

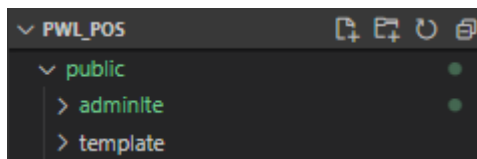
NIM : 2241720095
Nama : Ana Bellatus Mustaqfiro
Kelas : D4 TI - 2H
No. Urut : 04
Mata Kuliah : Pemrograman Web Lanjut Jobsheet 7 (Laravel Starter Code)

Praktikum 1 Layouting AdminLTE

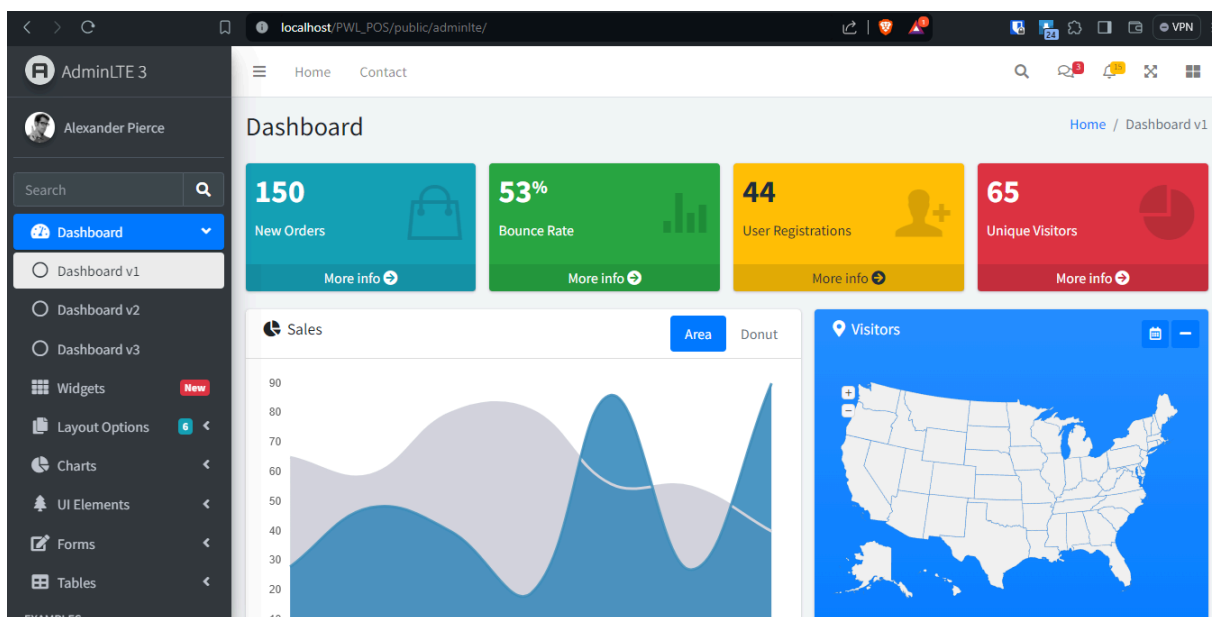
1. Kita download AdminLTE v3.2.0 yang rilis pada 8 Feb 2022



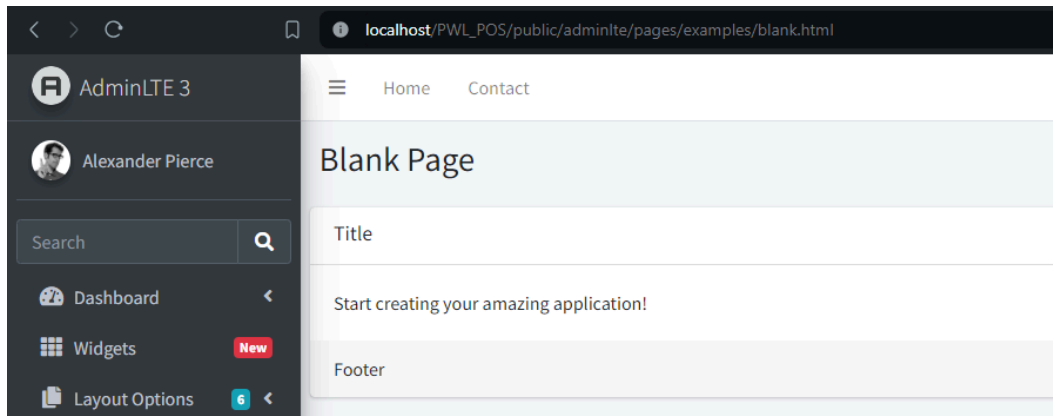
2. Setelah kita berhasil download, kita ekstrak file yang sudah di download ke folder project PWL_POS/public, kemudian kita rename folder cukup menjadi adminlte



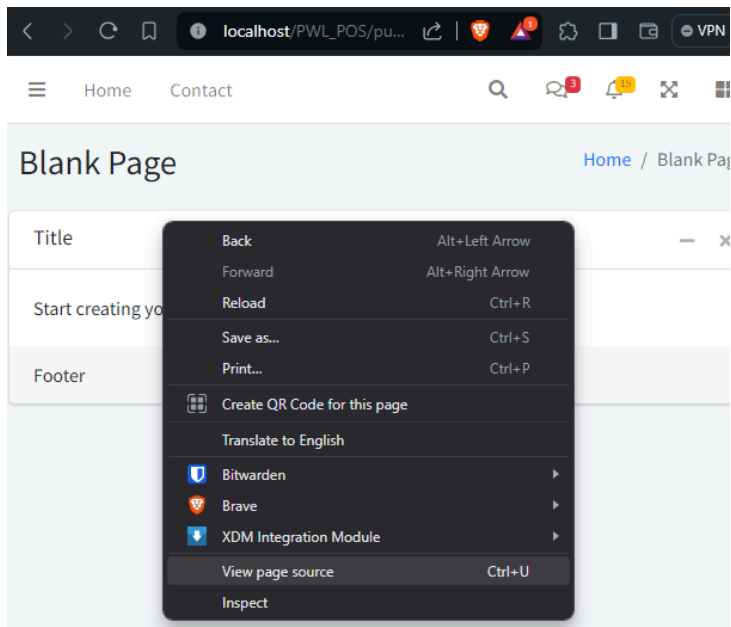
3. Selanjutnya kita buka di browser dengan alamat http://localhost/PWL_POS/public/adminlte maka akan muncul tampilan seperti berikut



