuk memecah web menjadi beberapa elemen. Komponen blade yang kita butuhkan untuk proses layouting pada pertemuan kali ini adalah

- `@include()`
- `@extend()`
- `@section()`
- `@push()`
- `@yield()`
- `@stack()`



Praktikum 1 - Layouting AdminLTE:

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

The screenshot shows the GitHub release page for AdminLTE v3.2.0. It displays the release date (Feb 8, 2022), the author (REJack), and the commit hash (bd4d9c7). The main content area features the title "AdminLTE v3.2.0" and a "Latest" button. Below it, there's a "Breaking Change!" note stating that node v12 has been dropped and support for node 12 bugs/errors will not be given. The "Top-Features:" section lists several improvements. The "Contributors" section shows a grid of 17 user icons. The "Assets" section contains two download links: "Source code (zip)" and "Source code (tar.gz)", with the "Source code (zip)" link highlighted by a large red checkmark. Below the assets, there are reaction counts for likes, comments, and stars.

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

The screenshot shows the "Extract Compressed (Zipped) Folders" dialog box from WinRAR. The "Select a Destination and Extract Files" section shows the destination folder path as "D:\Apps\laragon\www\PWL2025\week 5.2\jobsheet\public". There is a checkbox for "Show extracted files when complete".

The screenshot shows the "Extract" dialog box from WinRAR. The file list shows the contents of the "adminlte" folder, which includes ".htaccess", "favicon.ico", "index.php", and "robots.txt". The "Extract" and "Cancel" buttons are visible at the top right.



3. Selanjutnya kita buka di browser dengan alamat

http://localhost/PWL_POS/public/adminlte maka akan muncul tampilan seperti berikut



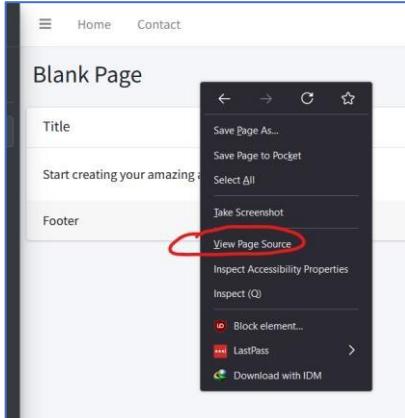
4. Kita klik menu **Extras > Blank Page**, page inilah yang akan menjadi dasar web template

The screenshot shows the AdminLTE 3 dashboard. On the left, there's a sidebar with 'Mailbox' and 'Pages' sections, followed by a 'Extras' section containing various templates like 'Login & Register v1', 'Lockscreen', and 'Blank Page'. The 'Blank Page' option is highlighted with a red box and a number '2' above it. The main content area shows a preview of the 'Blank Page' template with fields for 'Title' and 'Footer'.

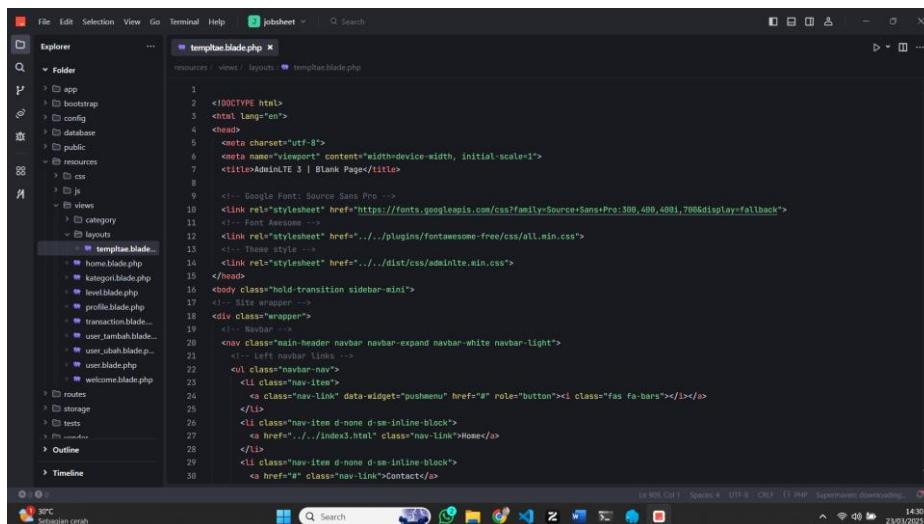
5. Dari sini kita bisa melakukan layouting halaman Blank Page ini menjadi 4 element seperti pada gambar berikut

The screenshot shows the AdminLTE 3 dashboard with the 'Blank Page' template applied. The layout is divided into five sections: '1 - Header / Navbar' at the top, '2 - Sidebar / Navigasi' on the left, '3 - Breadcrumb' just below the header, '4 - Content' in the center, and '5 - Footer' at the bottom. The sidebar and footer are labeled with red boxes and numbers. The main content area contains placeholder text for 'Title', 'Start creating your amazing application!', and 'Footer'.

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



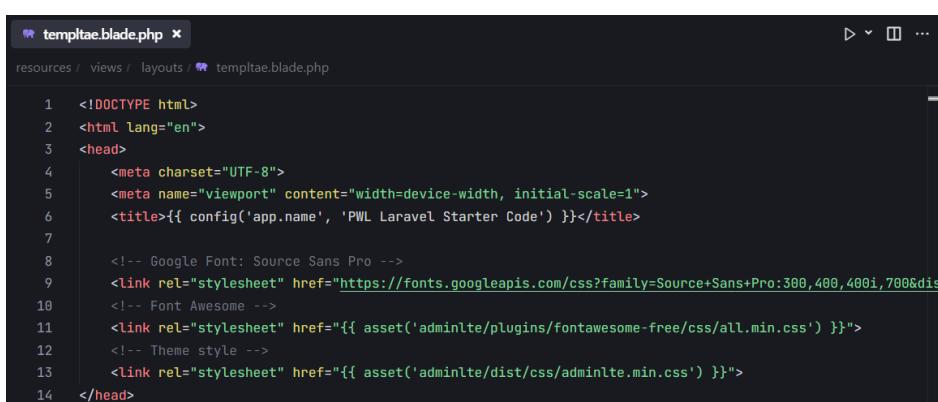
7. Selanjutnya kita copy page source dari halaman **Blank Page**, kemudia kita *paste* pada **PWL_POS/resource/view/layouts/template.blade.php** (buat dulu folder **layouts** dan file **template.blade.php**)



```
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>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=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('dist/css/adminlte.min.css') }}>
14
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       </ul>
26       <ul class="nav-item d-none d-sm-inline-block">
27         <a href="{{ asset('index.html') }}" class="nav-link">Home</a>
28       </ul>
29       <ul class="nav-item d-none d-sm-inline-block">
30         <a href="#" class="nav-link">Contact</a>
31       </ul>
32     </nav>
33   </div>
34 </body>
35 </html>
```

8. File **layouts/template.blade.php** adalah file utama untuk templating website
9. Pada baris **1-14** file **template.blade.php**, kita modifikasi

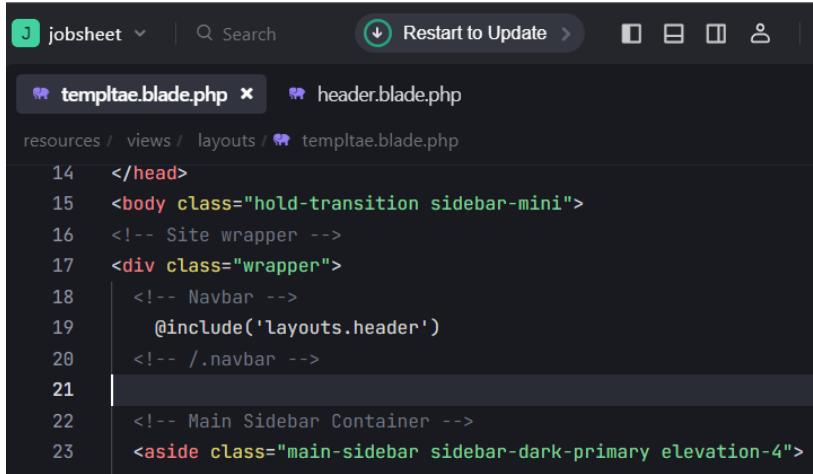


```
resources / views / layouts / template.blade.php

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>{{ 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>
```



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



```
jobsheet | Search | Restart to Update | 
tempitae.blade.php | header.blade.php
resources / views / layouts / tempitae.blade.php
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">
```

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**

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



12. 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

```
resources / views / layouts / templaе.blade.php
17 <div class="wrapper">
23   <aside class="main-sidebar sidebar-dark-primary elevation-4">
30     <!-- Sidebar -->
31     @include('layouts.sidebar')
32     <!-- ./sidebar -->
33   </aside>
```

13. 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

```
resources / views / layouts / templaе.blade.php
17 <div class="wrapper">
85   <!-- ./content-wrapper -->
86
87   @include('layouts.footer')
88 </div>
89 <!-- ./wrapper -->
```

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

```
resources / views / layouts / templaе.blade.php
89 <!-- ./wrapper -->
90
91   <!-- jQuery -->
92   <script src="{{ asset('adminlte/plugins/jquery/jquery.min.js') }}"></script>
93   <!-- Bootstrap 4 -->
94   <script src="{{ asset('adminlte/plugins/bootstrap/js/bootstrap.bundle.min.js') }}"></script>
95   <!-- AdminLTE App -->
96   <script src="{{ asset('adminlte/dist/js/adminlte.min.js') }}"></script>
97 </body>
98 </html>
```



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

```
resources / views / layouts / templaе.blade.php      breadcrumb.blade.php      footer.blade.php
17 <div class="wrapper">
35   <!-- Content Wrapper. Contains page content -->
36   <div class="content-wrapper">
37     <!-- Content Header (Page header) -->
38     @include('layouts.breadcrumb')
```

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')`

```
<!-- Main content -->
<section class="content">
  @yield('content')
</section>
```

19. Hasil akhir pada file utama `layouts/template.blade.php` adalah seperti berikut



```
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>
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="{{url('/')}}" class="brand-link">
26             
27             <span class="brand-text font-weight-light">PWL - Starter Code</span>
28         </a>
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
40         <!-- Main content -->
41         <section class="content">
42             @yield('content')
43         </section>
44         <!-- /.content -->
45     </div>
46     <!-- /.content-wrapper -->
47
48     @include('layouts.footer')
49 </div>
50 <!-- /.wrapper -->
51
52     <!-- jQuery -->
53     <script src="{{ asset('adminlte/plugins/jquery/jquery.min.js') }}></script>
54     <!-- Bootstrap 4 -->
55     <script src="{{ asset('adminlte/plugins/bootstrap/js/bootstrap.bundle.min.js') }}></script>
56     <!-- AdminLTE App -->
57     <script src="{{ asset('adminlte/dist/js/adminlte.min.js') }}></script>
58 </body>
59 </html>
```

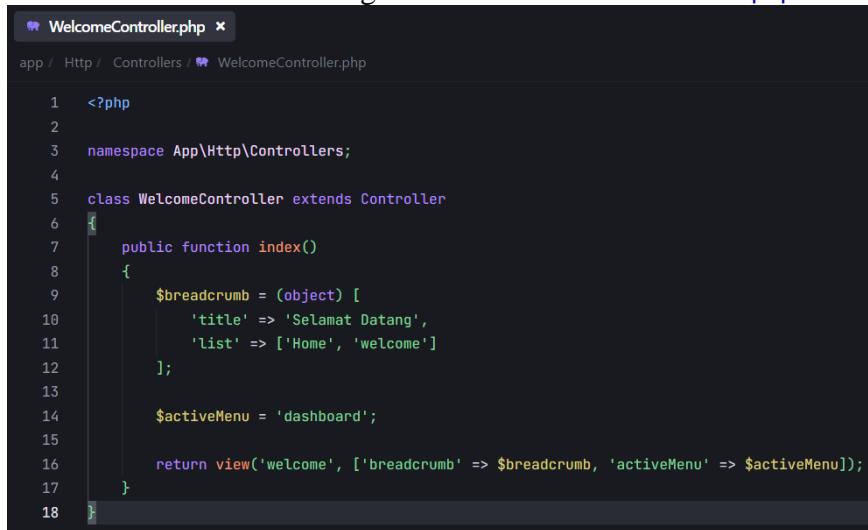


20. Selamat kalian sudah selesai dalam melakukan layouting website di laravel.
21. Jangan lupa commit dan push ke github untuk praktikum 1 ini

Praktikum 2 - Penerapan Layouting:

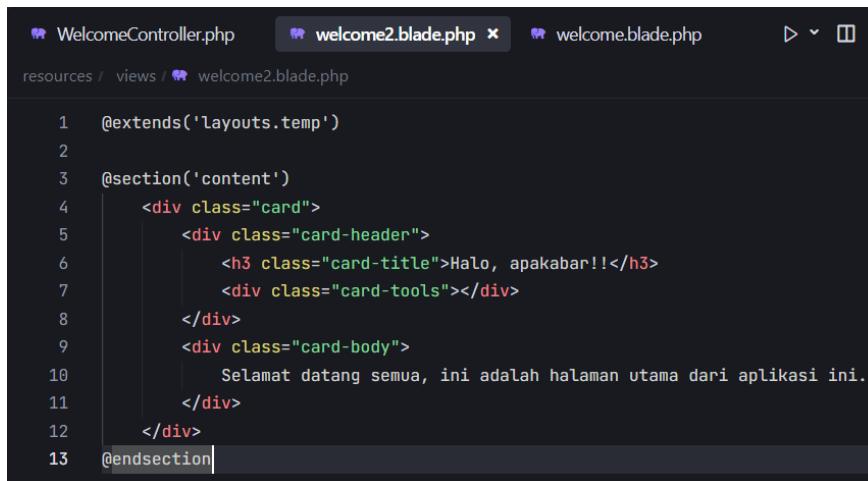
Sekarang kita akan mencoba melakukan penerapan terhadap layouting yang sudah kita lakukan.

