

LAPORAN
HandsOut(Jobsheet) - Pemrograman Web – LARAVEL STARTER
CODE

Fendi Zulkarnain



VSGA X TSA
POLITEKNIK NEGERI MALANG KAMPUS LUMAJANG
Jl. Lintas Timur, Area Sawah/Kebun, Jogotrunan, Kec. Lumajang, Kabupaten Lumajang,
Jawa Timur 67314
2024

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

```
W I N D O W S  X@DESKTOP-6E8EN92 MINGW64 /c:/laragon/www/VSGA_POS (main)
$ composer require yajra/laravel-datatables:^10.0
```

atau

composer require yajra/laravel-datatables-oracle

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

```
20 Route::group(['prefix' => 'user'], function() {
21     Route::get('/', [UserController::class, 'index']);
22     Route::post('/list', [UserController::class, 'list']);
23     Route::get('/create', [UserController::class, 'create']);
24     Route::post('/', [UserController::class, 'store']);
25     Route::get('/{id}', [UserController::class, 'show']);
26     Route::get('/{id}/edit', [UserController::class, 'edit']);
27     Route::put('/{id}', [UserController::class, 'update']);
28     Route::delete('/{id}', [UserController::class, 'destroy']);
29 });
```

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

```
13 public function index()
14 {
15     $breadcrumb = (object) [
16         'title' => 'Daftar User',
17         'list' => ['Home', 'User']
18     ];
19     $page = (object) [
20         'title' => 'Daftar user yang terdaftar dalam sistem'
21     ];
22     return view('user.index', ['breadcrumb' => $breadcrumb, 'page' => $page]);
23 }
```

5. Lalu kita buat view pada 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"><a href="{{ url('user/create') }}" class="btn btn-sm btn-primary mt-1"></a></div>
8         </div>
9         <div class="card-body">
10             <table class="table table-bordered table-striped table-hover table-sm" id="table_user">
11                 <thead>
12                     <tr>
13                         <th>ID</th>
14                         <th>Username</th>
15                         <th>Nama</th>
16                         <th>Level Pengguna</th>
17                         <th>Aksi</th>
18                     </tr>
19                 </thead>
20             </table>
21         </div>
22     </div>
23 @endsection
24
25 @push('css')
26 @endpush
```

```

28 @push('js')
29     <script>
30         $(document).ready(function() {
31             var dataUser = $('#table_user').DataTable({
32                 serverSide: true,
33                 ajax: {
34                     "url": "{{ url('user/list') }}",
35                     "dataType": "json",
36                     "type": "POST"
37                 },
38                 column: [
39                     {
40                         data: "DT_RowIndex",
41                         className: "text-center",
42                         orderable: false,
43                         searchable: false
44                     },
45                     {
46                         data: "username",
47                         className: "",
48                         orderable: true,
49                         searchable: true
50                     },
51                     {
52                         data: "nama",
53                         className: "",
54                         orderable: true,
55                         searchable: true
56                     },
57                     {
58                         data: "level.level_nama",
59                         className: "",
60                         orderable: false,
61                         searchable: false
62                     },
63                     {
64                         data: "aksi",
65                         className: "",
66                         orderable: false,
67                         searchable: false
68                     },
69                 ]
70             });
71         });
72     </script>
73 @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