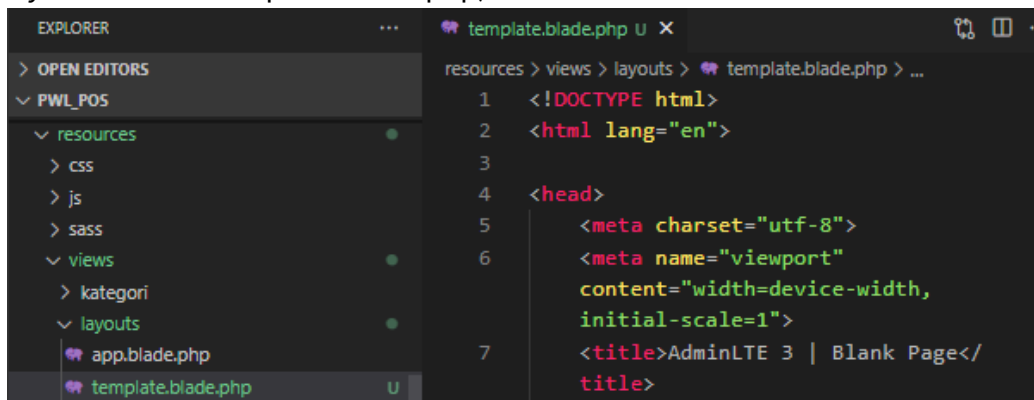
4. Kita klik menu Extras > Blank Page, page inilah yang akan menjadi dasar web template



5. Dari sini kita bisa melakukan layouting halaman Blank Page ini menjadi 4 element seperti pada gambar berikut
6. Selanjutnya kita klik kanan halaman Blank Page dan klik view page source



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



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 > html > head > link
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <title>{{ config('app.name', 'PWL Laravel Starter Code')}}</title>
7
8 <!-- Google Font: Source Sans Pro -->
9 <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">
10 <!-- Font Awesome -->
11 <link rel="stylesheet" href="{{ asset('adminlte/plugins/fontawesome-free/css/all.min.css') }}">
12 <!-- Theme style -->
13 <link rel="stylesheet" href="{{ asset('adminlte/dist/css/adminlte.min.css') }}">
14 </head>
```

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

```
resources > views > layouts > template.blade.php > html > body.hold-transition.sidebar-mini > div.wrapper
18 <!-- Navbar -->
19 @include('layouts.header')
20 <!-- /.navbar -->
```

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

```
resources > views > layouts > template.blade.php > html > body.hold-transition.sidebar-mini > div.wrapper > aside.main-sidebar.sidebar-dark-primary.elevation-4 > a.brand-link > img.brand-
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 
  <span class="brand-text font-weight-light">PWL - Starter Code</span>
27 </a>
28
```

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 > template.blade.php > html > body.hold-transition.sidebar-mini > div.wrapper
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>
```

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 > template.blade.php > ...
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 > template.blade.php > ...
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

```
layouts > template.blade.php > html > body.hold-transition.sidebar-mini > div.wrapper > div.co
35 <!-- Content Wrapper. Contains page content -->
36 <div class="content-wrapper">
37 <!-- Content Header (Page header) -->
38 @include('layouts.breadcrumb')
39
```

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')

```
resources > views > layouts > template.blade.php > html > body.hold-transition.sidebar-mini
40 <!-- Main content -->
41 <section class="content">
42 @yield('content')
43 </section>
44
45 <!-- /.content -->
```

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

```
resources > views > layouts > template.blade.php > html > body.hold-transition.sidebar-mini > div.wrapper > div.content-wrapper > section.content
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 -->
```

20. Selamat kalian sudah selesai dalam melakukan layouting website di laravel

Praktikum 2 - Penerapan Layouting

1. Kita buat file controller dengan nama WelcomeController.php

```
app > Http > Controllers > WelcomeController.php > ...
7 class WelcomeController extends Controller
8 {
9     public function index()
10     {
11         $breadcrumb = (object)[
12             'title' => 'Selamat Datang',
13             'list' => ['Home', 'Welcome']
14         ];
15
16         $activeMenu = 'dashboard';
17
18         return view('welcome', ['breadcrumb' => $breadcrumb, 'activeMenu' => $activeMenu]);
19     }
20 }
```

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

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

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

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

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

```
resources > views > layouts > sidebar.blade.php > div.sidebar
1 <div class="sidebar">
2   <!-- Sidebar Search 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  <!-- Sidebar Menu -->
14  <nav class="mt-2">
15    <ul class="nav nav-pills nav-sidebar flex-column" data-widget="treeview" role="menu" data-accordion="false">
16      <li class="nav-item">
17        <a href="{{ url('/') }}" class="nav-link {{ $activeMenu == 'dashboard' ? 'active' : '' }}">
18          <i class="nav-icon fas fa-tachometer-alt"></i>
19          <p>Dashboard</p>
20        </a>
21      </li>
22      <li class="nav-header">Data Pengguna</li>
23      <li class="nav-item">
24        <a href="{{ url('/level') }}" class="nav-link {{ $activeMenu == 'level' ? 'active' : '' }}">
25          <i class="nav-icon fas fa-layer-group"></i>
26          <p>Level User</p>
27        </a>
28      </li>
29      <li class="nav-item">
30        <a href="{{ url('/user') }}" class="nav-link {{ $activeMenu == 'user' ? 'active' : '' }}">
31          <i class="nav-icon far fa-user"></i>
32          <p>Data User</p>
33        </a>
34      </li>
35      <li class="nav-header">Data Barang</li>
36      <li class="nav-item">
37        <a href="{{ url('/kategori') }}" class="nav-link {{ $activeMenu == 'kategori' ? 'active' : '' }}">
38          <i class="nav-icon far fa-bookmark"></i>
39          <p>Kategori Barang</p>
40        </a>
41      </li>
42      <li class="nav-item">
43        <a href="{{ url('/barang') }}" class="nav-link {{ $activeMenu == 'barang' ? 'active' : '' }}">
44          <i class="nav-icon far fa-list-alt"></i>
45          <p>Data Barang</p>
46        </a>
47      </li>
```

```
resources > views > layouts > sidebar.blade.php > div.sidebar > nav.mt-2 > ul.nav.nav-pills.nav-sidebar.flex-column > li.nav-item
48 <li class="nav-header">Data Transaksi</li>
49 <li class="nav-item">
50 <a href="{{ url('/stok') }}" class="nav-link {{ $activeMenu == 'stok' ? 'active' : '' }}">
51 <i class="nav-icon fas fa-cubes"></i>
52 <p>Stok Barang</p>
53 </a>
54 </li>
55 <li class="nav-item">
56 <a href="{{ url('/barang') }}" class="nav-link {{ $activeMenu == 'penjualan' ? 'active' : '' }}">
57 <i class="nav-icon fas fa-cash-register"></i>
58 <p>Transaksi Penjualan</p>
59 </a>
60 </li>
61 </ul>
62 </nav>
63 </div>
```

5. Kita tambahkan kode berikut router web.php

```
routes > web.php > ...
52 Route::get('/', [WelcomeController::class, 'index']);
```

