

PEMROGRAMAN WEB LANJUT

JOBSHEET 7



MUHAMMAD IQBAL MAKMUR AL-MUNIRI
2241720099 / 15 / TI-2C
TEKNIK INFORMATIKA
TEKNOLOGI INFORMASI

Praktikum 1 – *Layouting AdminLTE*

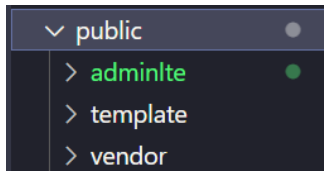
1. Kita download AdminLTE v3.2.0 yang rilis pada 8 Feb 2022

Jawab:

AdminLTE-3.2.0 26/03/2024 20:44 Compressed (zipped)... 28.333 KB

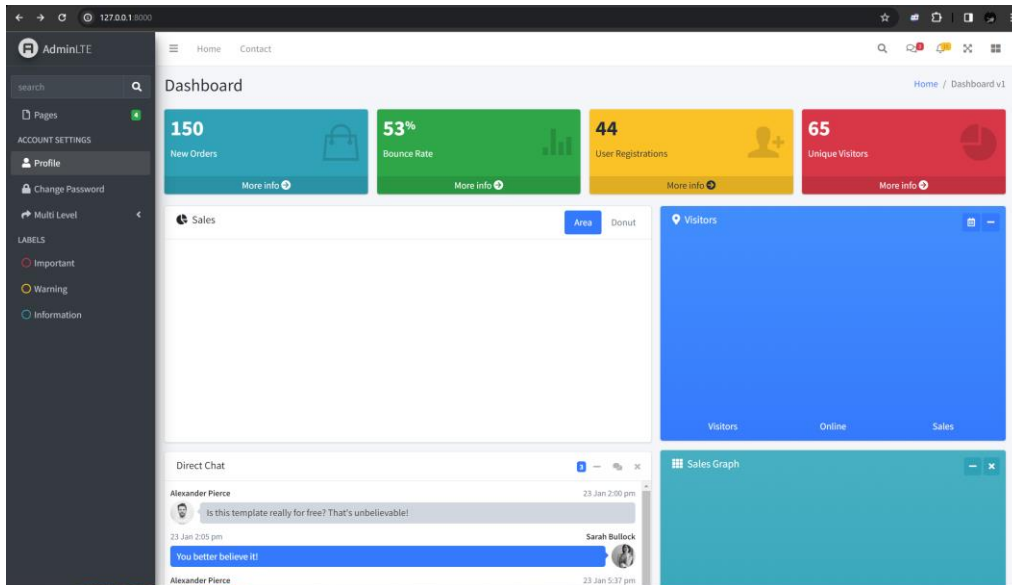
2. Setelah kita berhasil download, kita ekstrak file yang sudah di download ke folder project PWL_POS/public, kemudian kita rename folder cukup menjadi adminlte

Jawab:



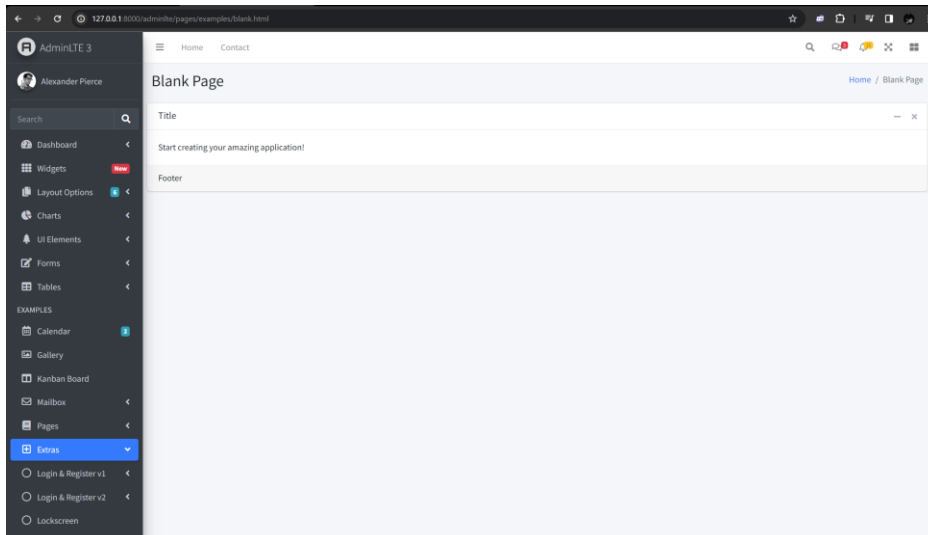
3. Selanjutnya kita buka di browser dengan alamat.