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



```
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 }
```

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



```
1 @extends('layouts.temp')
2
3 @section('content')
4     <div class="card">
5         <div class="card-header">
6             <h3 class="card-title">Halo, apakah??</h3>
7             <div class="card-tools"></div>
8         </div>
9         <div class="card-body">
10            Selamat datang semua, ini adalah halaman utama dari aplikasi ini.
11        </div>
12    </div>
13 @endsection
```



3. 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>
```

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

```
resources > views > layouts > sidebar.blade.php > div.sidebar
You, 14 seconds ago | 1 author (You)
1 <div class="sidebar">
2     <!-- SidebarSearch Form -->
3     <div class="form-inline mt-2">
4         <div class="input-group" data-widget="sidebar-search">
5             <input class="form-control form-control-sidebar" type="search" placeholder="Search" aria-label="Search">
6             <div class="input-group-append">
7                 <button class="btn btn-sidebar">
8                     <i class="fas fa-search fa-fw"></i>
9                 </button>
10            </div>
11        </div>
12    </div>
13
14    <!-- Sidebar Menu -->
15    <nav class="mt-2">
16        <ul class="nav nav-pills nav-sidebar flex-column" data-widget="treeview" role="menu" data-accordion="false">
17            <!-- Dashboard -->
18            <li class="nav-item">
19                <a href="{{ url('/') }}" class="nav-link {{ ($activeMenu == 'dashboard') ? 'active' : '' }}>
20                    <i class="nav-icon fas fa-tachometer-alt"></i>
21                    <p>Dashboard</p>
22                </a>
23            </li>
24
25            <!-- Data Pengguna -->
26            <li class="nav-header">Data Pengguna</li>
27            <li class="nav-item">
28                <a href="{{ url('/level') }}" class="nav-link {{ ($activeMenu == 'level') ? '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') ? 'active' : '' }}>
35                    <i class="nav-icon far fa-user"></i>
36                    <p>Data User</p>
37            </li>
38        </ul>
39    </nav>
40
```



5. Kita tambahkan kode berikut router web.php

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

6. Sekarang kita coba jalankan di browser dengan mengetikkan url

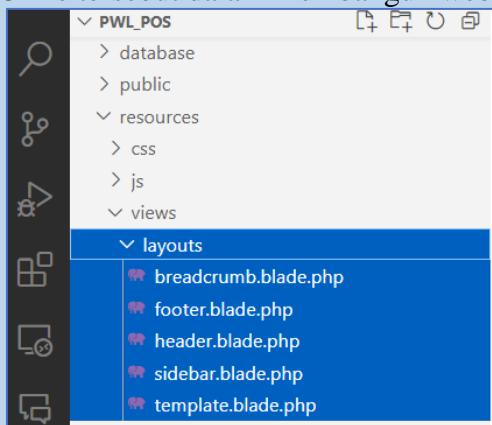
http://localhost/PWL_POS/public



7. Jangan lupa commit dan push ke github PWL_POS kalian

INFO

Terdapat 5 file utama pada folder *layouts* yang digunakan dalam membangun sebuah website, yaitu **template**, **header**, **sidebar**, **breadcrumb**, **footer**. Kalian bisa memodifikasi ke-5 file tersebut dalam membangun website.





D. jQuery Datatables di Laravel

jQuery DataTables adalah sebuah plugin jQuery yang sangat populer dan powerful untuk menampilkan dan mengelola data dalam bentuk tabel di halaman web. Untuk menampilkan banyak data, kita bisa menampilkan data tersebut dalam format tabel. DataTable ini merupakan plugin **jQuery** yang dibuat untuk mengelola data informasi dalam bentuk grid / table.

INFO

AdminLTE juga sudah menerapkan library Datatable dalam template nya. Datatable pada AdminLTE dapat kalian cek di

<http://localhost/PWL/public/adminlte/pages/tables/data.html>

Pada praktikum kali ini kita **tidak menggunakan Vite/NodeJS** dalam mengelola datatables, kita cukup menggunakan jQuery datatables bawaan AdminLTE dan library Yajra *laravel-datatables*.

E. Implementasi jQuery Datatables di Laravel

Kita akan menerapkan penggunaan jQuery datatable dari AdminLTE dengan Laravel. Dalam penerapan ini, kita menggunakan library Yajra laravel-datatables.

Praktikum 3 – Implementasi jQuery Datatable di AdminLTE :

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

```
Route::group(['prefix' => 'user'], function () {
    Route::get('/', [UserController::class, 'index']); // menampilkan halaman awal user
    Route::post('/list', [UserController::class, 'list']); // menampilkan data user dalam bentuk json untuk datatables
    Route::get('/create', [UserController::class, 'create']); // menampilkan halaman form tambah user
    Route::post('/', [UserController::class, 'store']); // menyimpan data user baru
    Route::get('/{id}', [UserController::class, 'show']); // menampilkan detail user
    Route::get('/{id}/edit', [UserController::class, 'edit']); // menampilkan halaman form edit user
    Route::put('/{id}', [UserController::class, 'update']); // menyimpan perubahan data user
    Route::delete('/{id}', [UserController::class, 'destroy']); // menghapus data user
});| You, 16 seconds ago * Uncommitted changes
```



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

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

$page = (object) [
    'title' => 'Daftar user yang terdaftar dalam sistem'
];

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

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



5. Lalu kita buat view pada [PWL/resources/views/user/index.blade.php](#)

```
resources > views > user > 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>
 24  </div>
 25  @endsection
 26
 27  @push('css')
 28  @endpush
 29
 30  @push('js')
 31      <script>
 32          $(document).ready(function () {
 33              var dataUser = $('#table_user').DataTable({
 34                  // serverSide: true, jika ingin menggunakan server side processing serverSide: true,
 35                  ajax: {
 36                      "url": "{{ url('user/list') }}",
 37                      "dataType": "json",
 38                      "type": "POST"
 39                  },
 40                  columns: [
 41                      {
 42                          // nomor urut dari laravel datatable addIndexColumn()
 43                          data: "DT_RowIndex",
 44                          className: "text-center",
 45                          orderable: false
 46                      }
 47                  ]
 48              });
 49          });
 50      </script>
 51  </push>
```



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
    You, 1 second ago | 1 author (You)
1   <!DOCTYPE html>
2   <html lang="en">
3
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     {{-- untuk mengirimkan token laravel CSRF pada setiap request ajax --}}
10    <meta name="csrf-token" content="{{ csrf_token() }}">
11
12    <!-- Google Font: Source Sans Pro -->
13    <link rel="stylesheet"
14      href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">
15    <!-- Font Awesome -->
16    <link rel="stylesheet" href="{{ asset('adminlte/plugins/fontawesome-free/css/all.min.css') }}">
17    {{-- Datatables --}}
18    <link rel="stylesheet" href="{{ asset('adminlte/plugins/datatables-bs4/css/dataTables.bootstrap4.min.css') }}"
19    <link rel="stylesheet" href="{{ asset('adminlte/plugins/datatables-responsive/css/responsive.bootstrap4.min.css') }}"
20    <link rel="stylesheet" href="{{ asset('adminlte/plugins/datatables-buttons/css/buttons.bootstrap4.min.css') }}"
21    <!-- Theme style -->
22    <link rel="stylesheet" href="{{ url('adminlte/dist/css/adminlte.min.css') }}>
23
24    {{-- digunakan untuk memanggil custom css dari perintah push('css') pada masing-masing view --}}
25    @stack('css')
26  </head>    You, 7 hours ago * feat: layouting adminlte

<!-- 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>
<!-- DataTables & Plugins -->
<script src="{{ asset('adminlte/plugins/datatables/jquery.dataTables.min.js') }}></script>
<script src="{{ asset('adminlte/plugins/datatables-bs4/js/dataTables.bootstrap4.min.js') }}></script>
<script src="{{ asset('adminlte/plugins/datatables-responsive/js/dataTables.responsive.min.js') }}></script>
<script src="{{ asset('adminlte/plugins/datatables-responsive/js/responsive.bootstrap4.min.js') }}></script>
<script src="{{ asset('adminlte/plugins/datatables-buttons/js/dataTables.buttons.min.js') }}></script>
<script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.bootstrap4.min.js') }}></script>
<script src="{{ asset('adminlte/plugins/jspdf/jspdf.min.js') }}></script>
<script src="{{ asset('adminlte/plugins/pdfmake/pdfmake.min.js') }}></script>
<script src="{{ asset('adminlte/plugins/pdfmake/vfs_fonts.js') }}></script>
<script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.html5.min.js') }}></script>
<script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.print.min.js') }}></script>
<script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.colVis.min.js') }}></script>
<!-- AdminLTE App -->
<script src="{{ asset('adminlte/dist/js/adminlte.min.js') }}></script>

<script>
  { --untuk mengirimkan token laravel csrf pada setiap ajax-- }
  $.ajaxSetup({ headers: { 'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content') } });
</script>

{{-- digunakan untuk memanggil custom js dari perintah push('js') pada masing-masing view --}}
@stack('js')
</body>
</html>
```



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

```
1 reference | 0 overrides
public function list(Request $request): JsonResponse
{
    // Ambil data user dengan kolom yang diperlukan dan relasi 'level'
    $users = UserModel::select('user_id', 'username', 'nama', 'level_id')
        ->with('level');

    return DataTables::of($users)
        // Tambahkan kolom index/no urut (default: DT_RowIndex)
        ->addIndexColumn()
        // Tambahkan kolom aksi dengan tombol Detail, Edit, dan Hapus
        ->addColumn('aksi', function ($user): string {
            $btn = ' <a href="' . url("/user/{$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?`)">Hapus</button>';
            $btn .= '</form>';

            return $btn;
        })
        // Beritahu DataTables bahwa kolom 'aksi' berisi HTML
        ->rawColumns(['aksi'])
        ->make(true);
}
```

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

Jawab: Berikut adalah tampilannya. Saat saya mengakses /user yg mengarah ke index di usercontroller, maka akan ditampilkan data user dalam bentuk json untuk datatables. Selanjutnya pada index tersebut dia akan membuat data breadcrumb (title: "Daftar User" dan list: Array berisi "Home" dan "User"), Membuat Data Halaman (title: "Daftar user yang terdaftar dalam sistem), Mengatur Menu Aktif (activeMenu: "user"), dan merender view di index.blade

ID	Username	Nama	Level Pengguna	Aksi
1	admin	Administrator	Administrator	Detail Edit Hapus
2	manager	Manager	Manager	Detail Edit Hapus
3	staff	Staf/Kasir	Staf/Kasir	Detail Edit Hapus
4	customer-1	Customer 1	Customer	Detail Edit Hapus
5	manager_dua	Manager 2	Manager	Detail Edit Hapus
6	manager33	Manager Tiga Tiga	Manager	Detail Edit Hapus
7	manager45	Manager44	Manager	Detail Edit Hapus
8	manager44	Manager44	Manager	Detail Edit Hapus
9	manager56	Manager55	Manager	Detail Edit Hapus
10	manager12	Manager11	Manager	Detail Edit Hapus



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

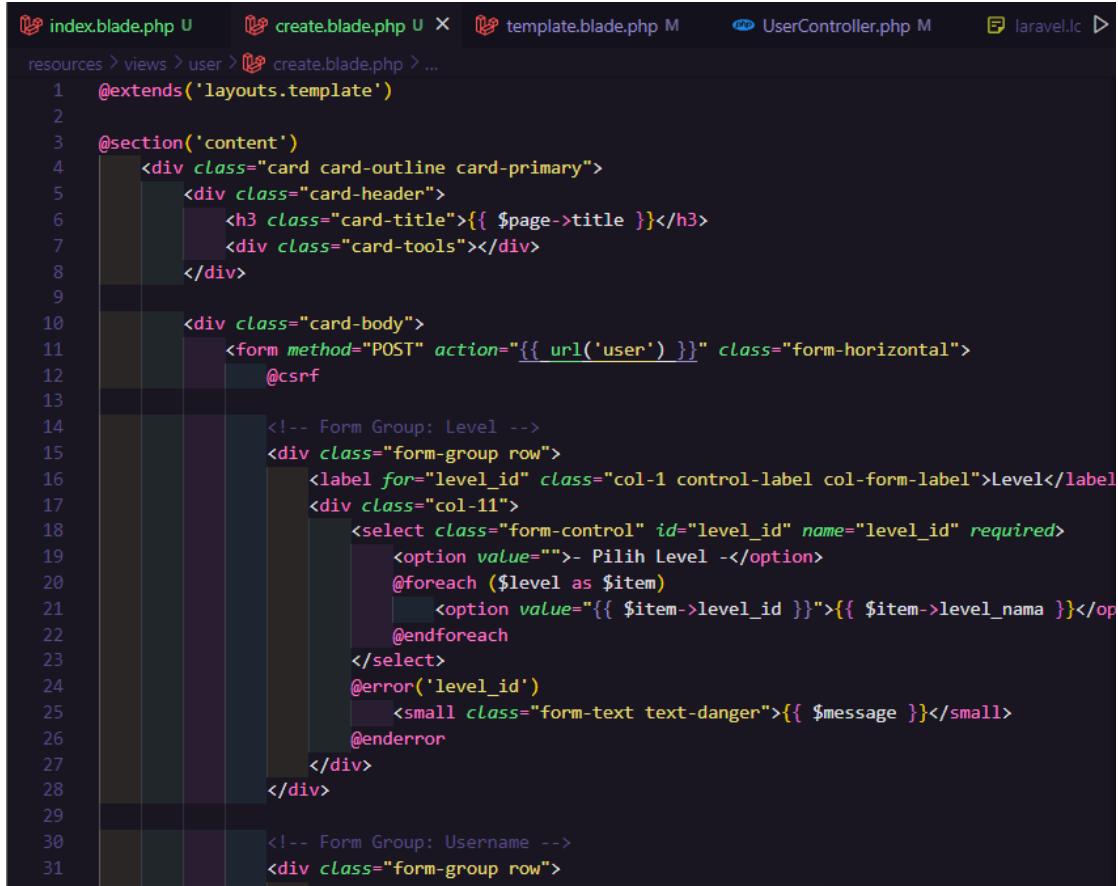
```
1 reference | 0 overrides
public function create(): Factory|View
{
    $breadcrumb = (object) [
        'title' => 'Tambah User',
        'list' => ['Home', 'User', 'Tambah']
    ];

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

    $level = LevelModel::all(); // ambil data level untuk ditampilkan di form
    $activeMenu = 'user'; // set menu yang sedang aktif
    return view('user.create', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' =>
} You, 1 se
```



10. Sekarang kita buat form untuk menambah data, kita buat file PWL/resources/views/user/create.blade.php



```
index.blade.php U create.blade.php X template.blade.php M UserController.php M laravel.lc ▶
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">{{ $page->title }}</h3>
7             <div class="card-tools"></div>
8         </div>
9
10        <div class="card-body">
11            <form method="POST" action="{{ url('user') }}" class="form-horizontal">
12                @csrf
13
14                <!-- Form Group: Level -->
15                <div class="form-group row">
16                    <label for="level_id" class="col-1 control-label col-form-label">Level</label>
17                    <div class="col-11">
18                        <select class="form-control" id="level_id" name="level_id" required>
19                            <option value="">- Pilih Level -</option>
20                            @foreach ($level as $item)
21                                <option value="{{ $item->level_id }}>{{ $item->level_nama }}</option>
22                            @endforeach
23                        </select>
24                        @error('level_id')
25                            <small class="form-text text-danger">{{ $message }}</small>
26                        @enderror
27                    </div>
28                </div>
29
30                <!-- Form Group: Username -->
31                <div class="form-group row">
```



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

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

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

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

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

Jawab: Berikut merupakan tampilannya. Ketika kita mengakses route /user/create, maka usercontroller akan menjalankan fungsi create() untuk menampilkan halaman form tambah user baru dan strore() untuk menyimpan data user baru

The screenshot shows a web application interface. On the left is a sidebar with a dark theme containing a navigation menu. The 'Data User' option is highlighted with a blue background. The main content area has a title 'Tambah User' and a subtitle 'Tambah user baru'. It contains four input fields: 'Level' (a dropdown menu currently showing '- Pilih Level -'), 'Username' (an input field), 'Nama' (an input field), and 'Password' (an input field). At the bottom of the form are two buttons: 'Simpan' (Save) and 'Kembali' (Back).