6. Sekarang kita coba jalankan di browser dengan mengetikkan url `http://localhost/PWL_POS/public`

Praktikum 3 - Implementasi jQuery Datatable di AdminLTE

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

composer require yajra/laravel-datatables:^10.0 atau
composer require yajra/laravel-datatables-oracle

```
PS C:\laragon\www\PWL_POS> composer require yajra/laravel-datatables:^10.0
The "10.0" constraint for "yajra/laravel-datatables" appears too strict and will likely not match what you want. See https://getcomposer.org/constraints
./composer.json has been updated
Running composer update yajra/laravel-datatables
Loading composer repositories with package information
Updating dependencies
Nothing to modify in lock file
Installing dependencies from lock file (including require-dev)
Nothing to install, update or remove
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

 INFO  Discovering packages.

jeroennoten/laravel-adminlte ..... DONE
laravel/sail ..... DONE
laravel/sanctum ..... DONE
laravel/tinker ..... DONE
laravel/ui ..... DONE
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE
spatie/laravel-ignition ..... DONE
yajra/laravel-datatables-buttons ..... DONE
yajra/laravel-datatables-editor ..... DONE
yajra/laravel-datatables-fractal ..... DONE
yajra/laravel-datatables-html ..... DONE
yajra/laravel-datatables-oracle ..... DONE

89 packages you are using are looking for funding.
Use the "composer fund" command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi --force

 INFO  No publishable resources for tag [laravel-assets].

No security vulnerability advisories found.
```

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

```
routes > web.php > ...  
53 Route::get('/', [WelcomeController::class, 'index']);  
54  
55 Route::group(['prefix' => 'user'], function(){  
56     Route::get('/', [UserController::class, 'index']);  
57     Route::post('/list', [UserController::class, 'list']);  
58     Route::get('/create', [UserController::class, 'create']);  
59     Route::post('/', [UserController::class, 'store']);  
60     Route::get('/{id}', [UserController::class, 'show']);  
61     Route::get('/{id}/edit', [UserController::class, 'edit']);  
62     Route::put('/{id}', [UserController::class, 'update']);  
63     Route::delete('/{id}', [UserController::class, 'destroy']);  
64 });
```

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

```
app > Http > Controllers > UserController.php > ...  
12 class UserController extends Controller  
13 {  
14     public function index()  
15     {  
16         $breadcrumb = (object)[  
17             'title' => 'Daftar User',  
18             'list' => ['Home', 'User']  
19         ];  
20  
21         $page = (object)[  
22             'title' => 'Daftar user yang terdaftar dalam sistem'  
23         ];  
24  
25         $activeMenu = 'user';  
26  
27         return view('user.index', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);  
28     }
```

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

```
resources > views > user > index.blade.php > ...  
1 @extends('layouts.template')  
2  
3 @section('content')  
4     <div class="card card-outline card-primary">  
5         <div class="card-header">  
6             <h3 class="card-title">{{ $page->title }}</h3>  
7             <div class="card-tools">  
8                 <a class="btn btn-sm btn-primary mt-1" href="{{ url('user/create') }}">Tambah</a>  
9             </div>  
10        </div>  
11        <div class="card-body">  
12            <table class="table table-bordered table-striped table-hover table-sm" id="table_user">  
13                <thead>  
14                    <tr>  
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
```



```

resources > views > user > index.blade.php > ...
27 | @push('css')
28 | @endpush
29 |
30 | @push('js')
31 | <script>
32 |     $(document).ready(function() {
33 |         var dataUser = $('#table_user').DataTable({
34 |             serverSide: true, // serverSide: true, jika ingin menggunakan server
35 |             side processing
36 |             ajax: {
37 |                 "url": "{{ url('user/list') }}",
38 |                 "dataType": "json",
39 |                 "type": "POST"
40 |             },
41 |             columns: [{
42 |                 data: "DT_RowIndex", // nomor urut dari laravel datatable
43 |                 addIndexColumn()
44 |                 className: "text-center",
45 |                 orderable: false,
46 |                 searchable: false
47 |             }, {
48 |                 data: "username",
49 |                 className: "",
50 |                 orderable: true, // orderable: true, jika ingin kolom ini bisa diurutkan
51 |                 searchable: true // searchable: true, jika ingin kolom ini bisa dicari
52 |             }, {
53 |                 data: "nama",
54 |                 className: "",
55 |                 orderable: true, // orderable: true, jika ingin kolom ini bisa diurutkan
56 |                 searchable: true // searchable: true, jika ingin kolom ini bisa dicari
57 |             }, {
58 |                 data: "level.level_nama",
59 |                 className: "",
60 |                 orderable: false, // orderable: true, jika ingin kolom ini bisa diurutkan
61 |                 searchable: false // searchable: true, jika ingin kolom ini bisa dicari
62 |             }, {
63 |                 data: "aksi",
64 |                 className: "",
65 |                 orderable: false, // orderable: true, jika ingin kolom ini bisa diurutkan
66 |                 searchable: false // searchable: true, jika ingin kolom ini bisa dicari
67 |             }
68 |         ]
69 |     });
70 | </script>
71 | @endpush

```

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

```

resources > views > layouts > template.blade.php > html > body,hold-transition,sidebar-mini
8 | <meta name="csrf-token" content="{{ csrf_token() }}"> <!-- Untuk mengirimkan token laravel CSRF pada setiap request ajax -->
9 |
10 | <!-- Google Font: Source Sans Pro -->
11 | <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">
12 | <!-- Font Awesome -->
13 | <link rel="stylesheet" href="{{ asset('adminlte/plugins/fontawesome-free/css/all.min.css') }}">
14 | <!-- DataTables -->
15 | <link rel="stylesheet" href="{{ asset('adminlte/plugins/datatables-bs4/css/dataTables.bootstrap4.min.css') }}">
16 | <link rel="stylesheet" href="{{ asset('adminlte/plugins/datatables-responsive/css/responsive.bootstrap4.min.css') }}">
17 | <link rel="stylesheet" href="{{ asset('adminlte/plugins/datatables-buttons/css/buttons.bootstrap4.min.css') }}">
18 | <!-- Theme style -->
19 | <link rel="stylesheet" href="{{ asset('adminlte/dist/css/adminlte.min.css') }}">
20 |
21 | @stack('css')