Jawab:

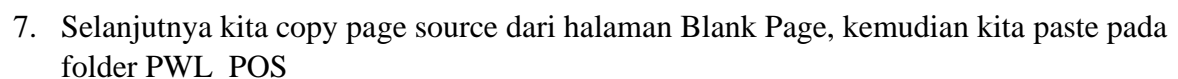


4. Kita klik menu Extras > Blank Page, page inilah yang akan menjadi dasar web template.

Jawab:



- Jawab:**



Jawab:



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

Jawab:

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

10. Kemudian kita blok baris 19-153 (baris untuk element 1-header), lalu kita cut, dan paste-kan di file. Sehingga tampilan dari file template.blade.php menjadi seperti berikut.

Jawab:

```
17 <div class="wrapper">
18 <!-- Navbar -->
19 @include('layouts.header')
20 <!-- /.navbar -->
```

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

Jawab:

```
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>
```

12. Selanjutnya kita blok baris 31-693 (baris untuk element 2-sidebar), lalu kita cut, dan paste-kan di file PWL_POs. Sehingga tampilan dari file template.blade.php menjadi seperti berikut.

Jawab:

```
29
30 <!-- Sidebar -->
31 @include('layouts.sidebar')
32 <!-- /.sidebar -->
33 </aside>
```

13. Selanjutnya perhatikan baris 97-98 (baris untuk element 5-footer), lali kita cut, dan paste-kan di file PWL_POS. Sehingga tampilan dari file template.blade.php menjadi seperti berikut.

Jawab:

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

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

Jawab:

```
91 <!-- jQuery -->
92 <script src="{{ asset('adminlte/plugins/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 element untuk content.

16. Perhatikan file template.blade.php pada baris 38-52 kita nadikan sebagai elemen 4-breadcrumb. Kita blok baris 38-52 lalu kita cut, dan paste-kan di file PWL_POS. Sehingga tampilan dari file template.blade.php menjadi seperti berikut.

Jawab:

```
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 bis akita buat dinamis, sesuai denga napa 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')

Jawab:

```
39
40 <!-- Main content -->
41 <section class="content">
42     @yield('content')
43 </section>
```

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

Jawab:

```
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 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>
```

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

Praktikum 2 – Penerapan Layouting

1. Kita buat file controller dengan nama WelcomeController.php

Jawab:

```
app > Http > Controllers > WelcomeController.php > PHP Intelephense > WelcomeController
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
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' => $activeMe
19     }
20 }
21
```

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

Jawab:

```
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 smeua, ini adalah halaman utama dari aplikasi ini.
12  </div>
13 </div>
14 @endsection
```

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

Jawab:

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

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

Jawab:

```
resources > views > layouts > sidebar.blade.php > ...
1 <div class="sidebar">
2   <div class="form-group">
3     <input type="text" value="Search" />
4     <button type="button" value="Search" />
5   </div>
6   <div class="nav">
7     <div class="nav-item">
8       <a href="#">Dashboard</a>
9     </div>
10    <div class="nav-item">
11      <a href="#">Data Pengguna</a>
12    </div>
13    <div class="nav-item">
14      <a href="#">Data Barang</a>
15    </div>
16    <div class="nav-item">
17      <a href="#">Kategori</a>
18    </div>
19  </div>
20 </div>
```

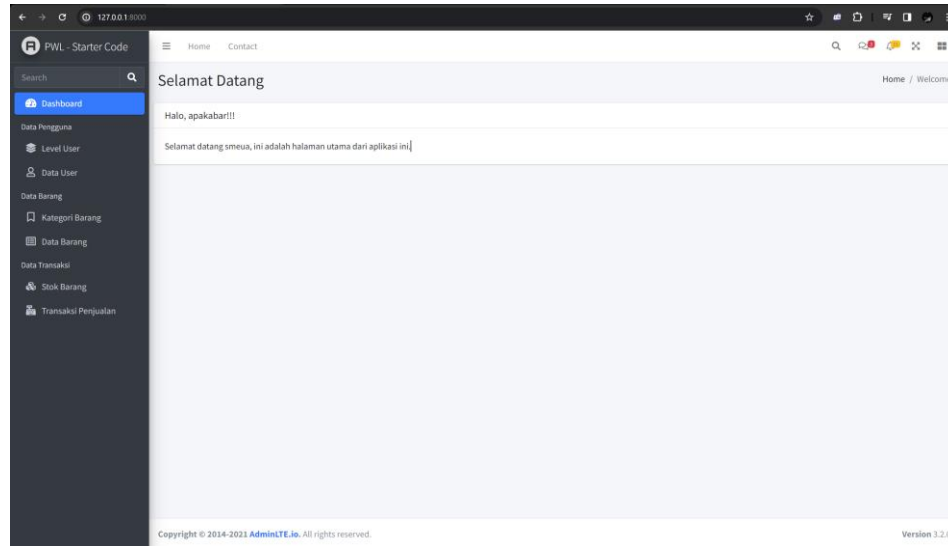
5. Kita tambahkan kode berikut router web.php