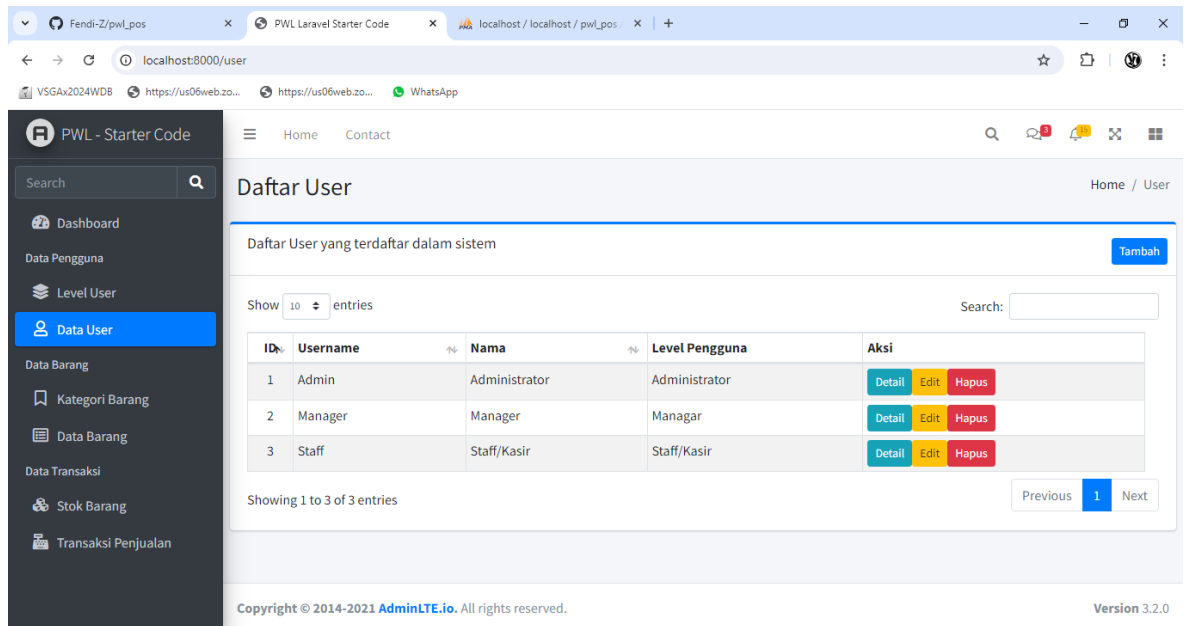
```
resources > views > layouts > template.blade.php > html > head
1 <!DOCTYPE html>
2 <html Lang="en">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <title>{{ config('app.name', 'PWL Laravel Starter Code') }}</title>
7
8 <meta name="csrf-token" 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/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')
22 </head>
```

```
60 <!-- jQuery -->
61 <script src="{{ asset('adminlte/plugins/jquery/jquery.min.js') }}"></script>
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/pdfmake/pdfmake.min.js') }}"></script>
72 <script src="{{ asset('adminlte/plugins/pdfmake/vfs_fonts.js') }}"></script>
73 <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.html5.min.js') }}"></script>
74 <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.print.min.js') }}"></script>
75 <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.colVis.min.js') }}"></script>
76 <!-- AdminLTE App -->
77 <script src="{{ asset('adminlte/dist/js/adminlte.min.js') }}"></script>
78 <script>
79 | $.ajaxSetup({headers: {'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')}});
80 </script>
81 @stack('js')
82 </body>
83 </html>
```

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

```
27 public function list(Request $request)
28 {
29     $users = UserModel::select('user_id', 'username', 'nama', 'level_id')->with('level');
30     return DataTables::of($users)->addIndexColumn()->addColumn('aksi', function ($user) {
31         $btn = '<a href="' . url('/user/' . $user->user_id) . '" class="btn btn-info btn-sm">Detail</a>';
32         $btn .= '<a href="' . url('/user/' . $user->user_id . '/edit') . '" class="btn btn-warning btn-sm">Edit</a>';
33         $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
yakini menghapus data ini?');">Hapus</button></form>';
34     }->rawColumn(['aksi']->make(true));
35 }
```

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



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

```

39     public function create()
40     {
41         $breadcrumb = (object) [
42             'title' => 'Daftar User',
43             'list' => ['Home', 'User', 'Tambah']
44         ];
45
46         $page = (object) [
47             'title' => 'Tambah User Baru'
48         ];
49
50         $level = LevelModel::all();
51
52         return view('user.create', ['breadcrumb' => $breadcrumb, 'page' => $page, 'level' => $level]);
53     }

```

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

```

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