```

```

resources > views > layouts > template.blade.php > html > body.hold-transition.sidebar-mini
62 <!-- Bootstrap 4 -->
63 <script src="{{ asset('adminlte/plugins/bootstrap/js/bootstrap.bundle.min.js') }}"></script>
64 <!-- DataTables & Plugins -->
65 <script src="{{ asset('adminlte/plugins/datatables/jquery.dataTables.min.js') }}"></script>
66 <script src="{{ asset('adminlte/plugins/datatables-bs4/js/dataTables.bootstrap4.min.js') }}"></script>
67 <script src="{{ asset('adminlte/plugins/datatables-responsive/js/dataTables.responsive.min.js') }}"></script>
68 <script src="{{ asset('adminlte/plugins/datatables-responsive/js/responsive.bootstrap4.min.js') }}"></script>
69 <script src="{{ asset('adminlte/plugins/datatables-buttons/js/dataTables.buttons.min.js') }}"></script>
70 <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.bootstrap4.min.js') }}"></script>
71 <script src="{{ asset('adminlte/plugins/jszip/jszip.min.js') }}"></script>
72 <script src="{{ asset('adminlte/plugins/pdfmake/pdfmake.min.js') }}"></script>
73 <script src="{{ asset('adminlte/plugins/pdfmake/vfs_fonts.js') }}"></script>
74 <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.html5.min.js') }}"></script>
75 <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.print.min.js') }}"></script>
76 <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.colvis.min.js') }}"></script>
77 <!-- AdminLTE App -->
78 <script src="{{ asset('adminlte/dist/js/adminlte.min.js') }}"></script>
79 <script>
80     // Untuk mengirimkan token Laravel CSRF pada setiap request ajax
81     $.ajaxSetup({headers: {'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')}});
82 </script>
83 @stack('js') <!-- Digunakan untuk memanggil custom js dari perintah push('js') pada masing-masing view -->
84 </body>

```

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

```

app > Http > Controllers > UserController.php > UserController > list > Closure
31 public function list(Request $request)
32 {
33     $users = UserModel::select('user_id', 'username', 'nama', 'level_id')
34     ->with('level');
35     return DataTables::of($users)
36     ->addIndexColumn() // menambahkan kolom index / no urut (default nama kolom: DT_RowIndex)
37     ->addColumn('aksi', function ($user) { // menambahkan kolom aksi
38         $btn = '<a href="' . url('/user/' . $user->user_id) . '" class="btn btninfo btn-sm">Detail</a> ';
39         $btn .= '<a href="' . url('/user/' . $user->user_id) . '/edit' . '" class="btn btn-warning btn-sm">Edit</a> ';
40         $btn .= '<form class="d-inline-block" method="POST" action="' .
41             url('/user/' . $user->user_id) . '">
42             . csrf_field() . method_field('DELETE') .
43             '<button type="submit" class="btn btn-danger btn-sm" onclick="return confirm('\'Apakah Anda yakin menghapus data ini?\'');">Hapus</button></form>';
44         return $btn;
45     });
46     ->rawColumns(['aksi']) // memberitahu bahwa kolom aksi adalah html
47     ->make(true);
48 }

```

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

localhost:PWL_POS/public/user

PWL - Starter Code

Search

Dashboard

Data Pengguna

Level User

Data User

Data Barang

Kategori Barang

Data Barang

Data Transaksi

Stok Barang

Transaksi Penjualan

Home Contact

Daftar User

Home / User

Daftar user yang terdaftar dalam sistem

Tambah

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	Staff/Kasir	Staff/Kasir	Detail Edit Hapus
4	customer-1	Pelanggan Pertama	Pelanggan	Detail Edit Hapus
5	manager_dua	Manager 2	Manager	Detail Edit Hapus
6	manager22	Manager Dua Dua	Manager	Detail Edit Hapus
7	manager33	Manager Tiga Tiga	Manager	Detail Edit Hapus
8	manager56	Manager55	Manager	Detail Edit Hapus

9. Selanjutnya kita modifikasi UserController.php untuk form tambah data user

```
app > Http > Controllers > UserController.php > UserController > store
47 public function create()
48 {
49     $breadcrumb = (object)[
50         'title' => 'Tambah User',
51         'list' => ['Home', 'User', 'Tambah']
52     ];
53
54     $page = (object)[
55         'title' => 'Tambah user baru'
56     ];
57
58     $level = LevelModel::all();
59     $activeMenu = 'user';
60
61     return view('user.create', ['breadcrumb' => $breadcrumb, 'page' => $page, 'level' => $level, 'activeMenu' => $activeMenu]);
62 }
```

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

```
resources > views > user > create.blade.php > div.card.card-outline.card-primary > div.card-body > form.form-horizontal > div.form-group.row > div.col-11 > input#username
1 @extends('layouts.template')
2
3 @section('content')
4     <div class="card card-outline card-primary">
5         <div class="card-header">
6             <h3 class="card-title">{{ $page->title }}</h3>
7             <div class="card-tools"></div>
8         </div>
9         <div class="card-body">
10             <form method="POST" action="{{ url('user') }}" class="form-horizontal">
11                 @csrf
12                 <div class="form-group row">
13                     <label class="col-1 control-label col-form-label">Level</label>
14                     <div class="col-11">
15                         <select class="form-control" id="level_id" name="level_id" required>
16                             <option value="">- Pilih Level -</option>
17                             @foreach ($level as $item)
18                                 <option value="{{ $item->level_id }}">{{ $item->level_nama }}</option>
19                             @endforeach
20                         </select>
21                         @error('level_id')
22                             <small class="form-text text-danger">{{ $message }}</small>
23                         @enderror
24                     </div>
25                 </div>
```

```
resources > views > user > create.blade.php > div.card.card-outline.card-primary > div.card-body
26 <div class="form-group row">
27     <label class="col-1 control-label col-form-label">Username</label>
28     <div class="col-11">
29         <input type="text" class="form-control" id="username" name="username"
30             value="{{ old('username') }}" required>
31         @error('username')
32             <small class="form-text text-danger">{{ $message }}</small>
33         @enderror
34     </div>
35 </div>
36 <div class="form-group row">
37     <label class="col-1 control-label col-form-label">Nama</label>
38     <div class="col-11">
39         <input type="text" class="form-control" id="nama" name="nama" value="{{ old('nama') }}"
40             required>
41         @error('nama')
42             <small class="form-text text-danger">{{ $message }}</small>
43         @enderror
44     </div>
45 </div>
46 <div class="form-group row">
47     <label class="col-1 control-label col-form-label">Password</label>
48     <div class="col-11">
49         <input type="password" class="form-control" id="password" name="password" required>
50         @error('password')
51             <small class="form-text text-danger">{{ $message }}</small>
52         @enderror
53     </div>
54 </div>
```

```

resources > views > user > create.blade.php > div.card.card-outline.card-primary > div.card-body
55 |
56 |         <div class="form-group row">
57 |             <label class="col-1 control-label col-form-label"></label>
58 |             <div class="col-11">
59 |                 <button type="submit" class="btn btn-primary btn-sm">Simpan</button>
60 |                 <a class="btn btn-sm btn-default ml-1" href="{{ url('user') }}">Kembali</a>
61 |             </div>
62 |         </div>
63 |     </form>
64 | </div>
65 | @endsection
66 |
67 | @push('css')
68 | @endpush
69 |
70 | @push('js')
71 | @endpush