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



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

```
1 reference | 0 overrides
public function show($id): Factory|View
{
    $user = UserModel::with('level')->find($id); // ambil data user berdasarkan 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,
        'activeMenu' => $activeMenu,
        'user' => $user // tambahkan variabel $user ke array data
    ]);
}
```

15. 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">{{ $page->title }}</h3>
7               <div class="card-tools"></div>
8           </div>
9
10          <div class="card-body">
11              @empty($user)
12                  <!-- Pesan error jika data user tidak ditemukan -->
13                  <div class="alert alert-danger alert-dismissible">
14                      <h5><i class="icon fas fa-ban"></i> Kesalahan!</h5>
15                      Data yang Anda cari tidak ditemukan.
16                  </div>
17              @else
18                  <!-- Tabel untuk menampilkan detail user -->
19                  <table class="table table-bordered table-striped table-hover table-sm">
20                      <tr>
21                          <th>ID</th>
22                          <td>{{ $user->user_id }}</td>
23                      </tr>
24                      <tr>
25                          <th>Level</th>
26                          <td>{{ $user->level->level_nama }}</td>
27                      </tr>
28                      <tr>
29                          <th>Username</th>
30                          <td>{{ $user->username }}</td>
31                      </tr>

```



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

Jawab: Berikut adalah tampilannya.

ID	10
Level	Manager
Username	manager12
Nama	Manager11
Password	*****

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

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

```
// Menampilkan halaman form edit user
1 reference | 0 overrides
public function edit(string $id): Factory|View
{
    $user = UserModel::find($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
    ]);
}
```



```
// Menyimpan perubahan data user
1 reference | 0 overrides
public function update(Request $request, string $id): mixed|RedirectResponse
{
    $request->validate([
        // username harus diisi, berupa string, minimal 3 karakter,
        // dan bernilai unik di tabel m_user kolom username kecuali user dengan id yang
        'username' => 'required|string|min:3|unique:m_user,username,' . $id . ',user_i
        // nama harus diisi, berupa string, dan maksimal 100 karakter
        'nama' => 'required|string|max:100',
        // password nullable (boleh kosong), 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::fin
        'level_id' => $request->level_id
    ]);

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

You, 1 second ago • Uncommitted changes

19. 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">{{ $page->title }}</h3>
7          <div class="card-tools"></div>
8      </div>
9
10     <div class="card-body">
11         @empty($user)
12             <!-- Pesan error jika data user tidak ditemukan -->
13             <div class="alert alert-danger alert-dismissible">
14                 <h5><i class="icon fas fa-ban"></i> Kesalahan!</h5>
15                 Data yang Anda cari tidak ditemukan.
16             </div>
17             <a href="{{ url('user') }}" class="btn btn-sm btn-default mt-2">Kembali</a>
18         @else
19             <!-- Form untuk mengedit data user -->
20             <form method="POST" action="{{ url('/user/' . $user->user_id) }}" class="form-h
21                 @csrf
22                 {!! method_field('PUT') !!}<!-- Untuk proses edit yang membutuhkan method
23
24                 <!-- Form Group: Level -->
25                 <div class="form-group row">
26                     <label for="level_id" class="col-1 control-label col-form-label">Level<
27                     <div class="col-11">
28                         <select class="form-control" id="level_id" name="level_id" required>
29                             <option value="">- Pilih Level -</option>
30                             @foreach ($level as $item)
31                                 <option value="{{ $item->level_id }}" @if ($item->level_id
```



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141
Telp. (0341) 404424 – 404425, Fax (0341) 404420
<http://www.polinema.ac.id>



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

Jawab: Berikut merupakan tampilannya. Ketika kita mengakses /user/id/edit maka akan menampilkan halaman form edit user dan jika kita menyimpannya maka akan ke route /update maka akan menyimpan perubahan data user

Disini saya mencoba mengedit level dari id 11, dari admin menjadi cust. Tampilannya akan seperti ini

The screenshot shows two pages of a web application. The top page is titled 'Edit User' and has a form with fields for Level (set to Administrator), Username (2341760184), Name (Yonanda), and Password (empty). Below the form are 'Simpan' and 'Kembali' buttons. The bottom page is titled 'Daftar User' and lists users with columns for ID, Username, Name, and Level Pengguna. It includes a 'Tambah' button and a table with three rows. The table has a 'Detail' button for each row.

ID	Username	Nama	Level Pengguna	Aksi
11	2341760184	Yonanda	Customer	Detail Edit Hapus
12	2241760013	Yonanda Mayla Rusdiaty	Manager	Detail Edit Hapus
13	customer-2	Customer 2	Customer	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}', [UserController::class, 'destroy']);
```

22. Jadi kita buat fungsi `destroy()` pada `UserController.php`



```
1 reference | 0 overrides
public function destroy(string $id): mixed|RedirectResponse
{
    $check = UserModel::find($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 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
        // return redirect('/user')->with('error', 'Data user gagal dihapus karena masih
    }
}
```

23. Selanjutnya kita modifikasi file [PWL/resources/views/user/index.blade.php](#) untuk menambahkan tampilan jika ada pesan error

```
<div class="card-body">
@if (session('success'))
    <div class="alert alert-success">{{ session('success') }}</div>
@endif

@if (session('error'))
    <div class="alert alert-danger">{{ session('error') }}</div>
@endif

<table class="table table-bordered table-striped table-hover table-sm" id="tab">
    <thead>
        <tr>
            <th>ID</th>
            <th>Username</th>
            <th>>Nama</th>
            <th>Level Pengguna</th>
            <th>Aksi</th>
        </tr>
    </thead>
</table>
</div>
```



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

Jawab: Berikut merupakan tampilannya. Disini saya akan menghapus id 1.

Daftar User Home / User

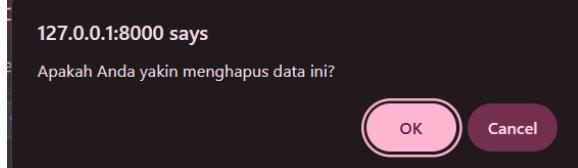
Daftar user yang terdaftar dalam sistem Tambah

Show 10 entries Search:

ID	Username	Nama	Level Pengguna	Aksi
11	2341760184	Yonanda	Customer	Detail Edit Hapus
12	2241760013	Yonanda Mayla Rusdiaty	Manager	Detail Edit Hapus
13	customer-2	Customer 2	Customer	Detail Edit Hapus

Showing 11 to 13 of 13 entries Previous 1 2 Next

Saat saya menekan button hapus, maka akan muncul alert seperti ini



Jika berhasil maka akan menampilkan pesan sukses berikut

Home Contact Search

Daftar User Home / User

Daftar user yang terdaftar dalam sistem Tambah

Data user berhasil dihapus

Show 10 entries Search:

ID	Username	Nama	Level Pengguna	Aksi
1	admin	Administrator	Administrator	Detail Edit Hapus
2	manager	Manager	Manager	Detail Edit Hapus
3	staff	Staf/Kasir	Staf/Kasir	Detail Edit Hapus
4	customer-1	Customer 1	Customer	Detail Edit Hapus
5	manager_dua	Manager 2	Manager	Detail Edit Hapus
6	manager33	Manager Tiga Tiga	Manager	Detail Edit Hapus
7	manager45	Manager44	Manager	Detail Edit Hapus
8	manager44	Manager44	Manager	Detail Edit Hapus
9	manager56	Manager55	Manager	Detail Edit Hapus
10	manager12	Manager11	Manager	Detail Edit Hapus

Showing 1 to 10 of 12 entries Previous 1 2 Next



-
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

F. Data Searching dan Filtering

Dalam menampilkan sebuah data, kadang kita perlu yang namanya melakukan pencarian (*searching*) berdasarkan kata-kunci (*keyword*) tertentu ataupun penyaringan data berdasarkan suatu kategori (*filtering*).

1. **Searching** (Pencarian):

- Searching adalah proses mencari informasi atau data tertentu yang sesuai dengan kriteria yang diberikan.
- Biasanya, pencarian dilakukan dengan menggunakan kata kunci atau frasa tertentu untuk mencocokkan dengan data yang ada.
- Tujuan dari pencarian adalah untuk menemukan entitas yang cocok dengan kriteria pencarian, baik itu dalam basis data, dalam dokumen teks, atau dalam konteks lainnya.
- Pencarian sering kali melibatkan pencocokan pola atau kata kunci dalam teks atau data yang ada, dan hasilnya mungkin mencakup entitas yang sebagian cocok atau mirip dengan kriteria pencarian.

2. **Filtering** (Penyaringan):

- Filtering adalah proses memilih atau membatasi sekumpulan data atau entitas berdasarkan kriteria tertentu.
- Biasanya, filtering dilakukan untuk menyaring data yang sudah ada berdasarkan atribut atau karakteristik tertentu, seperti tanggal, kategori, atau atribut lainnya.



- Tujuan dari penyaringan adalah untuk menyajikan data yang relevan atau relevan dengan kebutuhan pengguna atau kriteria tertentu.
- Filtering dapat dilakukan dengan memilih data berdasarkan nilai yang sesuai dengan kriteria tertentu atau dengan mengecualikan data yang tidak memenuhi kriteria tersebut.

Perbedaan utama antara **searching** dan **filtering** adalah bahwa **searching** berkaitan dengan pencarian data yang sesuai dengan kriteria tertentu, sementara **filtering** berkaitan dengan penyaringan data yang sudah ada berdasarkan kriteria tertentu. Meskipun keduanya sering digunakan bersama-sama dalam aplikasi atau sistem informasi, mereka melayani tujuan yang berbeda dalam proses pengelolaan dan analisis data.

Secara default, jQuery datatables sudah memiliki fitur searching yang sudah terintegrasi dengan laravel yajra-datatables. Akan tetapi, belum ada untuk proses **filtering** datanya. Untuk itu kita akan membuat filtering data pada Laravel Starter Code kita.

The screenshot shows a Laravel application interface. On the left is a sidebar with navigation links: Home, Contact, Dashboard, Data Pengguna, Level User, Data User (which is highlighted in blue), Data Barang, Kategori Barang, Data Barang, Data Transaksi, Stok Barang, and Transaksi Penjualan. The main content area has a header "Daftar User" and a sub-header "Daftar user yang terdaftar dalam sistem". It features a search bar with the placeholder "Search:" and a "Searching" button. Below is a table with columns: ID, Username, Nama, Level Pengguna, and Aksi (with buttons for Detail, Edit, and Hapus). The table contains 6 entries. At the bottom, it says "Showing 1 to 6 of 6 entries" and includes "Previous" and "Next" buttons.

Praktikum 4 – Implementasi **Filtering** Datatables:

Kita akan menerapkan filtering pada datatable yang sudah kita buat. Hal ini akan mempermudah kita dalam mengelompokkan suatu data sesuai kategori tertentu. Langkah-langkah yang kita kerjakan sebagai berikut

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



```
2 references | 0 overrides
public function index(): Factory|View
{
    $breadcrumb = (object) [
        'title' => 'Daftar User',
        'list' => ['Home', 'User']
    ];

    $page = (object) [
        'title' => 'Daftar user yang terdaftar dalam sistem'
    ];

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

    $level = LevelModel::all(); // ambil data level untuk ditampilkan di form

    return view('user.index', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMen
}
```

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

```
resources > views > user > index.blade.php > div.card.card-outline.card-primary > div.card-body > div.row
4   v class="card card-outline card-primary">
5     <div class="card-header">
10    </div>
11    <div class="card-body">
12      @if (session('success'))
13        <div class="alert alert-success">{{ session('success') }}</div>
14      @endif
15
16      @if (session('error'))
17        <div class="alert alert-danger">{{ session('error') }}</div>
18      @endif
19      <div class="row">
20        <div class="col-md-12">
21          <div class="form-group row">
22            <label class="col-1 control-label col-form-label">Filter:</label>
23            <div class="col-3">
24              <select class="form-control" id="level_id" name="level_id" required>
25                <option value="">- Semua -</option>
26                @foreach ($level as $item)
27                  <option value="{{ $item->level_id }}>{{ $item->level_nama }}</option>
28                @endforeach
29              </select>
30              <small class="form-text text-muted">Level Pengguna</small>
31            </div>
32          </div>
33        </div>
34      </div>| You, 1 second ago • Uncommitted changes
35      <table class="table table-bordered table-striped table-hover table-sm" id="table_user">
36        <thead>
37          <tr>
```



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

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



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

```
$('#level_id').on('change', function () {
    dataUser.ajax.reload();
});
```

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

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



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

ID	Username	Nama	Level Pengguna	Aksi
1	admin	Administrator	Administrator	[Detail] [Edit] [Hapus]
2	manager	Manager	Manager	[Detail] [Edit] [Hapus]
3	staff	Staf/Kasir	Staf/Kasir	[Detail] [Edit] [Hapus]
4	customer-1	Customer 1	Customer	[Detail] [Edit] [Hapus]
5	manager_dua	Manager 2	Manager	[Detail] [Edit] [Hapus]
6	manager33	Manager Tiga Tiga	Manager	[Detail] [Edit] [Hapus]
7	manager45	Manager44	Manager	[Detail] [Edit] [Hapus]
8	manager44	Manager44	Manager	[Detail] [Edit] [Hapus]
9	manager56	Manager55	Manager	[Detail] [Edit] [Hapus]
10	manager12	Manager11	Manager	[Detail] [Edit] [Hapus]

7. Selamat, sekarang Laravel Starter Code sudah ada filtering dan searching data. Starter Code sudah bisa digunakan dalam membangun sebuah sistem berbasis website.
8. Jangan lupa commit dan push ke github PWL_POS kalian

G. Tugas

Implementasikan web layout dan datatables, pada menu berikut ini

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



Jawab:

1. Untuk mengimplementasikan web layout dan DataTables pada menu Tabel m_level, berikut adalah langkah-langkahnya:
 - a. Tambahkan route untuk menu Tabel m_level di file web.php

```
// Tambahkan route grup untuk level
Route::group(['prefix' => 'level'], function () {
    Route::get('/', [LevelController::class, 'index']); // Menampilkan halaman tabel level
    Route::post('/list', [LevelController::class, 'list']); // Mengambil data level untuk DataTables
    Route::get('/create', [LevelController::class, 'create']); // Menampilkan form tambah level
    Route::post('/', [LevelController::class, 'store']); // Menyimpan data level baru
    Route::get('{id}', [LevelController::class, 'show']); // Menampilkan detail level
    Route::get('{id}/edit', [LevelController::class, 'edit']); // Menampilkan form edit level
    Route::put('{id}', [LevelController::class, 'update']); // Menyimpan perubahan level
    Route::delete('{id}', [LevelController::class, 'destroy']); // Menghapus level
});
```