```

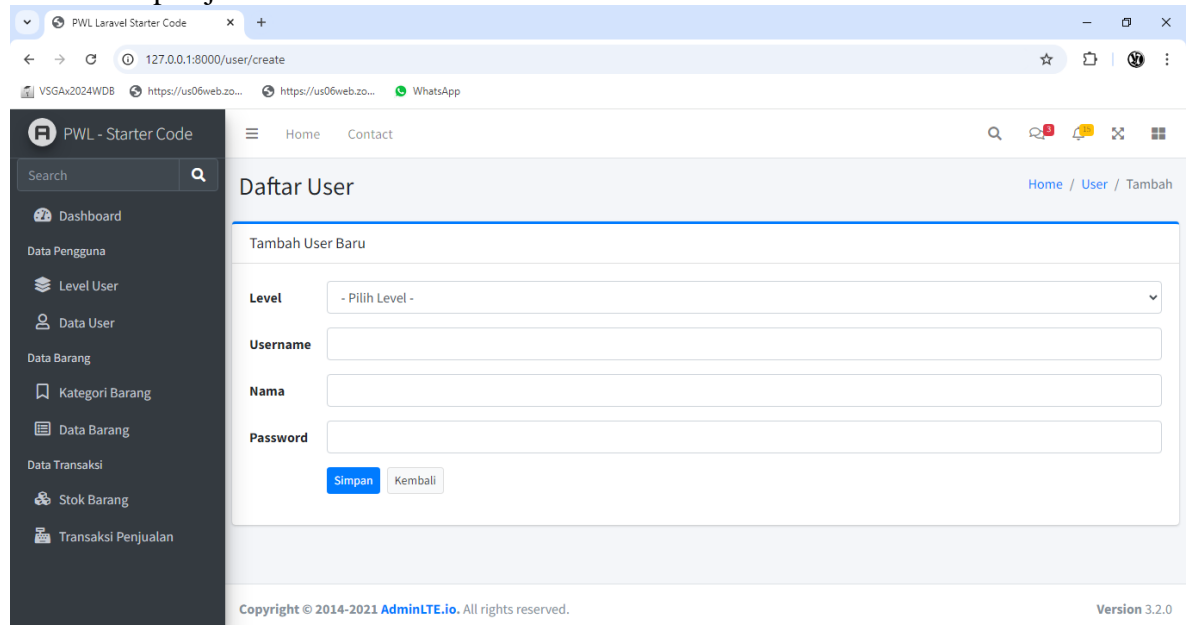
11. Kemudian untuk bisa meng-handle data yang akan disimpan ke database, kita buat fungsi store() di UserController.php

```

55     public function store(Request $request)
56     {
57         $request->validate([
58             'username' => 'required|string|min:3|unique:m_user,username',
59             'nama' => 'required|string|max:100',
60             'password' => 'required|min:5',
61             'level_id' => 'required|integer'
62         ]);
63
64         UserModel::create([
65             'username' => $request->username,
66             'nama' => $request->nama,
67             'password' => bcrypt($request->password),
68             'level_id' => $request->level_id
69         ]);
70
71         return redirect('/user')->with('success', 'Data User berhasil disimpan');
72     }

```

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

```

14     Route::get('/{id}', [UserController::class, 'show']);

```

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

```

74     public function show(string $id)
75     {
76         $user = UserModel::with('level')->find($id);
77
78         $breadcrumb = (object) [
79             'title' => 'Detail User',
80             'list' => ['Home', 'User', 'Detail']
81         ];
82
83         $page = (object) [
84             'title' => 'Detail User'
85         ];
86
87         return view('user.create', ['user' => $user, 'breadcrumb' => $breadcrumb, 'page' => $page]);
88     }