```

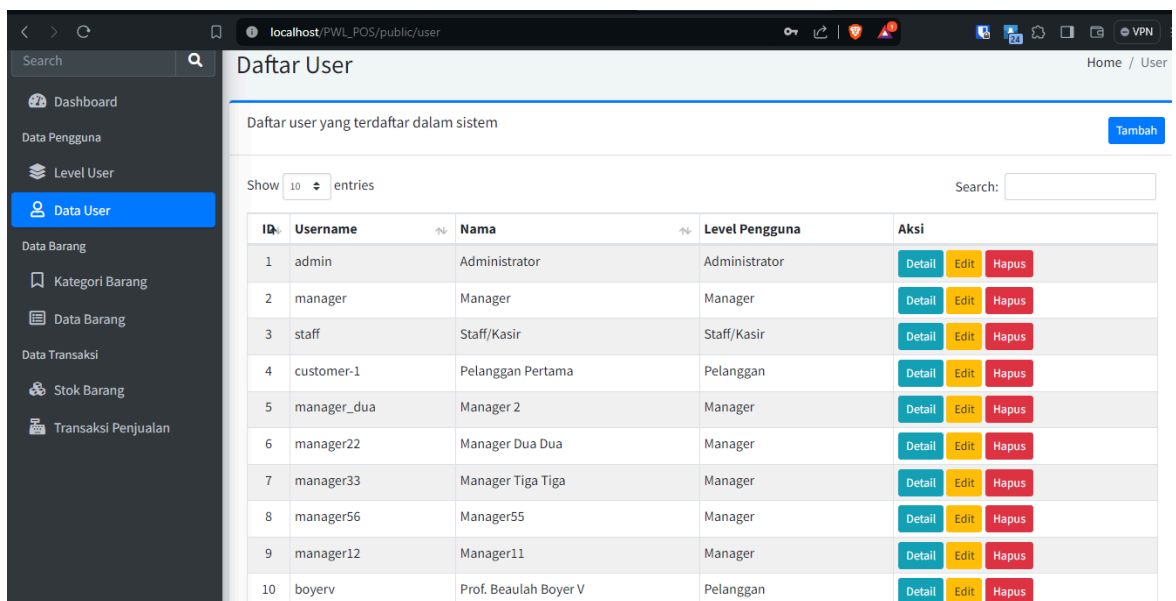
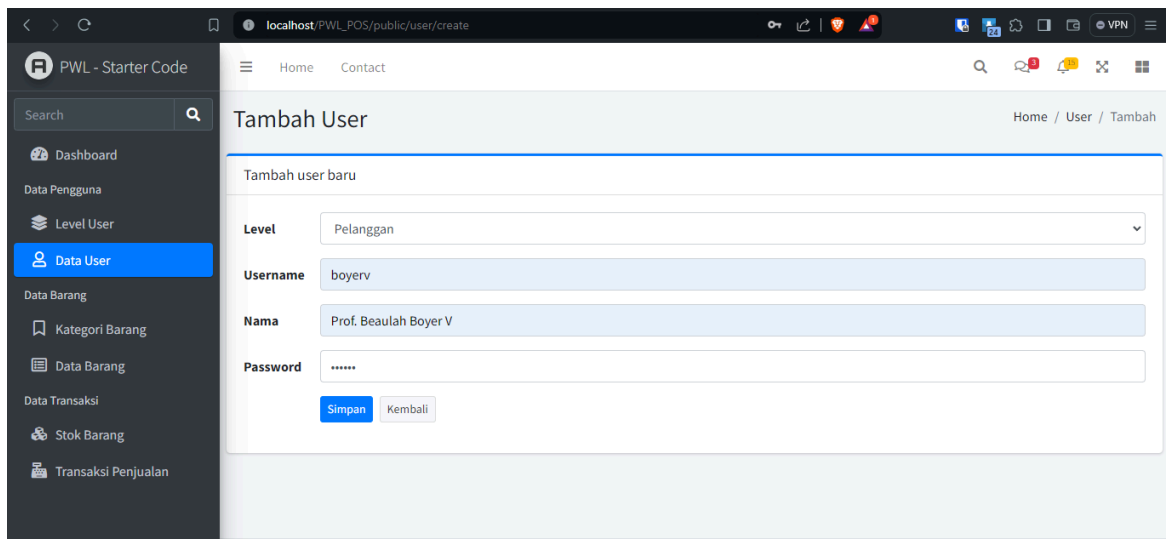
11. Kemudian untuk bisa menghandle data yang akan disimpan ke database, kita buat fungsi store() di UserController.php

```

app > Http > Controllers > UserController.php > UserController > edit
64 | public function store(Request $request)
65 | {
66 |     $request->validate([
67 |         'username' => 'required|string|min:3|unique:m_user,username',
68 |         'nama'      => 'required|string|max:100',
69 |         'password'  => 'required|min:5',
70 |         'level_id'  => 'required|integer',
71 |     ]);
72 |
73 |     UserModel::create([
74 |         'username' => $request->username,
75 |         'nama'     => $request->nama,
76 |         'password' => bcrypt($request->password),
77 |         'level_id' => $request->level_id,
78 |     ]);
79 |
80 |     return redirect('/user')->with('success', 'Data user berhasil disimpan');
81 | }

```

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



13. Selanjutnya, kita masuk pada bagian menampilkan detail data user

(klik tombol **Detail**) pada halaman user. Route yang bertugas untuk menangkap request detail adalah

```
routes > web.php > Closure
60 + | Route::get('/{id}', [UserController::class, 'show']);
```

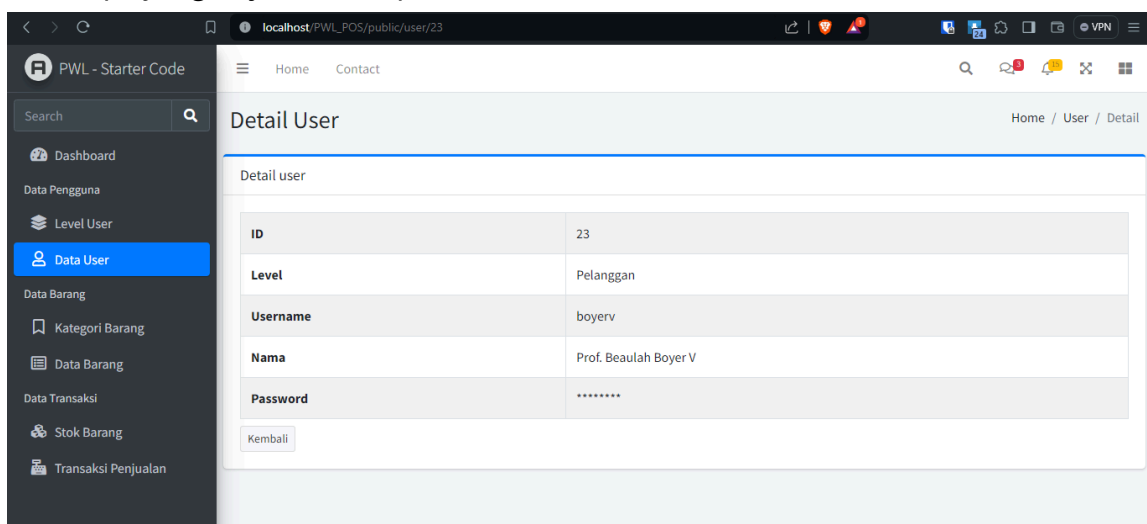
14. Jadi kita buat/modifikasi fungsi show() pada UserController.php seperti berikut