Jawab:

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

6. Sekarang kita coba jalankan di browser dengan mengetikkan url

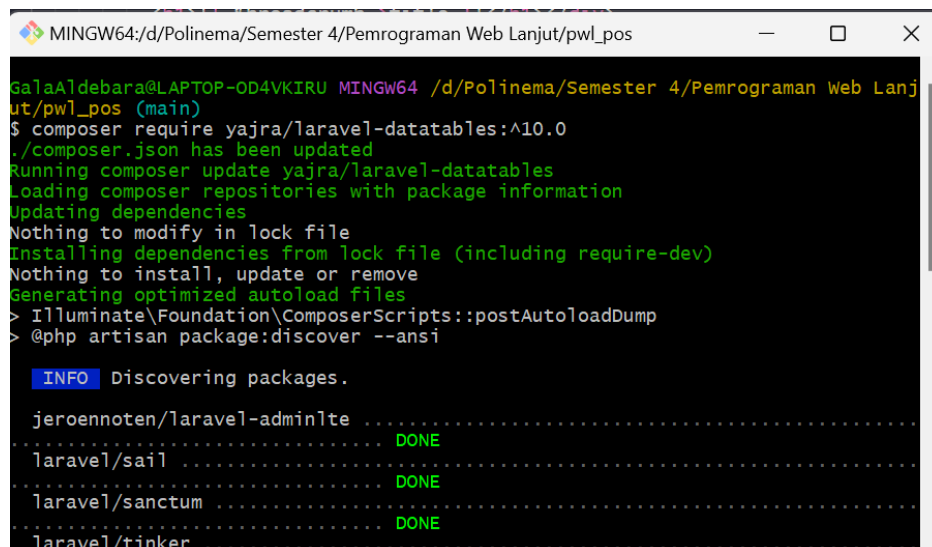
Jawab:



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

Jawab:



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

Jawab:

```
62 Route::get('/', [UserController::class, 'index']);
63
64 Route::group(['prefix' => 'user'], function () {
65     Route::get('/', [UserController::class, 'index']);
66     Route::post('/list', [UserController::class, 'list']);
67     Route::get('/create', [UserController::class, 'create']);
68     Route::get('/', [UserController::class, 'store']);
69     Route::get('/{id}', [UserController::class, 'show']);
70     Route::get('/{id}/edit', [UserController::class, 'edit']);
71     Route::get('/{id}', [UserController::class, 'update']);
72     Route::get('/{id}', [UserController::class, 'destroy']);
73 });
```

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

Jawab:

```
75 public function index()
76 {
77     $breadcrumb = (object) [
78         'title' => 'Daftar User',
79         'list' => ['Home', 'User']
80     ];
81
82     $page = (object) [
83         'title' => 'Daftar user terdaftar dalam sistem'
84     ];
85
86     $activeMenu = 'user';
87     return view('user.index', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
88 }
89 }
```

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

Jawab:

```
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 href="#" class="btn btn-sm btn-primary mt-1" href="{{ url('user/create') }}">Tambah</a>
9             </div>
10         </div>
11         <div class="card-body">
12             <table class="table table-bordered table-striped table-hover table-sm">
13                 <thead>
14                     <tr>
15                         <th>ID</th>
16                         <th>Username</th>
17                         <th>Nama</th>
18                         <th>Level</th>
19                     </tr>
20                 </thead>
21                 <tbody>
22                     <tr>
23                         <td>{{ $table->first()->id }}</td>
24                         <td>{{ $table->first()->username }}</td>
25                         <td>{{ $table->first()->nama }}</td>
26                         <td>{{ $table->first()->level }}</td>
27                     </tr>
28                 </tbody>
29             </table>
30         </div>
31     </div>
32
33 @push('css')
34 @endpush
35
36 @push('js')
37 <script>
38     $(document).ready(function() {
39         var dataTable = $('#table_user').DataTable({
40             serverSide: true, // server-side: true, jika login menggunakan server
41             sideProcessing: true, // sideProcessing: true, jika login menggunakan server
42             ajax: {
43                 url: "{{ url('user/list') }}",
44                 dataType: "json",
45                 type: "POST"
46             },
47             columns: [
48                 {
49                     data: "DT_RowIndex", // nomor urut dari laravel datatable
50                     className: "text-center",
51                     orderable: false,
52                     searchable: false
53                 },
54                 {
55                     data: "username",
56                     className: "",
57                     orderable: true, // orderable: true, jika login belum ter login
58                 }
59             ]
60         });
61     });
62 </script>
63 @endpush
64
65 @endsection
```