1. Kemudian tambahkan method berikut di LevelController:
 - 1) Index(): menampilkan halaman tabel level
 - 2) List(): menyediakan data level dalam format JSON untuk DataTables, termasuk filtering dan searching
 - 3) Show(): menampilkan detail level
 - 4) Edit(): Untuk menampilkan form edit level
 - 5) Update(): Untuk menyimpan perubahan data level
 - 6) Destroy: Untuk menghapus level
 - 7) Create(): Untuk menampilkan form tambah level
 - 8) Store(): Untuk menyimpan data level baru



```
1 // PHP
2
3 namespace App\UI\controllers;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Facades\DB;
7 use Illuminate\Support\Facades\Validator;
8 use App\Models\LevelModel; // Import the LevelModel class
9
10 class LevelController extends Controller
11 {
12     // menampilkan tampilan awal (level)
13     public function index()
14     {
15         $breadcrumb = (object) [
16             'title' => 'Bantul level',
17             'list' => ['home', 'level']
18         ];
19
20         $page = (object) [
21             'title' => 'Bantul level pengguna'
22         ];
23
24         $activeMenu = 'level'; // Set menu yang sedang aktif
25
26         return view('level.index', [
27             'breadcrumb' => $breadcrumb,
28             'page' => $page,
29             'activeMenu' => $activeMenu
30         ]);
31     }
32
33     // menampilkan detail bantul level dengan IDN untuk DataTables, termasuk filtering dan searching
34     public function list(Request $request)
35     {
36         $levels = LevelModel::query();
37
38         // Filter berdasarkan level code (like ini)
39         if ($request->has('level_code')) {
40             $levels->where('level_code', $request->level_code);
41         }
42
43         return Datatables::of($levels)
44             ->addColumn('aksi', function ($level) {
45                 $button = 'Edit | Delete | Detail';
46                 $button .= '<br>' . url("level/edit/{$level->id}") . ' | ' . url("level/delete/{$level->id}") . ' | ' . url("level/detail/{$level->id}");
47                 $button .= '<div class="d-inline-block" style="margin-left: 10px;">' . url("level/{$level->id}") . '</div>';
48                 $button .= '<br>' . $button;
49                 $button .= '<form style="margin-top: 10px;">' . $button . '</form>';
50                 $button .= '<br>' . $button;
51             })
52             ->rawColumns(['aksi']); // Pastikan kolom aksi dirender sebagai HTML
53             ->make(true);
54     }
55
56     // menampilkan detail bantul
57     public function show($id)
58     {
59         $level = LevelModel::find($id); // Ambil data level berdasarkan ID
60
61         $breadcrumb = (object) [
62             'title' => 'Detail level',
63             'list' => ['home', 'level', 'Detail']
64         ];
65
66         $page = (object) [
67             'title' => 'Detail level'
68         ];
69
70         $activeMenu = 'level'; // Set menu yang sedang aktif
71
72         return view('level.show', [
73             'breadcrumb' => $breadcrumb,
74             'page' => $page,
75             'activeMenu' => $activeMenu,
76             'level' => $level // Ambilkan variabel $level ke array data
77         ]);
78     }
79
80     // Untuk mengambil form edit bantul
81     public function edit($id)
82     {
83         $level = LevelModel::find($id); // Ambil data level berdasarkan ID
84
85         $breadcrumb = (object) [
86             'title' => 'Edit level',
87             'list' => ['home', 'level', 'Edit']
88         ];
89
90         $page = (object) [
91             'title' => 'Edit level'
92         ];
93
94         $activeMenu = 'level'; // Set menu yang sedang aktif
95
96         return view('level.edit', [
97             'breadcrumb' => $breadcrumb,
98             'page' => $page,
99             'activeMenu' => $activeMenu,
100             'level' => $level // Kiran data level ke view
101         ]);
102     }
103
104     // Untuk mengambil parameter data level
105     public function update(Request $request, $id)
106     {
107         $request->validate([
108             'level_name' => 'required|string|max:100', // Validasi nama level
109             'level_code' => 'required|string|max:10', // Validasi code level
110         ]);
111
112         $level = LevelModel::find($id);
113
114         $level->update([
115             'level_name' => $request->level_name,
116             'level_code' => $request->level_code
117         ]);
118
119         return redirect('/level')->with('success', 'Level berhasil diperbarui.');
120     }
121
122     // Untuk mengambil form tambah level
123     public function create()
124     {
125         $breadcrumb = (object) [
126             'title' => 'Tambah level',
127             'list' => ['home', 'level', 'Tambah']
128         ];
129
130         $page = (object) [
131             'title' => 'Tambah level baru'
132         ];
133
134         $activeMenu = 'level'; // Set menu yang sedang aktif
135
136         return view('level.create', [
137             'breadcrumb' => $breadcrumb,
138             'page' => $page,
139             'activeMenu' => $activeMenu
140         ]);
141     }
142
143     // Untuk menyimpan form tambah level
144     public function store(Request $request)
145     {
146         $request->validate([
147             'level_name' => 'required|string|max:100', // Validasi nama level
148             'level_code' => 'required|string|max:10', // Validasi code level
149         ]);
150
151         LevelModel::create([
152             'level_name' => $request->level_name,
153             'level_code' => $request->level_code
154         ]);
155
156         return redirect('/level')->with('success', 'Level berhasil ditambahkan.');
157     }
158 }
```



- c. Buat View level dengan beberapa file berikut:

1) Index.blade

```
extends('layouts.template')

@section('content')
    <div class="Card card-outline card-primary">
        <div class="card-header">
            <div class="card-title">({ $page->title })</div>
            <div class="card-tools">
                <a class="btn btn-sm btn-primary mt-1" href="{{ url('level/create') }}">>Tambah</a>
            </div>
        </div>
        <div class="card-body">
            @if ($session['success'])
                <div class="alert alert-success">({ session('success') })</div>
            @endif

            @if ($session['error'])
                <div class="alert alert-danger">({ session('error') })</div>
            @endif
        </div>
        <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_kode" name="level_kode" required>
                            <option value=""> Semua </option>
                            <option value="ADM">Administrator</option>
                            <option value="PMO">Manager</option>
                            <option value="SIT">Sistem Admin</option>
                            <option value="GHS">Customer</option>
                        </select>
                        <small class="form-text text-muted">Kode Level</small>
                    </div>
                </div>
            </div>
        </div>

        <table class="table table-bordered table-striped table-hover" table-sm" id="table_level">
            <thead>
                <tr>
                    <th>Name</th>
                    <th>Kode Level</th>
                    <th>Nama Level</th>
                    <th>aksi</th>
                </tr>
            </thead>
            <tbody>
            </tbody>
        </table>
    </div>
</div>
@endsection

@push('css')
    @push('js')
        <script>
            $(document).ready(function () {
                var tableLevel = $('#table_level').DataTable({
                    processing: true,
                    serverSide: true,
                    ajax: {
                        url: '{{ url('level/list') }}',
                        type: "POST",
                        data: function (d) {
                            d.level_kode = $('#level_kode').val(); // Filter berdasarkan level_kode
                            d._token = "{{ csrf_token() }}"; // Kirim token
                        }
                    },
                    columns: [
                        { data: "DT_RowIndex", className: "text-center", orderable: false, searchable: false },
                        { data: "level_kode", orderable: true, searchable: true },
                        { data: "level_name", orderable: true, searchable: true },
                        { data: "aksi", orderable: false, searchable: false }
                    ]
                });

                // Inisiasi data ketika filter level_kode berubah
                $('#level_kode').on('change', function () {
                    tableLevel.ajax.reload();
                });
            });
        </script>
    @endpush
</div>
```

2) Create.blade



3) Edit.blade

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

@section('content')
    <div class="card card-outline card-primary">
        <div class="card-header">
            <h3 class="card-title">Edit Level</h3>
        </div>
        <div class="card-body">
            <form action="{{ url('/level/' . $level->level_id) }}" method="POST">
                @csrf
                @method('PUT')
                <div class="form-group">
                    <label for="level_nama">Nama Level</label>
                    <input type="text" name="level_nama" id="level_nama" class="form-control" value="{{ $level->level_nama }}" required>
                </div>
                <div class="form-group">
                    <label for="level_kode">Kode Level</label>
                    <input type="text" name="level_kode" id="level_kode" class="form-control" value="{{ $level->level_kode }}" readonly>
                </div>
                <button type="submit" class="btn btn-primary">Simpan</button>
                <a href="{{ url('/level') }}" class="btn btn-secondary">Batal</a>
            </form>
        </div>
    </div>
@endsection

@push('js')
    <script>
        // Generate kode level otomatis berdasarkan nama level
        document.getElementById('level_nama').addEventListener('input', function () {
            const levelNama = this.value;
            const levelKode = levelNama.substring(0, 3).toUpperCase(); // Ambil 3 huruf pertama dan ubah ke huruf besar
            document.getElementById('level_kode').value = levelKode;
        });
    </script>
@endpush
```

4) Show.blade

```
You, yesterday | 1 author (You)
@extends('layouts.template')

@section('content')
    <div class="card card-outline card-primary">
        <div class="card-header">
            <h3 class="card-title">{{ $page->title }}</h3>
        </div>
        <div class="card-body">
            <p><strong>ID Level:</strong> {{ $level->level_id }}</p>
            <p><strong>Nama Level:</strong> {{ $level->level_nama }}</p>
            <a href="{{ url('/level') }}" class="btn btn-secondary">Kembali</a>
        </div>
    </div>
@endsection
```

d. Kemudian tambahkan properti berikut di LevelModel

```
You, yesterday | 1 author (You)
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

13 references | 0 implementations | You, yesterday | 1 author (You)
class LevelModel extends Model
{
    use HasFactory;

    protected $table = 'm_level'; // Nama tabel di database
    protected $primaryKey = 'level_id'; // Primary key tabel
    public $timestamps = false; // Nonaktifkan timestamps jika tidak digunakan

    protected $fillable = [
        'level_kode', // Kolom kode level
        'level_nama' // Kolom nama level
    ];
}
```

e. Maka tampilannya akan seperti berikut



Daftar Level

No	Kode Level	Nama Level	Aksi
1	ADM	Administrator	[Edit] [Delete]
2	MNG	Manager	[Edit] [Delete]
3	STF	Staff/Kasir	[Edit] [Delete]
4	CUS	Customer	[Edit] [Delete]
5	PEG	Pegawai	[Edit] [Delete]

1) Tambah level pengguna baru

Tambah Level

Nama Level: Pegawai

Kode Level: PEG

Simpan Batal

Daftar Level

No	Kode Level	Nama Level	Aksi
1	ADM	Administrator	[Edit] [Delete]
2	MNG	Manager	[Edit] [Delete]
3	STF	Staff/Kasir	[Edit] [Delete]
4	CUS	Customer	[Edit] [Delete]
5	PEG	Pegawai	[Edit] [Delete]

2) Menampilkan detail level pengguna

Detail Level

ID Level: 1

Nama Level: Administrator

Kembali



3) Mengedit level pengguna

The screenshot shows the 'Edit Level' page of the PWL - Starter Code application. On the left, there's a sidebar with various menu items like Dashboard, Data Pengguna, Data User, Data Barang, Kategori Barang, Data Barang, Data Supplier, and others. The 'Level User' item is currently selected. The main area has a title 'Edit Level' and two input fields: 'Nama Level' (containing 'Pegawai Baru') and 'Kode Level' (containing 'PEG'). At the bottom are two buttons: 'Simpan' (blue) and 'Batal' (grey). Below this is a larger table titled 'Daftar Level' showing a list of levels with columns for No, Kode Level, Nama Level, and Aksi (Detail, Edit, Hapus).

No	Kode Level	Nama Level	Aksi
1	ADM	Administrator	[Detail] [Edit] [Hapus]
2	MNG	Manager	[Detail] [Edit] [Hapus]
3	STF	StaffKosir	[Detail] [Edit] [Hapus]
4	CUS	Customer	[Detail] [Edit] [Hapus]
5	PEG	Pegawai Baru	[Detail] [Edit] [Hapus]

4) Filtering level pengguna

The screenshot shows the 'Daftar Level' page of the PWL - Starter Code application. A green banner at the top says 'Level berhasil diperbarui.' A dropdown filter is set to 'Administrator'. The table below shows one entry for 'ADM' with the name 'Administrator' and actions 'Detail', 'Edit', and 'Hapus'.

No	Kode Level	Nama Level	Aksi
1	ADM	Administrator	[Detail] [Edit] [Hapus]

5) Searching

The screenshot shows the 'Daftar Level' page of the PWL - Starter Code application. A search bar at the top right is filled with 'cus'. The table below shows one entry for 'CUS' with the name 'Customer' and actions 'Detail', 'Edit', and 'Hapus'. A note at the bottom says 'Showing 1 to 1 of 1 entries (filtered from 5 total entries)'.

No	Kode Level	Nama Level	Aksi
1	CUS	Customer	[Detail] [Edit] [Hapus]

6) Menghapus level pengguna

The screenshot shows the 'Daftar Level' page of the PWL - Starter Code application. A confirmation dialog box is centered over the page, asking 'Apakah Anda yakin menghapus data ini?'. It has 'OK' and 'Cancel' buttons. The background table shows one entry for 'PEG' with the name 'Pegawai Baru' and actions 'Detail', 'Edit', and 'Hapus'.

No	Kode Level	Nama Level	Aksi
1	PEG	Pegawai Baru	[Detail] [Edit] [Hapus]



2. Untuk mengimplementasikan web layout dan DataTables pada menu Tabel m_kategori, berikut adalah langkah-langkahnya:
- Tambahkan route untuk menu Tabel m_kategori di file web.php:

```
// Tambahkan route grup untuk kategori
Route::group(['prefix' => 'kategori'], function () {
    Route::get('/', [KategoriController::class, 'index']); // Menampilkan halaman tabel kategori
    Route::post('/list', [KategoriController::class, 'list']); // Mengambil data kategori untuk DataTables
    Route::get('/create', [KategoriController::class, 'create']); // Menampilkan form tambah kategori
    Route::post('/', [KategoriController::class, 'store']); // Menyimpan data kategori baru
    Route::get('{id}', [KategoriController::class, 'show']); // Menampilkan detail kategori
    Route::get('{id}/edit', [KategoriController::class, 'edit']); // Menampilkan form edit kategori
    Route::put('{id}', [KategoriController::class, 'update']); // Menyimpan perubahan kategori
    Route::delete('{id}', [KategoriController::class, 'destroy']); // Menghapus kategori
});
```

- Kemudian tambahkan method berikut di KategoriController:
 - Index(): menampilkan halaman tabel kategori
 - List(): menyediakan data kategori dalam format JSON untuk DataTables, termasuk filtering dan searching
 - Show(): menampilkan detail kategori
 - Edit(): Untuk menampilkan form edit kategori
 - Update(): Untuk menyimpan perubahan data kategori
 - Destroy(): Untuk menghapus kategori
 - Create(): Untuk menampilkan form tambah kategori
 - Store(): Untuk menyimpan data kategori baru