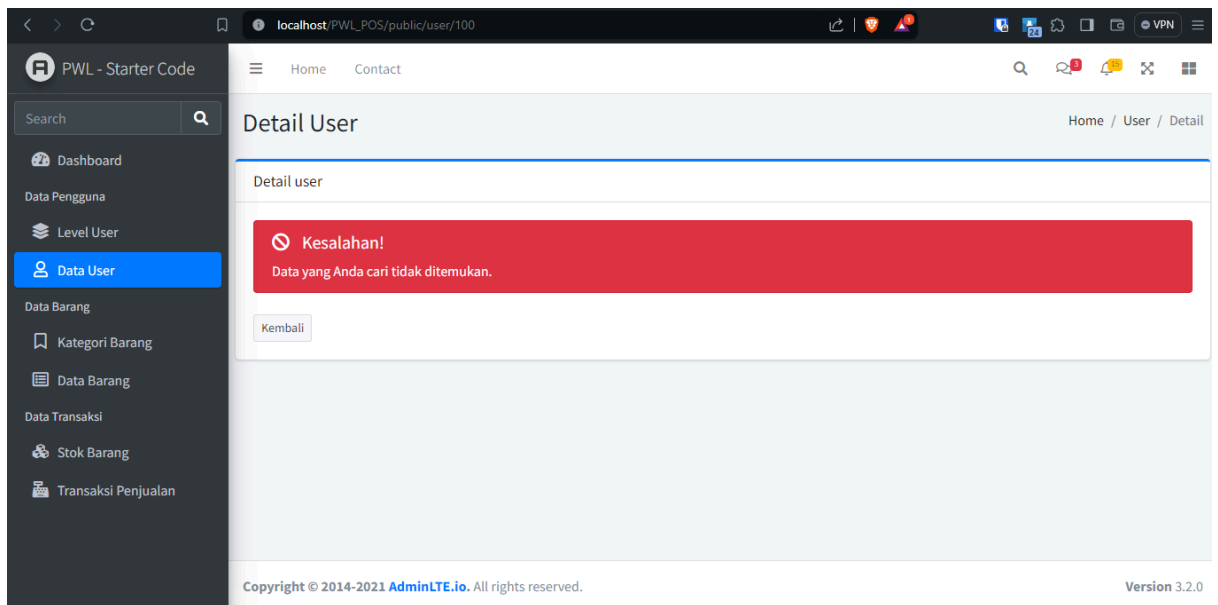
```
app > Http > Controllers > UserController.php > UserController > edit
83 public function show(string $id){
84     $user = UserModel::with('level')->find($id);
85
86     $breadcrumb = (object)[
87         'title' => 'Detail User',
88         'list' => ['Home', 'User', 'Detail']
89     ];
90
91     $page = (object)[
92         'title' => 'Detail user'
93     ];
94
95     $activeMenu = 'user';
96
97     return view('user.show', ['breadcrumb' => $breadcrumb, 'page' => $page, 'user' => $user, 'activeMenu' => $activeMenu]);
98 }
```

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

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

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





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

```
routes > web.php > ...
61 Route::get('/{id}/edit', [UserController::class, 'edit']);
62 Route::put('/{id}', [UserController::class, 'update']);
```

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

```
app > Http > Controllers > UserController.php > UserController > update
100 public function edit(string $id)
101 {
102     $user = UserModel::find($id);
103     $level = LevelModel::all();
104
105     $breadcrumb = (object)[
106         'title' => 'Edit User',
107         'list' => ['Home', 'User', 'Edit']
108     ];
109
110     $page = (object)[
111         'title' => 'Edit user'
112     ];
113
114     $activeMenu = 'user';
115
116     return view('user.edit', ['breadcrumb' => $breadcrumb, 'page' => $page, 'user' => $user, 'level' => $level, 'activeMenu' => $activeMenu]);
117 }
```

```
app > Http > Controllers > UserController.php > UserController > delete
120 public function update(Request $request, string $id)
121 {
122     $request->validate([
123         'username' => 'required|string|min:3|unique:m_user,username,' . $id . ',user_id',
124         'nama' => 'required|string|max:100',
125         'password' => 'nullable|min:5',
126         'level_id' => 'required|integer',
127     ]);
128
129     UserModel::find($id)->update([
130         'username' => $request->username,
131         'nama' => $request->nama,
132         'password' => $request->password ? bcrypt($request->password) : UserModel::find($id)->password,
133         'level_id' => $request->level_id,
134     ]);
135
136     return redirect('/user')->with('success', 'Data user berhasil diubah');
137 }
```

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

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


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

localhost/PWL_POS/public/user/23/edit

PWL - Starter Code

Search

Dashboard

Data Pengguna

Level User

Data User

Data Barang

Kategori Barang

Data Barang

Data Transaksi

Stok Barang

Transaksi Penjualan

Home Contact

Edit User

Home / User / Edit

Edit user

Level: Pelanggan

Username: boyerv

Nama: Prof. Beaulah Boyer V

Password:

Abaikan (jangan diisi) jika tidak ingin mengganti password user.

[Simpan](#) [Kembali](#)

localhost/PWL_POS/public/user/23/edit

PWL - Starter Code

Search

Dashboard

Data Pengguna

Level User

Data User

Data Barang

Kategori Barang

Data Barang

Data Transaksi

Stok Barang

Transaksi Penjualan

Home Contact

Edit User

Home / User / Edit

Edit user

Level: Staff/Kasir

Username: boyerv

Nama: Prof. Beaulah Boyer V

Password:

Abaikan (jangan diisi) jika tidak ingin mengganti password user.