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

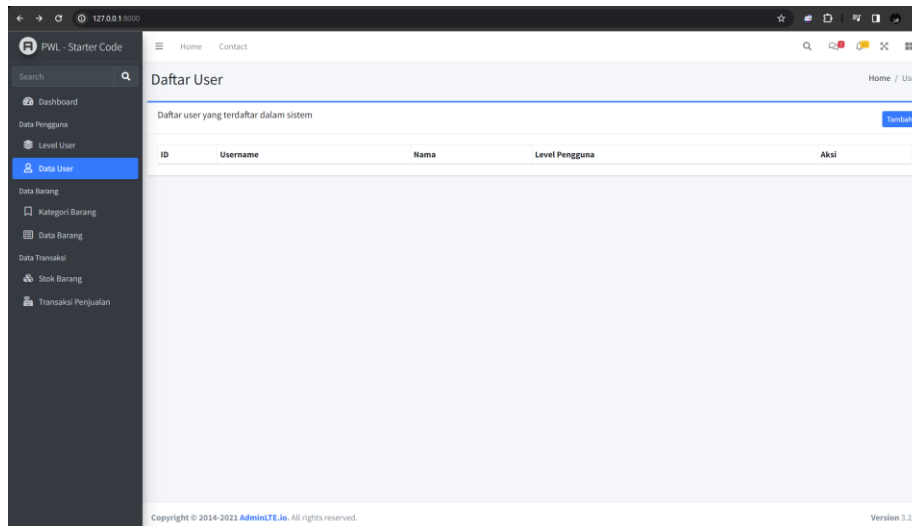
```
1 <!DOCTYPE html>
2 <html Lang="en">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <title>{{ config('app.name', 'PWL Laravel Starter Code') }}</title>
7
8 <meta name="csrf-token" content="{{ csrf_token() }}">
9
10 <!-- Google Font: Source 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/cs/responsive.bootstrap4.min.css') }}">
17 <link rel="stylesheet" href="{{ asset('adminlte/plugins/datatables-buttons/cs/buttons.bootstrap4.min.css') }}">
18 <!-- Theme style -->
19 <link rel="stylesheet" href="{{ asset('adminlte/dist/css/adminlte.min.css') }}">
20
21 @stack('css')
22 </head>
```

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

```
91 public function list(Request $request)
92 {
93     $users = UserModel::select('user_id', 'username', 'nama', 'level_id')
94         ->with('level');
95
96     return DataTables::of($users)
97         ->addIndexColumn() // menambahkan kolom index / no urut (default nama kolom: DT_RowIndex)
98         ->addColumn('aksi', function ($user) { // menambahkan kolom aksi
99             $btn = 'a href="" . url("/user/" . $user->user_id) . "" class="btn btn-info btn-sm">Detail</a> ';
100             $btn = 'a href="" . url("/user/" . $user->user_id) . "/edit" . "" class="btn btn-warning btn-sm">Edit</a> ';
101             $btn = 'form class="d-inline-block" method="POST" action="" . url("/user/" . $user->user_id) . "">
102                 csrf_field() . method_field("DELETE") .
103                 <button type="submit" class="btn btn-danger btn-sm" onclick="return confirm('Apakah Anda yakin menghapus data ini?');">Hapus</button></form>;
104             return $btn;
105         })
106         ->rawColumns(['aksi']) // memberitahu bahwa kolom aksi adalah html
107         ->make(true);
108 }
109 }
110 }
```

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

Jawab:



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

Jawab:

```
112 public function create()
113 {
114     $breadcrumb = (object) [
115         'title' => 'Tambah User',
116         'list' => ['Home', 'User', 'Tambah']
117     ];
118
119     $page = (object) [
120         'title' => 'Tambah user baru'
121     ];
122
123     $level = LevelModel::all();
124     $activeMenu = 'user';
125
126     return view('user.create', ['breadcrumb' => $breadcrumb, 'page' => $page, 'level' => $level, 'activeMenu' => $activeMenu]);
127 }
128
129 }
```