```
1 // Chdp
2
3 samsasca ApkMyInfoController;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Str;
7 use Illuminate\Database\Eloquent\Model;
8 use App\Kategori;
9
10 class KategoriController extends controller
11 {
12     // Inisiasi metode untuk tampilan tabel kategori
13     public function index()
14     {
15         $breadcrumb = (object) [
16             'title' => 'Daftar Kategori',
17             'list' => ['home', 'kategori']
18         ];
19
20         $page = (object) [
21             'title' => 'Daftar kategori'
22         ];
23
24         $activeMenu = 'kategori'; // Set menu yang sedang aktif
25
26         // Inisiasi view kategori dan set breadcrumb
27         $skategori = Kategori::select('Kategori.nama')->distinct()->get();
28
29         return view('kategori.index', [
30             'breadcrumb' => $breadcrumb,
31             'page' => $page,
32             'activeMenu' => $activeMenu,
33             'skategori' => $skategori
34         ]);
35     }
36
37     // Untuk menampilkan data kategori dalam format JSON untuk DataTables
38     public function datatables(Request)
39     {
40         $skategori = Kategori::select('category');
41
42         // Filter berdasarkan kategori nama
43         if ($request->has('kategori_nama')) {
44             $skategori->where('kategori_nama', 'like', '%' . $request->kategori_nama . '%');
45         }
46
47         return Datatables::of($skategori)
48             ->addColumn('action', function ($skategori) {
49                 $breadcrumb = (object) [
50                     'title' => 'Detail Kategori',
51                     'list' => ['home', 'kategori', 'tambah']
52                 ];
53
54                 $page = (object) [
55                     'title' => 'Tambah kategori baru'
56                 ];
57
58                 $activeMenu = 'kategori'; // Set menu yang sedang aktif
59
60                 return view('category.create', [
61                     'breadcrumb' => $breadcrumb,
62                     'page' => $page,
63                     'activeMenu' => $activeMenu
64                 );
65             })
66             ->addColumn('aksi') // Posisikan kolom aksi di sebelah seBELAH
67             ->make(true);
68     }
69
70     // Inisiasi metode untuk tambah kategori
71     public function create()
72     {
73         $breadcrumb = (object) [
74             'title' => 'Tambah Kategori',
75             'list' => ['home', 'kategori', 'tambah']
76         ];
77
78         $page = (object) [
79             'title' => 'Tambah kategori baru'
80         ];
81
82         $activeMenu = 'kategori'; // Set menu yang sedang aktif
83
84         return view('category.create', [
85             'breadcrumb' => $breadcrumb,
86             'page' => $page,
87             'activeMenu' => $activeMenu
88         );
89     }
90
91     // Untuk menyimpan data kategori baru
92     public function store(Request)
93     {
94         $request->validate([
95             'kategori_nama' => 'required|string|max:100', // Validasi nama kategori
96         ]);
97
98         // Generate kode kategori otomatis
99         $lastKategori = Kategori::latest('id')->first(); //ambil id terakhir kategori
100        $lastKategori = str_replace(' ', '', $lastKategori->id); // ambil angka terakhir dari id
101        $kode = 'KT-' . str_pad($lastKategori + 1, 3, '0', STR_PAD_LEFT); // Buat kode baru dengan format KT0001, KT0002, dst.
102
103        try {
104            $kategoriModel = new Kategori();
105            $kategoriModel->kode = $kode;
106            $kategoriModel->name = $request->kategori_nama; // Setelah validasi, langsung masukkan ke model
107            $kategoriModel->update();
108            return redirect('/kategori')->with('success', 'Kategori berhasil ditambahkan.');
109        } catch (\Exception $e) {
110            return redirect()->back()->withError('Terjadi kesalahan: ' . $e->getMessage());
111        }
112    }
113
114    // Inisiasi metode untuk edit kategori
115    public function edit($id)
116    {
117        $skategori = Kategori::find($id); // Ambil data kategori berdasarkan id
118
119        $breadcrumb = (object) [
120            'title' => 'Detail Kategori',
121            'list' => ['home', 'kategori', 'edit']
122        ];
123
124        $page = (object) [
125            'title' => 'Detail kategori'
126        ];
127
128        $activeMenu = 'kategori'; // Set menu yang sedang aktif
129
130        return view('category.edit', [
131             'breadcrumb' => $breadcrumb,
132             'page' => $page,
133             'activeMenu' => $activeMenu,
134             'skategori' => $skategori // Kirim data kategori ke view
135         );
136    }
137
138    // Untuk memperbarui form edit kategori
139    public function update(Request, $id)
140    {
141        $skategori = Kategori::find($id); // Ambil data kategori berdasarkan id
142
143        $breadcrumb = (object) [
144            'title' => 'Detail Kategori',
145            'list' => ['home', 'kategori', 'edit']
146        ];
147
148        $page = (object) [
149            'title' => 'Edit kategori'
150        ];
151
152        $activeMenu = 'kategori'; // Set menu yang sedang aktif
153
154        $skategori = 'kategori'; // Set menu yang sedang aktif
155
156        $request->validate([
157             'kategori_nama' => 'required|string|max:100', // Validasi nama kategori
158         ]);
159
160        $skategori = Kategori::find($id);
161        $skategori->update();
162        $skategori->name = $request->kategori_nama;
163
164        return redirect('/kategori')->with('success', 'Kategori berhasil diperbarui.');
165    }
166
167    // Untuk metode hapus
168    public function delete($id)
169    {
170        $skategori = Kategori::find($id);
171        $skategori->delete();
172
173        return redirect('/kategori')->with('success', 'Kategori berhasil dihapus.');
174    }
175}
```



c. Buat View kategori dengan beberapa file berikut:

1) Index.blade

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

@section('content')
    <div class="card card-outline card-primary">
        <div class="card-header">
            <h3 class="card-title">{{ $page->title }}</h3>
            <div class="card-tools">
                <a class="btn btn-sm btn-primary st-1" href="{{ url('kategori/create') }}>Tambah</a>
            </div>
        </div>
        <div class="card-body">
            @if (session('success'))
                <div class="alert alert-success">{{ session('success') }}</div>
            @endif
            @if (session('error'))
                <div class="alert alert-danger">{{ session('error') }}</div>
            @endif
            <div class="row">
                <div class="col-md-12">
                    <div class="form-group row">
                        <label class="col-1 control-label col-form-label">Filter:</label>
                        <div class="col-3">
                            <select class="form-control" id="kategori_nama" name="kategori_nama">
                                <option value=""> Semua </option>
                                @foreach ($kategoris as $kategori)
                                    <option value="{{ $kategori->kategori_nama }}>{{ $kategori->kategori_nama }}</option>
                                @endforeach
                            </select>
                            <small class="form-text text-muted">Nama Kategori</small>
                        </div>
                    </div>
                </div>
            </div>
            <table class="table table-bordered table-striped table-hover table-sm" id="table_kategori">
                <thead>
                    <tr>
                        <th>No/&th>
                        <th>Kode Kategori</th>
                        <th>Nama Kategori</th>
                        <th>Aksi</th>
                    </tr>
                </thead>
                <tbody>
                </tbody>
            </table>
        </div>
    @endsection

@push('css')
@push('js')
    <script>
        $(document).ready(function () {
            var tableKategori = $('#table_kategori').DataTable({
                processing: true,
                serverSide: true,
                ajax: [
                    {url: "{{ url('kategori/list') }}", type: "POST", data: function (d) {
                        d.kategori_nama = $('#kategori_nama').val(); // Filter berdasarkan kategori_nama
                        d._token = "{{ csrf_token() }}"; // Kirim token
                    }}
                ],
                columns: [
                    {data: "DT_RowIndex", className: "text-center", orderable: false, searchable: false },
                    {data: "kategori_kode", orderable: true, searchable: true },
                    {data: "kategori_nama", orderable: true, searchable: true },
                    {data: "aksi", orderable: false, searchable: false }
                ]
            });
            // Reload data ketika filter kategori nama berubah
            $('#kategori_nama').on('change', function () {
                tableKategori.ajax.reload();
            });
        });
    </script>
@endpush

```

2) Create.blade

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

@section('content')
    <div class="card card-outline card-primary">
        <div class="card-header">
            <h3 class="card-title">{{ $page->title }}</h3>
        </div>
        <div class="card-body">
            <form action="{{ url('/kategori') }}" method="POST">
                @csrf
                <div class="form-group">
                    <label for="kategori_nama">Nama Kategori</label>
                    <input type="text" name="kategori_nama" id="kategori_nama" class="form-control" placeholder="Masukkan nama kategori" required>
                </div>
                <button type="submit" class="btn btn-primary">Simpan</button>
                <a href="{{ url('/kategori') }}" class="btn btn-secondary">Batal</a>
            </form>
        </div>
    @endsection
    You, 3 hours ago * feat(kategori): implement dynamic filtering bas...
@push('js')
    <script>
        // Generate kode kategori otomatis berdasarkan nama kategori
        document.getElementById('kategori_nama').addEventListener('input', function () {
            const kategoriNama = this.value;
            const kategoriKode = 'KTG' + kategoriNama.substring(0, 3).toUpperCase(); // Format KTG + 3 huruf pertama
            document.getElementById('kategori_kode').value = kategoriKode;
        });
    </script>
@endpush

```



3) Show.blade

```
You, 3 hours ago | 1 author (You)
@extends('layouts.template')

@section('content')
    <div class="card card-outline card-primary">
        <div class="card-header">
            <h3 class="card-title">{{ $page->title }}</h3>
        </div>
        <div class="card-body">
            <p><strong>ID Kategori:</strong> {{ $kategori->kategori_id }}</p>
            <p><strong>Nama Kategori:</strong> {{ $kategori->kategori_nama }}</p>
            <a href="{{ url('/kategori') }}" class="btn btn-secondary">Kembali</a>
        </div>
    </div>
@endsection
```

4) Edit.blade

```
You, 3 hours ago | 1 author (You)
@extends('layouts.template')

@section('content')
    <div class="card card-outline card-primary">
        <div class="card-header">
            <h3 class="card-title">{{ $page->title }}</h3>
        </div>
        <div class="card-body">
            <form action="{{ url('/kategori/' . $kategori->kategori_id) }}" method="POST">
                @csrf
                @method('PUT')
                <div class="form-group">
                    <label for="kategori_nama">Nama Kategori</label>
                    <input type="text" name="kategori_nama" id="kategori_nama" class="form-control" value="{{ $kategori->kategori_nama }}" required>
                </div>
                <button type="submit" class="btn btn-primary">Simpan</button>
                <a href="{{ url('/kategori') }}" class="btn btn-secondary">Batal</a>
            </form>
        </div>
    </div>
@endsection
```

d. Kemudian tambahkan properti berikut di LevelModel

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

13 references | 0 implementations | You, yesterday | 1 author (You)
class LevelModel extends Model
{
    use HasFactory;

    0 references
    protected $table = 'm_level'; // Nama tabel di database
    0 references
    protected $primaryKey = 'level_id'; // Primary key tabel
    0 references
    public $timestamps = false; // Nonaktifkan timestamps jika tidak digunakan

    0 references
    protected $fillable = [
        'level_kode', // Kolom kode level
        'level_nama' // Kolom nama level
    ];
}
```



- e. Maka tampilannya akan seperti berikut:

No	Kode Kategori	Nama Kategori	Aksi
1	KTG001	Elektronik	[Detail] [Edit] [Hapus]
2	KTG002	Pakaian	[Detail] [Edit] [Hapus]
3	KTG003	Makanan	[Detail] [Edit] [Hapus]
4	KTG004	Minuman	[Detail] [Edit] [Hapus]
5	KTG005	Obat-obatan	[Detail] [Edit] [Hapus]
6	KTG006	Cemilan	[Detail] [Edit] [Hapus]
7	KTG007	Minuman Ringan	[Detail] [Edit] [Hapus]
8	KTG008	Sepatu	[Detail] [Edit] [Hapus]
9	KTG009	Sandal	[Detail] [Edit] [Hapus]
10	KTG010	Baju	[Detail] [Edit] [Hapus]

1) Tambah kategori barang baru

Tambah Kategori

Tambah kategori baru

Nama Kategori

Sandal

Simpan Batal

No	Kode Kategori	Nama Kategori	Aksi
11	KTG011	Sandal	[Detail] [Edit] [Hapus]

2) Menampilkan detail kategori pengguna baru

Detail Kategori

Detail kategori

ID Kategori: 13

Nama Kategori: Sandal

Kembali



3) Mengedit kategori pengguna

Edit Kategori

Nama Kategori

Celana

Simpan **Batal**

Daftar Kategori

Kategori berhasil diperbarui.

No	Kode Kategori	Nama Kategori	Aksi
11	KTG011	Celana	Detail Edit Hapus

4) Filtering kategori pengguna

Daftar Kategori

Kategori berhasil diperbarui.

Filter: Pakaian

No	Kode Kategori	Nama Kategori	Aksi
1	KTG002	Pakaian	Detail Edit Hapus

5) Searching

Daftar Kategori

Kategori berhasil diperbarui.

Search: ce

No	Kode Kategori	Nama Kategori	Aksi
1	KTG006	Cemilan	Detail Edit Hapus
2	KTG011	Celana	Detail Edit Hapus



6) Menghapus kategori pengguna

The screenshot shows a web application interface for managing categories. A modal dialog box is open, asking "Apakah Anda yakin menghapus data ini?" (Are you sure you want to delete this data?) with "OK" and "Cancel" buttons. Below the modal, a green success message says "Kategori berhasil diperbarui." (Category has been updated successfully). The main table shows two categories: KTG006 (Cemilan) and KTG011 (Celana). On the left sidebar, under the "Data Barang" section, the "Kategori Barang" option is selected. The table has columns: No, Kode Kategori, Nama Kategori, and Aksi (Actions). The actions column contains "Detail", "Edit", and "Hapus" buttons. The bottom of the page shows a footer with "Showing 1 to 10 of 10 entries".