[Simpan](#) [Kembali](#)

localhost/PWL_POS/public/user

PWL - Starter Code

Search

Dashboard

Data Pengguna

Level User

Data User

Data Barang

Kategori Barang

Data Barang

Data Transaksi

Stok Barang

Transaksi Penjualan

Home / User

Daftar User

Daftar user yang terdaftar dalam sistem

[Tambah](#)

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	Staff/Kasir	Staff/Kasir	Detail Edit Hapus
4	customer-1	Pelanggan Pertama	Pelanggan	Detail Edit Hapus
5	manager_dua	Manager 2	Manager	Detail Edit Hapus
6	manager22	Manager Dua Dua	Manager	Detail Edit Hapus
7	manager33	Manager Tiga Tiga	Manager	Detail Edit Hapus
8	manager56	Manager55	Manager	Detail Edit Hapus
9	manager12	Manager11	Manager	Detail Edit Hapus
10	boyerv	Prof. Beaulah Boyer V	Staff/Kasir	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']);`

```
routes > web.php > Closure
63 | Route::delete('/{id}', [UserController::class, 'destroy']);
```

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

```
app > Http > Controllers > UserController.php > ...
139 | public function destroy(string $id)
140 | {
141 |     $check = UserModel::find($id);
142 |     if (!$check) {
143 |         return redirect('/user')->with('error', 'Data user tidak ditemukan');
144 |     }
145 |     try {
146 |         UserModel::destroy($id);
147 |         return redirect('/user')->with('success', 'Data user berhasil dihapus');
148 |     } catch (\Illuminate\Database\QueryException $e) {
149 |         return redirect('/user')->with('error', 'Data user gagal dihapus karena masih terdapat tabel lain yang terkait dengan data ini');
150 |     }
151 | }
```

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

```
resources > views > user > index.blade.php > script
11 | <div class="card-body">
12 |     @if (session('success'))
13 |         <div class="alert alert-success">{{ session('success') }}</div>
14 |     @endif
15 |     @if (session('error'))
16 |         <div class="alert alert-danger">{{ session('error') }}</div>
17 |     @endif
```

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

localhost/PWL_POS/public/user

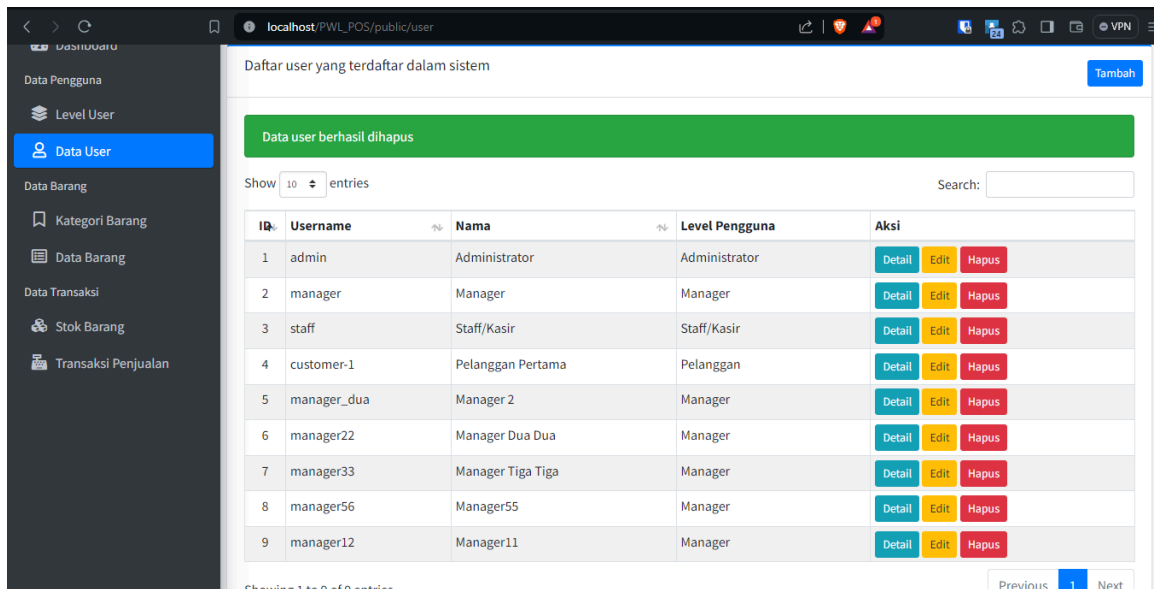
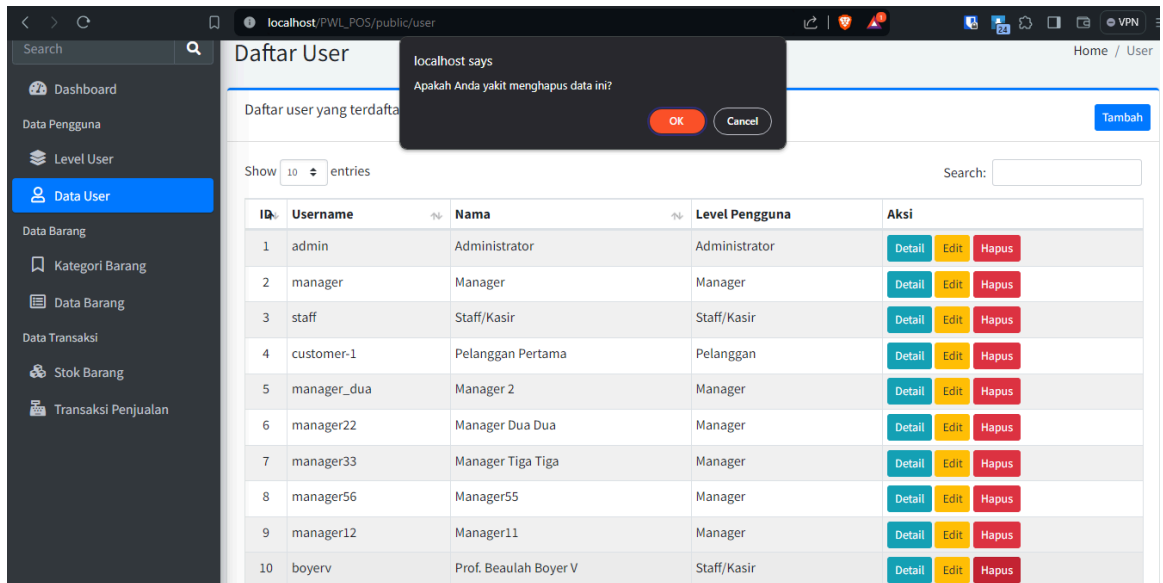
Daftar User

Daftar user yang terdaftar dalam sistem

Search:

Show 10 entries

ID	Username	Nama	Level Pengguna	Aksi
1	admin	Administrator	Administrator	Detail Edit Hapus
2	manager	Manager	Manager	Detail Edit Hapus
3	staff	Staff/Kasir	Staff/Kasir	Detail Edit Hapus
4	customer-1	Pelanggan Pertama	Pelanggan	Detail Edit Hapus
5	manager_dua	Manager 2	Manager	Detail Edit Hapus
6	manager22	Manager Dua Dua	Manager	Detail Edit Hapus
7	manager33	Manager Tiga Tiga	Manager	Detail Edit Hapus
8	manager56	Manager55	Manager	Detail Edit Hapus
9	manager12	Manager11	Manager	Detail Edit Hapus
10	boyerv	Prof. Beaulah Boyer V	Staff/Kasir	Detail Edit Hapus



25. Selamat, kalian sudah membuat Laravel Starter Code untuk proses CRUD dengan menggunakan template AdminLTE dan plugin jQuery Datatables.