10. Sekarang kita buat form untuk menambah data, kita buat file

PWL_POS/resources/views/user/create.blade.php

Jawab:

```
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">{{ $page->title }}</h3>
7         </div>
8         <div class="card-body">
9             <form method="POST" action="{{ url('user') }}" class="form-horizontal">
10                 @csrf
11                 <div class="form-group row">
12                     <div class="col-1 control-label col-form-label">level</div>
13                     <div class="col-11">
14                         <select class="form-control" id="level_id" name="level_id" required>
15                             <option value="">Pilih level </option>
16                             @foreach($level as $item)
17                                 <option value="{{ $item->level_id }}">{{ $item->level_nama }}</option>
18                             @endforeach
19                         </select>
20                         @error('level_id')
21                             <small class="form-text text-danger">{{ $message }}</small>
22                         @enderror
23                     </div>
24                 </div>
25                 <div class="form-group row">
26                     <div class="col-1 control-label col-form-label">username</div>
27                     <div class="col-11">
28                         <input type="text" class="form-control" id="username" name="username" value="{{ old('username') }}" required>
29                         @error('username')
30                             <small class="form-text text-danger">{{ $message }}</small>
31                         @enderror
32                     </div>
33                 </div>
34                 <div class="form-group row">
35                     <div class="col-1 control-label col-form-label">nama</div>
36                     <div class="col-11">
37                         <input type="text" class="form-control" id="nama" name="nama" value="{{ old('nama') }}" required>
38                         @error('nama')
39                             <small class="form-text text-danger">{{ $message }}</small>
40                         @enderror
41                     </div>
42                 </div>
43                 <div class="form-group row">
44                     <div class="col-1 control-label col-form-label">password</div>
45                     <div class="col-11">
46                         <input type="password" class="form-control" id="password" name="password" required>
47                     </div>
48                 </div>
49             </form>
50         </div>
51     </div>
52 @endsection
```

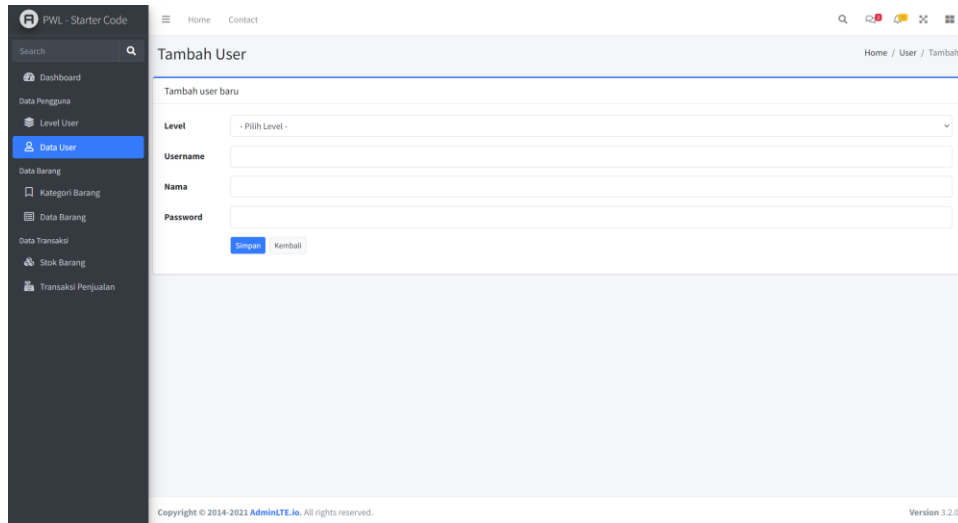
11. Kemudian untuk bisa menghandle data yang akan disimpan ke database, kita buat fungsi store() di UserController.php

Jawab:

```
129 public function store(Request $request)
130 {
131     $request->validate([
132         'username' => 'required|string|min:3|unique:m_user,username',
133         'nama' => 'required|string|max:100',
134         'password' => 'required|min:5',
135         'level_id' => 'required|integer'
136     ]);
137
138     UserModel::create([
139         'username' => $request->username,
140         'nama' => $request->nama,
141         'password' => bcrypt($request->password),
142         'level_id' => $request->level_id
143     ]);
144
145     return redirect('/user')->with('success', 'Data user berhasil disimpan');
146 }
147 }
```

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

Jawab:



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

Jawab:

```
148 public function show(string $id)
149 {
150     $user = UserModel::with('level')->find($id);
151
152     $breadcrumb = (object) [
153         'title' => 'Detail User',
154         'list' => ['Home', 'User', 'Detail']
155     ];
156
157     $page = (object) [
158         'title' => 'Detail User'
159     ];
160
161     $activeMenu = 'user';
162
163     return view('user.show', ['breadcrumb' => $breadcrumb, 'page' => $page, 'user' => $user, 'activeMenu' => $activeMenu]);
164 }
165 }
```