3. Untuk mengimplementasikan web layout dan DataTables pada menu Tabel m_barang, berikut adalah langkah-langkahnya:

- a. Tambahkan route untuk menu Tabel m_barang di file web.php

```
// Tambahkan route grup untuk barang
Route::group(['prefix' => 'barang'], function () {
    Route::get('/', [BarangController::class, 'index']); // Menampilkan halaman tabel barang
    Route::post('/list', [BarangController::class, 'list']); // Mengambil data barang untuk DataTables
    Route::get('/create', [BarangController::class, 'create']); // Menampilkan form tambah barang
    Route::post('/', [BarangController::class, 'store']); // Menyimpan data barang baru
    Route::get('{id}', [BarangController::class, 'show']); // Menampilkan detail barang
    Route::get('{id}/edit', [BarangController::class, 'edit']); // Menampilkan form edit barang
    Route::put('{id}', [BarangController::class, 'update']); // Menyimpan perubahan barang
    Route::delete('{id}', [BarangController::class, 'destroy']); // Menghapus barang
});
```

- b. Membuat controller BarangController, dan mengisinya dengan beberapa function berikut:

```
PS D:\Apps\laragon\www\PWL2025\week 5.2\jobsheet> php artisan make:controller BarangController
INFO Controller [D:\Apps\laragon\www\PWL2025\week 5.2\jobsheet\app\Http\Controllers\BarangController.php] created successfully.
```

- 1) Index(): menampilkan halaman tabel barang
- 2) List(): menyediakan data kategori dalam format JSON untuk DataTables, termasuk filtering dan searching



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141
Telp. (0341) 404424 – 404425, Fax (0341) 404420
<http://www.polinema.ac.id>

- 3) Show(): menampilkan detail barang
- 4) Edit(): Untuk menampilkan form edit barang
- 5) Update(): Untuk menyimpan perubahan data barang
- 6) Destroy: Untuk menghapus barang
- 7) Create(): Untuk menampilkan form tambah barang
- 8) Store(): Untuk menyimpan data barang baru



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141
Telp. (0341) 404424 – 404425, Fax (0341) 404420
<http://www.polinema.ac.id>



c. Buat View barang dengan beberapa file berikut:

1) Index.blade

```
<extends>'layouts.template'

<section>'content'
    <div class="card card-outline card-primary">
        <div class="card-title"><span>Title </span></div>
        <div class="card-body">
            <a href="#"><button type="button" class="btn btn-sm btn-primary w-100" href="{{ url('barang/create') }}"></button></a>
        </div>
        <div class="card-body">
            <div>{{ session('success') }}<br>
                <div class="alert alert-success">{{ session('success') }}{{ session('error') }}>- Semua Kategori -</option>
                            @foreach ($kategoris as $kategori)
                                <option value="{{ $kategori->kategori_id }}>{{ $kategori->kategori_nama }}</option>
                            @endforeach
                        </select>
                        <small class="form-text text-muted">Kategori barang</small>
                    </div>
                </div>
            </div>
        </div>
        <table border="1" style="width:100%; border-collapse: collapse;" id="table_barang">
            <thead>
                <tr>
                    <th>Barcode</th>
                    <th>Nama barang</th>
                    <th>Harga barang</th>
                    <th>Stok barang</th>
                    <th>Barang Baku</th>
                    <th>Barcode</th>
                </tr>
            </thead>
            <tbody>
                @foreach ($barangs as $barang)
                    <tr>
                        <td>{{ $barang->barcode }}</td>
                        <td>{{ $barang->nama }}</td>
                        <td>{{ $barang->harga }}</td>
                        <td>{{ $barang->stok }}</td>
                        <td>{{ $barang->barang_baku }}</td>
                        <td>{{ $barang->barcode }}</td>
                    </tr>
                @endforeach
            </tbody>
        </table>
    </div>
</section>
<script>
    $(document).ready(function () {
        var tablebarang = $('#table_barang').DataTable({
            processing: true,
            serverSide: true,
            ajax: {
                url: "{{ url('barang/list') }}",
                type: "POST",
                data: function (d) {
                    d.kategori_id = $('#barang_id').val(); // Filter kategori barang
                    d.barang_baku = $('#barang_baku').val(); // Filter barang baku
                    d._token = '{{ csrf_token() }}'; // Token anti-CSRF
                }
            },
            columns: [
                { data: "barcode", className: "text-align" , searchable: false, sortable: false },
                { data: "barang_id", searchable: true, searchable: true },
                { data: "barang_nama", searchable: true, searchable: true },
                { data: "barang_harga", searchable: true, searchable: true },
                { data: "barang_stok", searchable: true, searchable: true },
                { data: "barang_baku_id", searchable: true, searchable: false },
                { data: "barcode", searchable: false, searchable: false }
            ]
        });

        $('#barang_id').on('change', function () {
            tablebarang.ajax.reload();
        });

        let typingTimer;
        $('#barang_nama').on('keyup', function () {
            clearTimeout(typingTimer);
            typingTimer = setTimeout(function () {
                tablebarang.ajax.reload();
            }, 500);
        });

        $('#barang_harga').on('change', function () {
            clearTimeout(typingTimer);
            typingTimer = setTimeout(function () {
                tablebarang.ajax.reload();
            }, 500);
        });
    });
</script>

```

2) Create.blade



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI MALANG

JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141

Telp. (0341) 404424 – 404425, Fax (0341) 404420

<http://www.polinema.ac.id>

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

@section('content')
    <div class="card card-outline card-primary">
        <div class="card-header">
            <h3 class="card-title"><{ $page>title }></h3>
        </div>
        <div class="card-body">
            <form action="[{ url('/barang') }]" method="POST">
                @csrf
                <div class="form-group">
                    <label for="barang_nama">Nama Barang</label>
                    <input type="text" name="barang_nama" id="barang_nama" class="form-control" placeholder="Masukan nama barang" value="[{ old('barang_nama') }]" required>
                    @error('barang_nama')
                        <small class="text-danger">[{ $message }]</small>
                    @enderror
                </div>
                <div class="form-group">
                    <label for="kategori_id">Kategori</label>
                    <select name="kategori_id" id="kategori_id" class="form-control" required>
                        <option value="">-- Pilih Kategori --</option>
                        @foreach ($kategoris as $skategori)
                            <option value="[{ $kategori->kategori_id }]">{ ($old('kategori_id') == $skategori->kategori_id) ? 'selected' : '' } { $skategori->kategori_nama }</option>
                        @endforeach
                    </select>
                    @error('kategori_id')
                        <small class="text-danger">[{ $message }]</small>
                    @enderror
                </div>
                <div class="row">
                    <div class="col-md-6">
                        <div class="form-group">
                            <label for="harga_beli">Harga Beli</label>
                            <div class="input-group">
                                <div class="input-group-prepend">
                                    <span class="input-group-text">$Rp /<span>
                                </div>
                                <input type="number" name="harga_beli" id="harga_beli" class="form-control" placeholder="Masukan harga beli" value="[{ old('harga_beli') }]" required>
                                <div>
                                    @error('harga_beli')
                                        <small class="text-danger">[{ $message }]</small>
                                    @enderror
                                </div>
                            </div>
                        </div>
                    </div>
                    <div class="col-md-6">
                        <div class="form-group">
                            <label for="harga_jual">Harga Jual</label>
                            <div class="input-group">
                                <div class="input-group-prepend">
                                    <span class="input-group-text">$Rp /<span>
                                </div>
                                <input type="number" name="harga_jual" id="harga_jual" class="form-control" placeholder="Masukan harga jual" value="[{ old('harga_jual') }]" required>
                                <div>
                                    @error('harga_jual')
                                        <small class="text-danger">[{ $message }]</small>
                                    @enderror
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
                <button type="submit" class="btn btn-primary">Simpan</button>
                <a href="[{ url('/barang') }]" class="btn btn-secondary">Batal</a>
            </form>
        </div>
    </div>
    @endsection

@push('js')
    <script>
        validated_harga_jual_kurang_lebih_besar_dari_harga_beli();
        document.getElementById('harga_jual').addEventListener('change', function() {
            const hargabeli = parseFloat(document.getElementById('harga_beli').value) || 0;
            const hargajual = parseFloat(this.value) || 0;

            if (hargajual <= hargabeli) {
                alert('Harga jual sebaiknya lebih besar dari harga beli');
            }
        });
    </script>
    @endpush

```

3) Show.blade

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

@section('content')
    <div class="card card-outline card-primary">
        <div class="card-header">
            <h3 class="card-title"><{ $page>title }></h3>
        </div>
        <div class="card-body">
            <div class="table-responsive">
                <table class="table table-bordered">
                    <thead>
                        <tr>
                            <th style="width: 80px;">Kode Barang</th>
                            <th>[{ barang->barang_kode }]</th>
                        </tr>
                    </thead>
                    <tbody>
                        <tr>
                            <th>Nama Barang</th>
                            <td>{ ($barang->barang_nama)}</td>
                        </tr>
                        <tr>
                            <th>Kategori</th>
                            <td>{ ($barang->kategori->kategori_nama )}</td>
                        </tr>
                        <tr>
                            <th>Harga Beli</th>
                            <td>Rp {{ number_format(num: $barang->harga_beli, decimals: 0, decimal_separator: ',', thousands:'.') }}</td>
                        </tr>
                        <tr>
                            <th>Harga Jual</th>
                            <td>Rp {{ number_format(num: $barang->harga_jual, decimals: 0, decimal_separator: ',', thousands:'.') }}</td>
                        </tr>
                        <tr>
                            <th>Margin</th>
                            <td>Rp {{ number_format(num: $barang->harga_jual - $barang->harga_beli, decimals: 0, decimal_separator: ',', thousands:'.') }}  

                                {{{ round(num: ($barang->harga_jual - $barang->harga_beli) / $barang->harga_beli * 100, precision: 2) }}%</td>
                        </tr>
                        <tr>
                            <th>Tanggal Dibuat</th>
                            <td>{{ ($barang->created_at ? $barang->created_at->Format('d-m-Y H:i:s') : '-' )}}</td>
                        </tr>
                        <tr>
                            <th>Terakhir Diupdate</th>
                            <td>{{ ($barang->updated_at ? $barang->updated_at->Format('d-m-Y H:i:s') : '-' )}}</td>
                        </tr>
                    </tbody>
                </table>
            </div>
            <div class="mt-3">
                <a href="[{ url('/barang') }]" class="btn btn-secondary">Batal</a>
                <a href="[{ url('/barang', $barang->barang_id, 'edit') }]" class="btn btn-warning">Edit</a>
                <form class="d-inline-block" method="POST" action="[{ url('/barang', $barang->barang_id, 'delete') }]">
                    @csrf
                    @method('DELETE')
                    <button type="submit" class="btn btn-danger" onclick="return confirm('Apakah Anda yakin menghapus data ini?')>Hapus</button>
                </form>
            </div>
        </div>
    </div>
    @endsection
```



4) Edit.blade

```
<button type="button" class="btn btn-primary">Batal
<button type="button" class="btn btn-secondary">Simpan
</div>
</div>

```

- d. Buat BarangModel dan isi propertinya seperti berikut

```
PS D:\Apps\laragon\www\PWL2025\week 5.2\jobsheet> php artisan make:model BarangModel

[INFO] Model [D:\Apps\laragon\www\PWL2025\week 5.2\jobsheet\app\Models\BarangModel.php] created successfully.

You, 2 hours ago | 1 author (You)
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;

0 references | 0 implements
You, 2 hours ago | 1 author (You)
class BarangModel extends Model
{
    use HasFactory;
    You, 2 hours ago * Fatt(supplier): implement CRUD operations for ...
    0 references
    protected $table = 'm_barang';
    0 references
    protected $primaryKey = 'barang_id';

    0 references
    protected $fillable = [
        'barang_kode',
        'barang_nama',
        'harga_beli',
        'harga_jual',
        'kategori_id'
    ];

    /**
     * Get kategori barang
     */
    0 references | 0 implements
    public function kategori(): BelongsTo
    {
        return $this->belongsTo(KategoriModel::class, 'kategori_id', 'kategori_id');
    }
}
```



e. Maka tampilannya akan seperti berikut:

No+	Kode Barang	Nama Barang	Kategori	Harga Beli	Harga Jual	Aksi
1	BRG001	Laptop ASUS ROG	Elektronik	Rp 15.000.000	Rp 17.500.000	[Detail] [Edit] [Hapus]
2	BRG002	MacBook Air M2	Elektronik	Rp 18.000.000	Rp 20.500.000	[Detail] [Edit] [Hapus]
3	BRG003	Monitor 27 inch IPS	Elektronik	Rp 3.500.000	Rp 4.200.000	[Detail] [Edit] [Hapus]
4	BRG004	Kaos Polos Katun	Pakaian	Rp 50.000	Rp 75.000	[Detail] [Edit] [Hapus]
5	BRG005	Jaket Denim	Pakaian	Rp 200.000	Rp 250.000	[Detail] [Edit] [Hapus]
6	BRG006	Sepatu Sneakers	Pakaian	Rp 400.000	Rp 550.000	[Detail] [Edit] [Hapus]
7	BRG007	Roti Tawar	Makanan	Rp 15.000	Rp 20.000	[Detail] [Edit] [Hapus]
8	BRG008	Biskuit Gandum	Makanan	Rp 25.000	Rp 35.000	[Detail] [Edit] [Hapus]
9	BRG009	Mie Instan	Makanan	Rp 3.500	Rp 5.000	[Detail] [Edit] [Hapus]
10	BRG010	Air Mineral 1L	Minuman	Rp 5.000	Rp 8.000	[Detail] [Edit] [Hapus]

1) Tambah barang baru

Barang berhasil ditambahkan.

No+	Kode Barang	Nama Barang	Kategori	Harga Beli	Harga Jual	Aksi
11	BRG011	Jus Jeruk Botol	Minuman	Rp 11.000	Rp 18.000	[Detail] [Edit] [Hapus]
12	BRG012	Kopi Bubuk 250g	Minuman	Rp 30.000	Rp 40.000	[Detail] [Edit] [Hapus]
13	BRG013	Paracetamol 500mg	Obat-obatan	Rp 2.500	Rp 5.000	[Detail] [Edit] [Hapus]
14	BRG014	Vitamin C 1000mg	Obat-obatan	Rp 10.000	Rp 15.000	[Detail] [Edit] [Hapus]
15	BRG015	Antiseptik Cair 100ml	Obat-obatan	Rp 18.000	Rp 25.000	[Detail] [Edit] [Hapus]
16	BRG016	Masako	Minuman	Rp 1.000	Rp 2.000	[Detail] [Edit] [Hapus]
17	BRG017	Masako	Makanan	Rp 1.000	Rp 2.000	[Detail] [Edit] [Hapus]



2) Menampilkan detail barang baru

Detail barang

Kode Barang	BRG011
Nama Barang	Jus Jeruk Botol
Kategori	Minuman
Harga Beli	Rp 11.000
Harga Jual	Rp 18.000
Margin	Rp 7.000 (63.64%)
Tanggal Dibuat	-
Terakhir Diupdate	25-03-2025 12:49:10

Kembali Edit Hapus

3) Mengedit barang pengguna

Daftar Barang

Daftar barang

Barang berhasil diperbarui.