Praktikum 4 - Implementasi Filtering Datatables

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

```
app > Http > Controllers > UserController.php > UserController > index
15 | public function index()
16 | {
17 |     $breadcrumb = (object)[
18 |         'title' => 'Daftar User',
19 |         'list' => ['Home', 'User']
20 |     ];
21 |
22 |     $page = (object)[
23 |         'title' => 'Daftar user yang terdaftar dalam sistem'
24 |     ];
25 |
26 |     $activeMenu = 'user';
27 |     $level = LevelModel::all();
28 |
29 |     return view('user.index', ['breadcrumb' => $breadcrumb, 'page' => $page, 'level' => $level, 'activeMenu' => $activeMenu]);
30 | }
```

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

```
resources > views > user > index.blade.php > div.card.card-outline.card-primary > div.card-body > div.row
15      @if (session('error'))
16          <div class="alert alert-danger">{{ session('error') }}</div>
17      @endif
18      <div class="row">
19          <div class="col-md-12">
20              <div class="form-group row">
21                  <label class="col-1 control-label col-form-label">Filter:</label>
22                  <div class="col-3">
23                      <select name="level_id" id="level_id" class="form-control" required>
24                          <option value="">- Semua -</option>
25                          @foreach ($level as $item)
26                              <option value="{{ $item->level_id }}">{{ $item->level_nama }}</option>
27                          @endforeach
28                      </select>
29                      <small class="form-text text-muted">Level Pengguna</small>
30                  </div>
31              </div>
32          </div>
33      </div>
```

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

```
resources > views > user > index.blade.php > script > <function> > dataUser
52      @push('js')
53          <script>
54              $(document).ready(function() {
55                  var dataUser = $('#table_user').DataTable({
56                      serverSide: true, // serverSide: true, jika ingin menggunakan server side processing
57                      ajax: {
58                          "url": "{{ url('user/list') }}",
59                          "dataType": "json",
60                          "type": "POST"
61                          "data": function(d) {
62                              d.level_id = $('#level_id').val();
63                          }
64                      }
65                  });
66              }
67          </script>
68      @endpush
```

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

```
resources > views > user > index.blade.php > script > <function>
90      }
91      });
92
93      $('#level_id').on('change', function() {
94          dataUser.ajax.reload();
95      });
96  });
97  </script>
98  @endpush
```

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

```
app > Http > Controllers > UserController.php > UserController > list > Closure
32      public function list(Request $request)
33      {
34          $users = UserModel::select('user_id', 'username', 'nama',
35                                  'level_id')->with('level');
36
37          if ($request->level_id) {
38              $users->where('level_id', $request->level_id);
39          }
40      }
```

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

The screenshots show a web application interface for managing users. The left sidebar contains a search bar and a menu with items: Dashboard, Data Pengguna, Level User, Data User (highlighted), Data Barang, Kategori Barang, Data Barang, Data Transaksi, Stok Barang, and Transaksi Penjualan. The main content area is titled 'Daftar User' and includes a 'Tambah' button. Below the title is a filter dropdown set to '- Semua -' and a 'Show 10 entries' indicator. A search bar is also present. The table displays user data with columns: ID, Username, Nama, Level Pengguna, and Aksi. The first screenshot shows 6 entries. The second screenshot shows 6 entries filtered by 'Manager'. The third screenshot shows 1 entry filtered by 'Semua' with a search for 'admin'. The footer includes copyright information and the version number 3.2.0.

Daftar User

Daftar user yang terdaftar dalam sistem

Filter: - Semua -

Level Pengguna

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	Staff/Kasir	Staff/Kasir	Detail Edit Hapus
4	customer-1	Pelanggan Pertama	Pelanggan	Detail Edit Hapus
5	manager_dua	Manager 2	Manager	Detail Edit Hapus
6	manager22	Manager Dua Dua	Manager	Detail Edit Hapus

Daftar User

Daftar user yang terdaftar dalam sistem

Filter: Manager

Level Pengguna

Show 10 entries

Search:

ID	Username	Nama	Level Pengguna	Aksi
1	manager	Manager	Manager	Detail Edit Hapus
2	manager_dua	Manager 2	Manager	Detail Edit Hapus
3	manager22	Manager Dua Dua	Manager	Detail Edit Hapus
4	manager33	Manager Tiga Tiga	Manager	Detail Edit Hapus
5	manager56	Manager55	Manager	Detail Edit Hapus
6	manager12	Manager11	Manager	Detail Edit Hapus

Daftar User

Daftar user yang terdaftar dalam sistem

Filter: - Semua -

Level Pengguna

Show 10 entries

Search: admin

ID	Username	Nama	Level Pengguna	Aksi
1	admin	Administrator	Administrator	Detail Edit Hapus

Showing 1 to 1 of 1 entries (filtered from 9 total entries)

Previous 1 Next

Copyright © 2014-2021 AdminLTE.io. All rights reserved. Version 3.2.0

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

F. Pertanyaan

Jawablah pertanyaan berikut sesuai pemahaman materi di atas

1. Apa perbedaan frontend template dengan backend template?
 - Frontend Template
untuk mengatur tampilan (UI/UX) yang dilihat dan diakses oleh pengguna, biasanya berisi elemen-elemen HTML, CSS, dan JavaScript yang membentuk tampilan yang interaktif dan responsif.
 - Backend Template:
untuk mengatur tampilan dan logika yang berhubungan dengan sisi administratif atau pengelolaan website, biasanya digunakan oleh user dengan level administrator, manajemen dan sebagainya yang berfokus pada manajemen website.
2. Apakah layouting itu penting dalam membangun sebuah website?
ya, layouting penting dalam membangun sebuah website karena layouting mempengaruhi cara pengguna berinteraksi dengan website, membantu menentukan navigasi, dan memastikan informasi yang ditampilkan dapat dengan mudah dipahami oleh pengguna.
3. Jelaskan fungsi dari komponen laravel blade berikut : @include(), @extend(), @section(), @push(), @yield(), dan @stack()
4. Apa fungsi dan tujuan dari variable \$activeMenu ?
digunakan untuk menyimpan nilai dari menu navigasi yang sedang aktif, biasanya berisi identifier atau kode unik dari menu navigasi yang sedang dipilih atau dibuka oleh pengguna yang gunanya untuk menandai atau memberi tahu tata letak (layout) halaman web bahwa suatu menu tertentu sedang aktif.