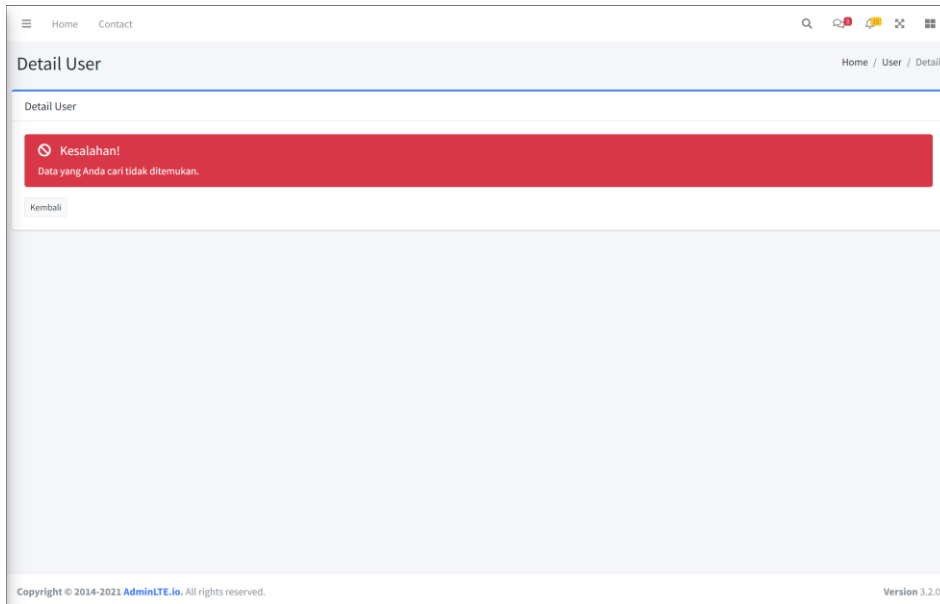
15. Kemudian kita buat view di PWL_POS/resources/views/user/show.blade.php

Jawab:

```
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 <h4 class="icon fas fa-ban">{{ $page->title }}</h4>
13 Data yang Anda cari tidak ditemukan.
14 </div>
15 @else
16 <table class="table table-bordered table-striped table-hover table-sm">
17 <tr>
18 <th>ID</th>
19 <td>{{ $user->user_id }}</td>
20 </tr>
21 <tr>
22 <th>Level</th>
23 <td>{{ $user->level->level_nama }}</td>
24 </tr>
25 <tr>
26 <th>Username</th>
27 <td>{{ $user->username }}</td>
28 </tr>
29 <tr>
30 <th>Name</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 @push('css')
44 @endpush
45 @push('js')
46 @endpush
47 @endpush
```

16. Sekarang kalian coba untuk melihat detail data user di browser, dan coba untuk mengetikkan id yang salah.

Jawab:



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

Jawab:

```
166 | public function edit(string $id)
167 | {
168 |     $user = UserModel::find($id);
169 |     $level = LevelModel::all();
170 |
171 |     $breadcrumb = (object) [
172 |         'title' => 'Edit User',
173 |         'list' => ['Home', 'User', 'Edit']
174 |     ];
175 |
176 |     $page = (object) [
177 |         'title' => 'Edit User'
178 |     ];
179 |
180 |     $activeMenu = 'user';
181 |
182 |     return view('user.edit', ['breadcrumb' => $breadcrumb, 'page' => $page, 'user' => $user, 'level' => $level, 'activeMenu' => $activeMenu]);
183 | }
184 |
185 | public function update(Request $request, string $id)
186 | {
187 |     $request->validate([
188 |         'username' => 'required|string|min:3|unique:m_user,username',
189 |         'nama' => 'required|string|max:100',
190 |         'password' => 'required|min:5',
191 |         'level_id' => 'required|integer'
192 |     ]);
193 |
194 |     UserModel::find($id)->update([
195 |         'username' => $request->username,
196 |         'nama' => $request->nama,
197 |         'password' => $request->password ? bcrypt($request->password) : UserModel::find($id)->password,
198 |         'level_id' => $request->level_id
199 |     ]);
200 |
201 |     return redirect('/user')->with('success', 'Data berhasil diubah');
202 | }
203 | }
```

Jawab:

Jawab:

Jadi kita buat fungsi destroy() pada UserController.php

Jawab:

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

Jawab:

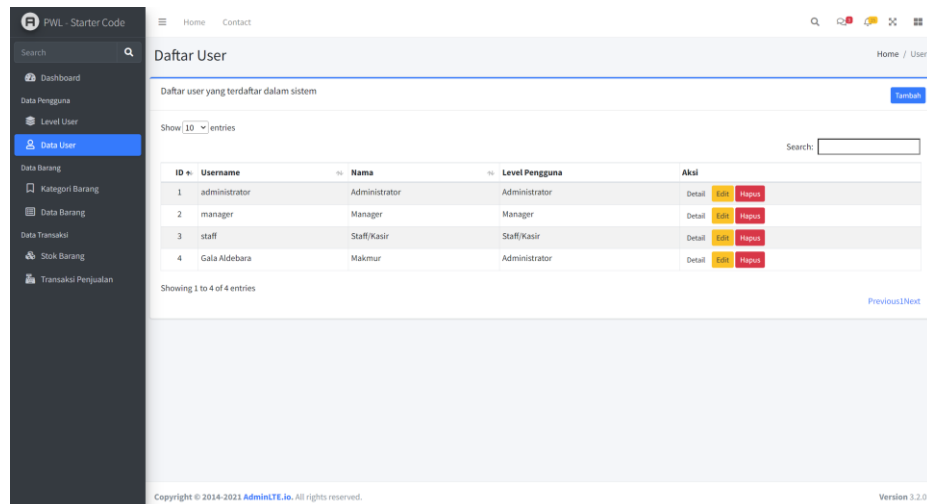
```

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') }}
17     @endif
18     <table class="table table-bordered table-striped table-hover table-sm" id="table_user">
19         <thead>
20             <tr><th>ID</th><th>Username</th><th>Nama</th><th>Level Pengguna</th><th>Aksi</th></tr>
21         </thead>
22     </table>
23 </div>
24 </div>
25 @endsection

```

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

Jawab:



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

Jawab:

```

76 public function index()
77 {
78     $breadcrumb = (object) [
79         'title' => 'Daftar User',
80         'list' => ['Home', 'User']
81     ];
82
83     $page = (object) [
84         'title' => 'Daftar user yang terdaftar dalam sistem'
85     ];
86
87     $activeMenu = 'user';
88     $level = LevelModel::all();
89     return view('user.index', ['breadcrumb' => $breadcrumb, 'page' => $page, 'level' => $level, 'activeMenu' => $activeMenu]);
90 }

```

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

Jawab:

```
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 class="form-control" id="level_id" name="level_id" 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

Jawab:

```
50 ajax: {
51     "url": "{ url('user/list') }",
52     "dataType": "json",
53     "type": "POST"
54     "data": function (d) {
55         d.level_id = $('#level_id').val();
56     }
57 },
```

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

Jawab:

```
88 $('#level_id').on('change', function() {
89     dataUser.ajax.reload();
90 });
91 });
```

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

Jawab:

```
98 if ($request->level_id) {
99     $users->where('level_id', $request->level_id);
100 }
```


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

Jawab:

Daftar user yang terdaftar dalam sistem Tambah

Filter:

- Semua -

Level Pengguna

Show

10

 entries

Search:

Adm

ID	Username	Nama	Level Pengguna	Aksi
1	administrator	Administrator	Administrator	<div>Detail</div> <div>Edit</div> <div>Hapus</div>

Showing 1 to 1 of 1 entries (filtered from 4 total entries)

[Previous](#)[Next](#)

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

Pertanyaan

Jawablah pertanyaan berikut sesuai pemahaman materi di atas

1. Apa perbedaan frontend template dengan backend template?
2. Apakah layouting itu penting dalam membangun sebuah website?
3. Jelaskan fungsi dari komponen Laravel blade berikut:
4. Apa fungsi dan tujuan dari variable \$activeMenu?

TUGAS

Implementasikan menu yang belum ada di Laravel starter code ini sesuai dengan Studi Kasus Point of Sales Sederhana. Silahkan terapkan kode di Laravel starter code untuk menu-menu yang sesuai dengan menu di samping ini.

Level User

Daftar Level

Daftar level yang terdaftar dalam sistem

Filter: - Semua -
Level Pengguna

Show 10 entries

Search:

ID	level kode	level Nama	Aksi
1	ADM	Administrator	Detail Edit Hapus
2	MNG	Manager	Detail Edit Hapus
3	STF	Staff/Kasir	Detail Edit Hapus

Showing 1 to 3 of 3 entries

Previous1Next

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

Kategori Barang

Daftar Kategori

Daftar kategori yang terdaftar dalam sistem

Filter: - Semua -
Kategori Pengguna

Show 10 entries

Search:

ID	Kategori kode	Kategori Nama	Aksi
1	Lenovo	laptop	Detail Edit Hapus
2	Redmi	HP	Detail Edit Hapus
3	Toshiba	Televisi	Detail Edit Hapus
4	Samsung	Air Purifier	Detail Edit Hapus
5	Ferrari	Mobil	Detail Edit Hapus
6	Apple	iWatch	Detail Edit Hapus

Showing 1 to 6 of 6 entries

Previous1Next

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

Data Barang

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 Barang

Home / Barang

Daftar barang yang terdaftar dalam sistem

Tambah

Filter: - Semua -
Kategori Barang

Show 10 entries

Search:

ID	Barang Kode	Barang Nama	Harga Beli	Harga Jual	Kategori	Aksi
1	Lenovo	Lenovo Legion 5 Pro	20000000	15000000	laptop	Detail Edit Hapus
2	HP	HP Victus 16	20000000	15000000	laptop	Detail Edit Hapus
3	Apple	Iphone 15 Promax	20000000	15000000	HP	Detail Edit Hapus
4	Redmi	Redmi Note 13	20000000	15000000	HP	Detail Edit Hapus
5	Toshiba	Canvio Basic 1TB	20000000	15000000	Televisi	Detail Edit Hapus
6	LG	Mesin Cuci	20000000	15000000	Televisi	Detail Edit Hapus
7	Samsung	Galaxy S23 Ultra	20000000	15000000	Air Purifier	Detail Edit Hapus
8	Poco	Poco X6 Pro	20000000	15000000	Air Purifier	Detail Edit Hapus
9	Honda	Civic Turbo	20000000	15000000	Mobil	Detail Edit Hapus
10	Lambo	Aventador	20000000	15000000	Mobil	Detail Edit Hapus

Showing 1 to 10 of 10 entries

PreviousNext

Stok Barang

PWL - Starter Code

Search

Dashboard

Data Pengguna

Level User

Data User

Data Barang

Kategori Barang

Data Barang

Stok Barang

Transaksi Penjualan

Home

Contact

Daftar Stok

Home / Stok

Daftar stok yang terdaftar dalam sistem

Tambah

Show 10 entries

Search:

ID	Barang	User	Stok Tanggal	Stok Jumlah	Aksi
1	Lenovo Legion 5 Pro	administrator	2005-01-17 00:00:00	10	Detail Edit Hapus
2	HP Victus 16	administrator	2005-01-17 00:00:00	10	Detail Edit Hapus
3	Iphone 15 Promax	administrator	2005-01-17 00:00:00	10	Detail Edit Hapus
4	Redmi Note 13	administrator	2005-01-17 00:00:00	10	Detail Edit Hapus
5	Canvio Basic 1TB	manager	2005-01-17 00:00:00	10	Detail Edit Hapus
6	Mesin Cuci	manager	2005-01-17 00:00:00	10	Detail Edit Hapus
7	Galaxy S23 Ultra	manager	2005-01-17 00:00:00	10	Detail Edit Hapus
8	Poco X6 Pro	staff	2005-01-17 00:00:00	10	Detail Edit Hapus
9	Civic Turbo	staff	2005-01-17 00:00:00	10	Detail Edit Hapus
10	Aventador	staff	2005-01-17 00:00:00	10	Detail Edit Hapus

Showing 1 to 10 of 10 entries

PreviousNext

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

Transaksi Penjualan

PWL - Starter Code

Search

Dashboard

Data Pengguna

Level User

Data User

Data Barang

Kategori Barang

Data Barang

Data Transaksi

Stok Barang

Transaksi Penjualan

HomeContact

Daftar Penjualan

Home / Penjualan

Daftar penjualan yang terdaftar dalam sistem

Tambah

Filter: - Semua -

Nama User

Show 10 entries

Search:

ID	User	Pembeli	Penjualan kode	Penjualan Tanggal	Aksi
1	staff	Arya Bagus	KODE1	2005-01-17 00:00:00	<div>DetailEditHapus</div>
2	staff	Yusriyah Firjatullah	KODE2	2005-01-17 00:00:00	<div>DetailEditHapus</div>
3	staff	Dido Imam	KODE3	2005-01-17 00:00:00	<div>DetailEditHapus</div>
4	manager	Atha	KODE4	2005-01-17 00:00:00	<div>DetailEditHapus</div>
5	manager	Naufal	KODE5	2005-01-17 00:00:00	<div>DetailEditHapus</div>
6	manager	Iqbal	KODE6	2005-01-17 00:00:00	<div>DetailEditHapus</div>
7	administrator	Makmur	KODE7	2005-01-17 00:00:00	<div>DetailEditHapus</div>
8	staff	Royhan	KODE8	2005-01-17 00:00:00	<div>DetailEditHapus</div>
9	staff	Chevalerie	KODE9	2005-01-17 00:00:00	<div>DetailEditHapus</div>
10	staff	Akhyar	KODE10	2005-01-17 00:00:00	<div>DetailEditHapus</div>

Showing 1 to 10 of 10 entries

PreviousNext