No	Kode Barang	Nama Barang	Kategori	Harga Beli	Harga Jual	Aksi
11	BRG011	Jus Jeruk Botol	Minuman	Rp 11.000	Rp 18.000	[Detail] [Edit] [Hapus]
12	BRG012	Kopi Bubuk 250g	Minuman	Rp 30.000	Rp 40.000	[Detail] [Edit] [Hapus]
13	BRG013	Paracetamol 500mg	Obat-obatan	Rp 2.500	Rp 5.000	[Detail] [Edit] [Hapus]
14	BRG014	Vitamin C 1000mg	Obat-obatan	Rp 10.000	Rp 15.000	[Detail] [Edit] [Hapus]
15	BRG015	Antiseptik Cair 100ml	Obat-obatan	Rp 18.000	Rp 25.000	[Detail] [Edit] [Hapus]
16	BRG016	Masako	Minuman	Rp 1.000	Rp 2.000	[Detail] [Edit] [Hapus]
17	BRG017	Masako Sapi	Makanan	Rp 1.000	Rp 2.000	[Detail] [Edit] [Hapus]

4) Filtering barang pengguna

Daftar Barang

Daftar barang

Barang berhasil diperbarui.

No	Kode Barang	Nama Barang	Kategori	Harga Beli	Harga Jual	Aksi
1	BRG001	Laptop ASUS ROG	Elektronik	Rp 15.000.000	Rp 17.500.000	[Detail] [Edit] [Hapus]
2	BRG002	MacBook Air M2	Elektronik	Rp 18.000.000	Rp 20.500.000	[Detail] [Edit] [Hapus]
3	BRG003	Monitor 27 inch IPS	Elektronik	Rp 3.500.000	Rp 4.200.000	[Detail] [Edit] [Hapus]



5) Searching

Barang berhasil diperbarui.						
Filter: - Semua Kategori - Kategori Barang						
Show 10 entries Search: m						
No	Kode Barang	Nama Barang	Kategori	Harga Beli	Harga Jual	Aksi
1	BRG002	MacBook Air M2	Elektronik	Rp 18.000.000	Rp 20.500.000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
2	BRG016	Masako	Minuman	Rp 1.000	Rp 2.000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
3	BRG017	Masako Sapi	Makanan	Rp 1.000	Rp 2.000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>

Showing 1 to 3 of 3 entries (Filtered from 17 total entries)

Previous **1** Next

6) Menghapus barang pengguna



127.0.0.1:8000 says						
Apakah Anda yakin menghapus data ini?						
OK Cancel						
Daftar Barang						
Daftar barang						
Barang berhasil diperbarui.						
Filter: - Semua Kategori - Kategori Barang						
Show 10 entries Search:						
No	Kode Barang	Nama Barang	Kategori	Harga Beli	Harga Jual	Aksi
11	BRG011	Jus Jeruk Botol	Minuman	Rp 11.000	Rp 18.000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
12	BRG012	Kopi Bubuk 250g	Minuman	Rp 30.000	Rp 40.000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
13	BRG013	Paracetamol 500mg	Obat-obatan	Rp 2.500	Rp 5.000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
14	BRG014	Vitamin C 1000mg	Obat-obatan	Rp 10.000	Rp 15.000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
15	BRG015	Antiseptik Cair 100ml	Obat-obatan	Rp 18.000	Rp 25.000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
16	BRG016	Masako	Minuman	Rp 1.000	Rp 2.000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
17	BRG017	Masako Sapi	Makanan	Rp 1.000	Rp 2.000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>

Showing 11 to 16 of 16 entries

Previous **1** Next

4. Untuk implementasikan web layout dan DataTables pada menu Tabel m_supplier, berikut adalah langkah-langkahnya:

- a. Membuat migration untuk m_supplier

PS D:\Apps\laragon\www\PWL2025\week 5.2\jobsheet> `php artisan make:migration create_m_supplier_table`

```
INFO Migration [D:\Apps\laragon\www\PWL2025\week 5.2\jobsheet\database\migrations\2025_03_25_130309_create_m_supplier_table.php] created successfully.
```



```
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('m_supplier', function (Blueprint $table): void {
            $table->id('supplier_id');
            $table->string('supplier_kode', 10)->unique();
            $table->string('supplier_nama', 100);
            $table->string('supplier_alamat', 255)->nullable();
            $table->string('supplier_email', 100)->nullable();
            $table->string('supplier_kontak', 100)->nullable();
            $table->timestamp();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('m_supplier');
    }
};
```

● PS D:\Apps\laragon\www\PWL2025\week 5.2\jobsheet> php artisan migrate

INFO Running migrations.

2025_03_25_130309_create_m_supplier_table 156ms **DONE**

b. Mengisi seeder m_supplier

● PS D:\Apps\laragon\www\PWL2025\week 5.2\jobsheet> php artisan make:seeder SupplierSeeder

INFO Seeder [D:\Apps\laragon\www\PWL2025\week 5.2\jobsheet\database\seeders\SupplierSeeder.php] created successfully.



```
base > seeders > SupplierSeeder.php > PHP Inteference > SupplierSeeder  
<?php  
  
namespace Database\Seeders;  
  
use Illuminate\Database\Console\Seeds\WithoutModelEvents;  
use Illuminate\Seeder;  
use Illuminate\Support\Facades\DB;  
  
class SupplierSeeder extends Seeder  
{  
    /**  
     * Run the database seeds.  
     */  
    public function run(): void  
    {  
        DB::table('m_supplier')->insert([  
            [  
                'supplier_id' => 1,  
                'supplier_kode' => 'SUP001',  
                'supplier_nama' => 'PT. Elektronik Jaya',  
                'supplier_alamat' => 'Jl. Industri No. 123, Jakarta',  
                'supplier_telp' => '021-5551234',  
                'supplier_email' => 'info@elektronikjaya.com',  
                'supplier_kontak' => 'Budi Santoso',  
                'created_at' => now(),  
                'updated_at' => now(),  
            ],  
            [  
                'supplier_id' => 2,  
                'supplier_kode' => 'SUP002',  
                'supplier_nama' => 'CV. Fashion Indonesia',  
                'supplier_alamat' => 'Jl. Tekstil No. 45, Bandung',  
                'supplier_telp' => '022-6667890',  
                'supplier_email' => 'sales@fashionindonesia.co.id',  
                'supplier_kontak' => 'Dewi Lestari',  
                'created_at' => now(),  
                'updated_at' => now(),  
            ],  
            [  
                'supplier_id' => 3,  
                'supplier_kode' => 'SUP003',  
                'supplier_nama' => 'UD. Makanan Sehat',  
                'supplier_alamat' => 'Jl. Pasar Baru No. 78, Surabaya',  
                'supplier_telp' => '031-7772345',  
                'supplier_email' => 'order@makanansehat.com',  
                'supplier_kontak' => 'Agus Wijaya',  
                'created_at' => now(),  
                'updated_at' => now(),  
            ],  
            [  
                'supplier_id' => 4,  
                'supplier_kode' => 'SUP004',  
                'supplier_nama' => 'PT. Minuman Prima',  
                'supplier_alamat' => 'Jl. Industri Minuman Blok A5, Tangerang',  
                'supplier_telp' => '021-8883456',  
                'supplier_email' => 'cs@minumanprima.com',  
                'supplier_kontak' => 'Siti Rahayu',  
                'created_at' => now(),  
                'updated_at' => now(),  
            ],  
            [  
                'supplier_id' => 5,  
                'supplier_kode' => 'SUP005',  
                'supplier_nama' => 'PT. Farma Indonesia',  
                'supplier_alamat' => 'Jl. Farmasi No. 56, Semarang',  
                'supplier_telp' => '024-9994567',  
                'supplier_email' => 'info@farmaindonesia.com',  
                'supplier_kontak' => 'Dr. Hendra Wijaya',  
                'created_at' => now(),  
                'updated_at' => now(),  
            ],  
        ]);  
    }  
}
```

● PS D:\Apps\laragon\www\PWL2025\week 5.2\jobsheet> php artisan db:seed --class=SupplierSeeder

INFO Seeding database.

	supplier_id	supplier_kode	supplier_nama	supplier_alamat	supplier_telp	supplier_email	supplier_kontak	created_at	updated_at
□	Edit	Copy	Delete	1 SUP001	PT. Elektronik Jaya	Jl. Industri No. 123, Jakarta	021-5551234	info@elektronikjaya.com	Budi Santoso 2025-03-25 13:08:58 2025-03-25 13:08:58
□	Edit	Copy	Delete	2 SUP002	CV. Fashion Indonesia	Jl. Tekstil No. 45, Bandung	022-6667890	sales@fashionindonesia.co.id	Dewi Lestari 2025-03-25 13:08:58 2025-03-25 13:08:58
□	Edit	Copy	Delete	3 SUP003	UD. Makanan Sehat	Jl. Pasar Baru No. 78, Surabaya	031-7772345	order@makanansehat.com	Agus Wijaya 2025-03-25 13:08:58 2025-03-25 13:08:58
□	Edit	Copy	Delete	4 SUP004	PT. Minuman Prima	Jl. Industri Minuman Blok A5, Tangerang	021-8883456	cs@minumanprima.com	Siti Rahayu 2025-03-25 13:08:58 2025-03-25 13:08:58
□	Edit	Copy	Delete	5 SUP005	PT. Farma Indonesia	Jl. Farmasi No. 56, Semarang	024-9994567	info@farmaindonesia.com	Dr. Hendra Wijaya 2025-03-25 13:08:58 2025-03-25 13:08:58



- c. Tambahkan route untuk menu Tabel m_supplier di file web.php

```
// Tambahkan route grup untuk supplier
Route::group(['prefix' => 'supplier'], function () {
    Route::get('/', [SupplierController::class, 'index']); // Menampilkan halaman tabel supplier
    Route::post('/list', [SupplierController::class, 'list']); // Mengambil data supplier untuk DataTables
    Route::get('/create', [SupplierController::class, 'create']); // Menampilkan form tambah supplier
    Route::post('/', [SupplierController::class, 'store']); // Menyimpan data supplier baru
    Route::get('/{id}', [SupplierController::class, 'show']); // Menampilkan detail supplier
    Route::get('/{id}/edit', [SupplierController::class, 'edit']); // Menampilkan form edit supplier
    Route::put('/{id}', [SupplierController::class, 'update']); // Menyimpan perubahan supplier
    Route::delete('/{id}', [SupplierController::class, 'destroy']); // Menghapus supplier
});
```

- d. Membuat controller SupplierController, dan mengisinya dengan beberapa function berikut:

```
● PS D:\Apps\laragon\www\PWL2025\week 5.2\jobsheet> php artisan make:controller SupplierController --resource
INFO Controller [D:\Apps\laragon\www\PWL2025\week 5.2\jobsheet\app\Http\Controllers\SupplierController.php] created successfully.
```

- 1) Index(): menampilkan halaman tabel supplier
- 2) List(): menyediakan data supplier dalam format JSON untuk DataTables, termasuk filtering dan searching
- 3) Show(): menampilkan detail supplier
- 4) Edit(): Untuk menampilkan form edit supplier
- 5) Update(): Untuk menyimpan perubahan data supplier
- 6) Destroy: Untuk menghapus supplier
- 7) Create(): Untuk menampilkan form tambah supplier
- 8) Store(): Untuk menyimpan data supplier baru



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141
Telp. (0341) 404424 – 404425, Fax (0341) 404420
<http://www.polinema.ac.id>



e. Buat View supplier dengan beberapa file berikut:

1) Index.blade

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

@section('content')
    <div class="card card-outline card-primary">
        <div class="card-header">
            <h3 class="card-title">({ $page->title })</h3>
            <div class="card-tools">
                <a class="btn btn-sm btn-primary et-i" href="{{ url('supplier/create') }}>Tambah</a>
            </div>
        </div>
        <div class="card-body">
            @if (session('success'))
                <div class="alert alert-success">{( session('success') )}</div>
            @endif

            @if (session('error'))
                <div class="alert alert-danger">{( session('error') )}</div>
            @endif
            <div class="row">
                <div class="col-md-12">
                    <div class="form-group row">
                        <label class="control-label col-form-label">Filter:</label>
                        <div class="col">
                            <select class="form-control" id="supplier_kode" name="supplier_kode">
                                <option value=> Semua </option>
                                <option value="SUP001">PT. Elektronik Jaya</option>
                                <option value="SUP002">PT. Makana Sabtu</option>
                                <option value="SUP003">AB. Makana Sabtu</option>
                                <option value="SUP004">PT. Minuman Prima</option>
                                <option value="SUP005">PT. Fama Indonesia</option>
                            </select>
                            <small class="form-text text-muted">Supplier</small>
                        </div>
                    </div>
                </div>
            </div>
            <table class="table table-bordered table-striped table-hover table-sm" id="table_supplier">
                <thead>
                    <tr>
                        <th><input type="checkbox" /></th>
                        <th>Nama Supplier</th>
                        <th>Nama Person</th>
                        <th>Alamat</th>
                    </tr>
                </thead>
                <tbody>
                    <tr>
                        <td><input type="checkbox" /></td>
                        <td>Bina Media</td>
                        <td>Dwi Nurwulan</td>
                        <td>Jl. Raya Ciputat Km 12</td>
                    </tr>
                    <tr>
                        <td><input type="checkbox" /></td>
                        <td>Bina Media</td>
                        <td>Dwi Nurwulan</td>
                        <td>Jl. Raya Ciputat Km 12</td>
                    </tr>
                </tbody>
            </table>
        </div>
    @endsection

    @push('css')
        <script>
            $(document).ready(function () {
                var tableSupplier = $('#table_supplier').DataTable({
                    processing: true,
                    serverSide: true,
                    ajax: {
                        url: "{{ url('supplier/list') }}",
                        type: "POST",
                        data: function (d) {
                            d.supplier_kode = $('#supplier_kode').val(); // Filter berdasarkan supplier_kode
                            d.token = {{ csrf_token() }}; // Add token
                        }
                    },
                    columns: [
                        { data: "UI_RowIndex", className: "text-center", orderable: false, searchable: false },
                        { data: "supplier_kode", orderable: true, searchable: true },
                        { data: "supplier_nama", orderable: true, searchable: true },
                        { data: "supplier_telp", orderable: true, searchable: true },
                        { data: "supplier_email", orderable: true, searchable: true },
                        { data: "supplier_kontak", orderable: true, searchable: true },
                        { data: "aksi", orderable: false, searchable: false }
                    ]
                });
                // Nama data ketika filter supplier_kode berubah
                $('#supplier_kode').on('change', function () {
                    tableSupplier.ajax.reload();
                });
            });
        </script>
    @endpush

```

2) Create.blade

```
@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">
                <form action="{{ url('supplier') }}" method="POST">
                    <button type="button" class="btn btn-primary">Simpan</button>
                </form>
            </div>
        </div>
        <div class="card-body">
            <form action="{{ url('supplier') }}" method="POST">
                <div class="form-group">
                    <label for="supplier_nama">Nama Supplier</label>
                    <input type="text" name="supplier_nama" id="supplier_nama" class="form-control" placeholder="Masukkan nama supplier" value="{{ old('supplier_nama') }}" required>
                    @error('supplier_nama')
                        <small class="text-danger">{{ $message }}</small>
                    @enderror
                </div>
                <div class="form-group">
                    <label for="supplier_alamat">Alamat</label>
                    <textarea name="supplier_alamat" id="supplier_alamat" class="form-control" rows="3" placeholder="Masukkan alamat supplier" value="{{ old('supplier_alamat') }}"></textarea>
                    @error('supplier_alamat')
                        <small class="text-danger">{{ $message }}</small>
                    @enderror
                </div>
                <div class="form-group">
                    <label for="supplier_telp">Telepon</label>
                    <input type="text" name="supplier_telp" id="supplier_telp" class="form-control" placeholder="Masukkan nomor telepon" value="{{ old('supplier_telp') }}"><br>
                    @error('supplier_telp')
                        <small class="text-danger">{{ $message }}</small>
                    @enderror
                </div>
                <div class="form-group">
                    <label for="supplier_email">Email</label>
                    <input type="email" name="supplier_email" id="supplier_email" class="form-control" placeholder="Masukkan email supplier" value="{{ old('supplier_email') }}"><br>
                    @error('supplier_email')
                        <small class="text-danger">{{ $message }}</small>
                    @enderror
                </div>
                <div class="form-group">
                    <label for="supplier_kontak">Kontak Person</label>
                    <input type="text" name="supplier_kontak" id="supplier_kontak" class="form-control" placeholder="Masukkan nomor kontak person" value="{{ old('supplier_kontak') }}"><br>
                    @error('supplier_kontak')
                        <small class="text-danger">{{ $message }}</small>
                    @enderror
                </div>
                <div class="form-group">
                    <button type="submit" class="btn btn-primary">Simpan</button>
                    <a href="{{ url('supplier') }}" class="btn btn-secondary">Batal</a>
                </div>
            </form>
        </div>
    @endsection

```



3) Show.blade

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

@section('content')
    <div class="card card-outline card-primary">
        <div class="card-header">
            <h3 class="card-title">{{ $page->title }}</h3>
        </div>
        <div class="card-body">
            <div class="table-responsive">
                <table class="table table-bordered">
                    <tr>
                        <th style="width: 20px;>Kode Supplier</th>
                        <td>{{ $supplier->supplier_kode }}</td>
                    </tr>
                    <tr>
                        <th>Nama Supplier</th>
                        <td>{{ $supplier->supplier_nama }}</td>
                    </tr>
                    <tr>
                        <th>Alamat</th>
                        <td>{{ $supplier->supplier_alamat ?? '-' }}</td>
                    </tr>
                    <tr>
                        <th>Telp</th>
                        <td>{{ $supplier->supplier_telp ?? '-' }}</td>
                    </tr>
                    <tr>
                        <th>Email</th>
                        <td>{{ $supplier->supplier_email ?? '-' }}</td>
                    </tr>
                    <tr>
                        <th>Kontak Person</th>
                        <td>{{ $supplier->supplier_kontak ?? '-' }}</td>
                    </tr>
                    <tr>
                        <th>Tanggal Dibuat</th>
                        <td>{{ $supplier->created_at ? $supplier->created_at->format('d-m-Y H:i:s') : '-' }}</td>
                    </tr>
                    <tr>
                        <th>Tanggal Diupdate</th>
                        <td>{{ $supplier->updated_at ? $supplier->updated_at->format('d-m-Y H:i:s') : '-' }}</td>
                    </tr>
                </table>
            </div>
            <div class="mt-3">
                <a href="{{ url('/supplier') }}" class="btn btn-secondary">Kembali</a>
                <a href="{{ url('/supplier/' . $supplier->supplier_id . '/edit') }}" class="btn btn-warning">Edit</a>
                <form class="d-inline-block" method="POST" action="{{ url('/supplier/' . $supplier->supplier_id) }}">
                    @csrf
                    @method('DELETE')
                    <button type="submit" class="btn btn-danger" onclick="return confirm('Apakah Anda yakin menghapus data ini?');">Hapus</button>
                </form>
            </div>
        </div>
    </div>
@endsection
```

4) Edit.blade

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

@section('content')
    <div class="card card-outline card-primary">
        <div class="card-header">
            <h3 class="card-title">{{ $page->title }}</h3>
        </div>
        <div class="card-body">
            <form action="{{ url('/supplier/' . $supplier->supplier_id) }}" method="POST">
                @csrf
                @method('PUT')
                <div class="form-group">
                    <label for="supplier.kode">Kode Supplier</label>
                    <input type="text" name="supplier.kode" id="supplier.kode" class="form-control" value="{{ $supplier->supplier_kode }}" readonly>
                </div>
                <div class="form-group">
                    <label for="supplier.nama">Nama Supplier</label>
                    <input type="text" name="supplier.nama" id="supplier.nama" class="form-control" placeholder="Masukkan nama supplier" value="{{ old('supplier.nama', $supplier->supplier_nama) }}" required>
                    @error('supplier.nama')
                        <small class="text-danger">{{ $message }}</small>
                    @enderror
                </div>
                <div class="form-group">
                    <label for="supplier.alamat">Alamat</label>
                    <textarea name="supplier.alamat" id="supplier.alamat" class="form-control" rows="3" placeholder="Masukkan alamat supplier" value="{{ old('supplier.alamat', $supplier->supplier_alamat) }}></textarea>
                    @error('supplier.alamat')
                        <small class="text-danger">{{ $message }}</small>
                    @enderror
                </div>
                <div class="row">
                    <div class="col-md-6">
                        <div class="form-group">
                            <label for="supplier.telp">Telepon</label>
                            <input type="text" name="supplier.telp" id="supplier.telp" class="form-control" placeholder="Masukkan nomor telepon" value="{{ old('supplier.telp', $supplier->supplier_telp) }}>
                            @error('supplier.telp')
                                <small class="text-danger">{{ $message }}</small>
                            @enderror
                        </div>
                    </div>
                    <div class="col-md-6">
                        <div class="form-group">
                            <label for="supplier.email">Email</label>
                            <input type="email" name="supplier.email" id="supplier.email" class="form-control" placeholder="Masukkan email supplier" value="{{ old('supplier.email', $supplier->supplier_email) }}>
                            @error('supplier.email')
                                <small class="text-danger">{{ $message }}</small>
                            @enderror
                        </div>
                    </div>
                </div>
                <div class="form-group">
                    <label for="supplier.kontak">Kontak Person</label>
                    <input type="text" name="supplier.kontak" id="supplier.kontak" class="form-control" placeholder="Masukkan nama kontak person" value="{{ old('supplier.kontak', $supplier->supplier_kontak) }}>
                    @error('supplier.kontak')
                        <small class="text-danger">{{ $message }}</small>
                    @enderror
                </div>
                <button type="submit" class="btn btn-primary">Perbarui</button>
                <a href="{{ url('/supplier') }}" class="btn btn-secondary">Batal</a>
            </form>
        </div>
    </div>
@endsection
```



f. Buat SupplierModel dan isi propertiya seperti berikut

```
● PS D:\Apps\laragon\www\PWL2025\week 5.2\jobsheet> php artisan make:model SupplierModel
INFO Model [D:\Apps\laragon\www\PWL2025\week 5.2\jobsheet\app\Models\SupplierModel.php] created successfully.

Models > SupplierModel.php > PHP Intelphense > supplierModel
<php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

0 references | 0 implementations
class SupplierModel extends Model
{
    use HasFactory;

    protected $table = 'm_supplier';

    protected $primaryKey = 'supplier_id';

    protected $fillable = [
        'supplier_kode',
        'supplier_nama',
        'supplier_alamat',
        'supplier_telp',
        'supplier_email',
        'supplier_kontak',
    ];
}
```

g. Maka tampilannya akan seperti berikut:

No	Kode Supplier	Nama Supplier	Telepon	Email	Kontak Person	Aksi
1	SUP001	PT. Elektronik Jaya	021-5551234	info@elektronikjaya.com	Budi Santoso	Detail Edit Hapus
2	SUP002	CV. Fashion Indonesia	022-6667890	sales@fashionindonesia.co.id	Dewi Lestari	Detail Edit Hapus
3	SUP003	UD. Makanan Sehat	031-7772345	order@makanansehat.com	Agus Wijaya	Detail Edit Hapus
4	SUP004	PT. Minuman Prima	021-8883456	cs@minumanprima.com	Siti Rahayu	Detail Edit Hapus
5	SUP005	PT. Farma Indonesia	024-9994567	info@farmaindonesia.com	Dr. Hendra Wijaya	Detail Edit Hapus

1) Tambah supplier baru

The screenshot shows a 'Tambah Supplier' (Add Supplier) form. The left sidebar navigation includes 'Data Pengguna', 'Level User', 'Data User', 'Data Barang', 'Kategori Barang', 'Data Barang', 'Data Supplier', 'Data Supplier' (highlighted in blue), 'Data Transaksi', 'Stok Barang', and 'Transaksi Perjaluan'. The main form has fields for 'Nama Supplier' (containing 'PT. Pei Hai'), 'Alamat' (containing 'Jombang'), 'Telepon' (containing '082139631334'), and 'Kontak Person' (containing 'Yonanda'). At the bottom are 'Simpan' and 'Batal' buttons.

Supplier berhasil ditambahkan.

No	Kode Supplier	Nama Supplier	Telepon	Email	Kontak Person	Aksi
1	SUP001	PT. Elektronik Jaya	021-5551234	info@elektronikjaya.com	Budi Santoso	Detail Edit Hapus
2	SUP002	CV. Fashion Indonesia	022-6667890	sales@fashionindonesia.co.id	Dewi Lestari	Detail Edit Hapus
3	SUP003	UD. Makanan Sehat	031-7772345	order@makanansehat.com	Agus Wijaya	Detail Edit Hapus
4	SUP004	PT. Minuman Prima	021-8883456	cs@minumanprima.com	Siti Rahayu	Detail Edit Hapus
5	SUP005	PT. Farma Indonesia	024-9994567	info@farmaindonesia.com	Dr. Hendra Wijaya	Detail Edit Hapus
6	SUP006	PT. Pei Hai	082139631334		Yonanda	Detail Edit Hapus



2) Menampilkan detail supplier baru

Detail Supplier

Detail supplier	
Kode Supplier	SUP006
Nama Supplier	PT. Pei Hai
Alamat	Jombang
Telepon	082139631334
Email	maylayonanda@gmail.com
Kontak Person	Yonanda
Tanggal Dibuat	25-03-2025 16:11:41
Terakhir Diupdate	25-03-2025 16:13:19

Kembali **Edit** **Hapus**

3) Mengedit supplier pengguna

Edit Supplier

Edit supplier	
Kode Supplier	SUP006
Nama Supplier *	PT. Pei Hai
Alamat	Jombang
Telepon	082139631334
Email	maylayonanda@gmail.com
Kontak Person	Yonanda

Perbarui **Batal**

Supplier berhasil diperbarui.

Filter:	- Semua -					
Show	10 entries					
Search:						
No.	Kode Supplier	Nama Supplier	Telepon	Email	Kontak Person	Aksi
1	SUP001	PT. Elektronik Jaya	021-5551234	info@elektronikjaya.com	Budi Santoso	Detail Edit Hapus
2	SUP002	CV. Fashion Indonesia	022-6667890	sales@fashionindonesia.co.id	Dewi Lestari	Detail Edit Hapus
3	SUP003	UD. Makanan Sehat	031-7772345	order@makanansehat.com	Agus Wijaya	Detail Edit Hapus
4	SUP004	PT. Minuman Prima	021-8883456	cs@minumanprima.com	Siti Rahayu	Detail Edit Hapus
5	SUP005	PT. Farma Indonesia	024-9994567	info@farmaindonesia.com	Dr. Hendra Wijaya	Detail Edit Hapus
6	SUP006	PT. Pei Hai	082139631334	maylayonanda@gmail.com	Yonanda	Detail Edit Hapus

Showing 1 to 6 of 6 entries



4) Filtering supplier pengguna

A screenshot of a web-based application interface. At the top, there is a search bar labeled "Filter:" containing "PT. Elektronik Jaya". Below it is a table header with columns: No., Kode Supplier, Nama Supplier, Telepon, Email, Kontak Person, and Aksi. The table contains one entry: SUP001, PT. Elektronik Jaya, 021-5551234, info@elektronikjaya.com, Budi Santoso, with buttons for Detail, Edit, and Hapus. Navigation buttons at the bottom right include "Previous", "1", and "Next".

No.	Kode Supplier	Nama Supplier	Telepon	Email	Kontak Person	Aksi
1	SUP001	PT. Elektronik Jaya	021-5551234	info@elektronikjaya.com	Budi Santoso	<button>Detail</button> <button>Edit</button> <button>Hapus</button>

5) Searching

A screenshot of a web-based application interface. At the top, there is a search bar labeled "Filter:" containing "- Semua -". Below it is a table header with columns: No., Kode Supplier, Nama Supplier, Telepon, Email, Kontak Person, and Aksi. The table contains four entries: SUP001 (PT. Elektronik Jaya), SUP004 (PT. Minuman Prima), SUP005 (PT. Farma Indonesia), and SUP006 (PT. Pei Hai). Each entry has buttons for Detail, Edit, and Hapus. Navigation buttons at the bottom right include "Previous", "1", and "Next".

No.	Kode Supplier	Nama Supplier	Telepon	Email	Kontak Person	Aksi
1	SUP001	PT. Elektronik Jaya	021-5551234	info@elektronikjaya.com	Budi Santoso	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
2	SUP004	PT. Minuman Prima	021-8883456	cs@minumanprima.com	Siti Rahayu	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
3	SUP005	PT. Farma Indonesia	024-9994567	info@farmaindonesia.com	Dr. Hendra Wijaya	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
4	SUP006	PT. Pei Hai	082139631334	maylayonanda@gmail.com	Yonanda	<button>Detail</button> <button>Edit</button> <button>Hapus</button>

6) Menghapus supplier pengguna

A screenshot of a web-based application interface. In the center, a modal dialog box asks "127.0.0.1:8000 says Apakah Anda yakin menghapus data ini?". Below the modal is a table titled "Daftar Supplier" with a header: No., Kode Supplier, Nama Supplier, Telepon, Email, and Kontak Person. The table contains four entries: SUP001 (PT. Elektronik Jaya), SUP004 (PT. Minuman Prima), SUP005 (PT. Farma Indonesia), and SUP006 (PT. Pei Hai). At the bottom of the page, a green success message says "Supplier berhasil dihapus." (Supplier successfully deleted).

No.	Kode Supplier	Nama Supplier	Telepon	Email	Kontak Person
1	SUP001	PT. Elektronik Jaya	021-5551234	info@elektronikjaya.com	Budi Santoso
2	SUP004	PT. Minuman Prima	021-8883456	cs@minumanprima.com	Siti Rahayu
3	SUP005	PT. Farma Indonesia	024-9994567	info@farmaindonesia.com	Dr. Hendra Wijaya
4	SUP006	PT. Pei Hai	082139631334	maylayonanda@gmail.com	Yonanda