```

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

```

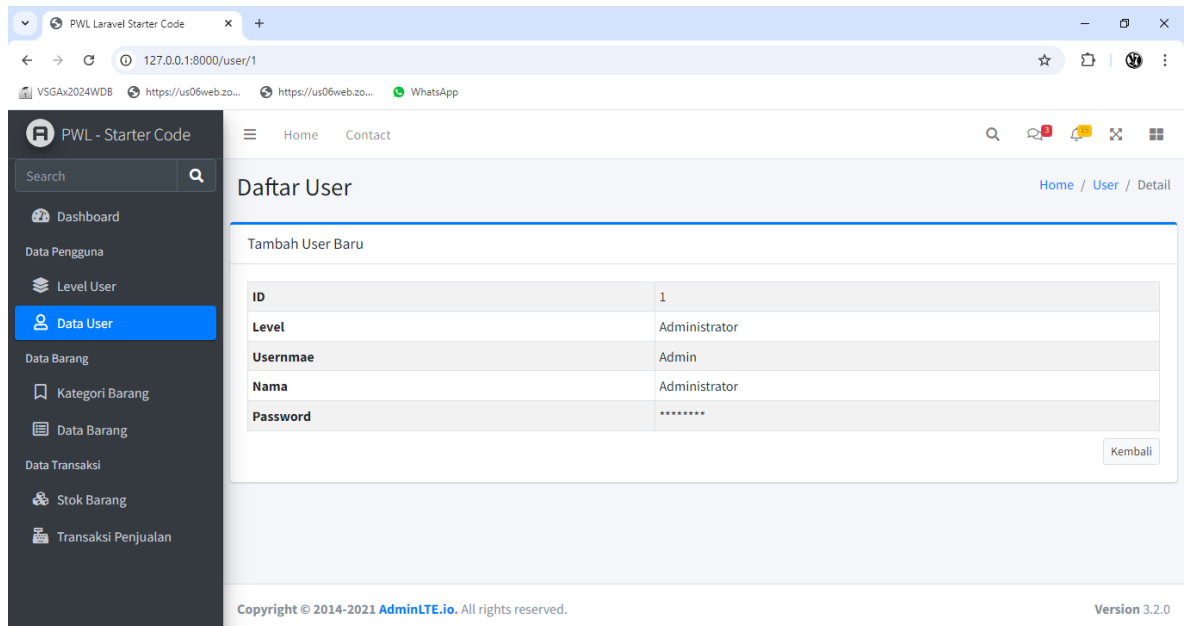
resources > views > user > show.blade.php > ...
1  @extends('layouts.template')
2
3  @section('content')
4      <div class="card card-outline card-primary">
5          <div class="card-header">
6              <h3 class="card-title">{{ $page->title }}</h3>
7              <div class="card-tools"></div>
8          </div>
9          <div class="card-body">
10             @empty($user)
11                 <div class="alert alert-danger alert-dismissible">
12                     <h5><i class="icon fas fa-ban">Kesalahan!!!</i></h5>
13                     Data yang Snda 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>Usernama</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="{{ ('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/VSGA_POS/public/user/1

amati apa yang terjadi, dan laporkan!!!



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

```
15 Route::get('/{id}/edit', [UserController::class, 'edit']);
16 Route::put('/{id}', [UserController::class, 'update']);
```

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

```
91 public function edit(string $id)
92 {
93     $user = UserModel::find($id);
94     $level = LevelModel::all();
95
96     $breadcrumb = (object) [
97         'title' => 'Edit User',
98         'list' => ['Home', 'User', 'Edit']
99     ];
100
101     $page = (object) [
102         'title' => 'Edit User'
103     ];
104
105     return view('user.edit', ['breadcrumb' => $breadcrumb, 'page' => $page, 'user' => $user, 'level' => $level]);
106 }
107
108 public function update(Request $request, string $id)
109 {
110     $request->validate([
111         'username' => 'required|string|min:3|unique:m_user,username,' . $id . ',user_id',
112         'nama' => 'required|string|max:100',
113         'password' => 'nullable|min:5',
114         'level_id' => 'required|integer'
115     ]);
116
117     UserModel::find($id)->update([
118         'username' => $request->username,
119         'nama' => $request->nama,
120         'password' => $request->password,
121         'level_id' => $request->level_id,
122     ]);
123
124     return redirect('/user')->with('success', "Data User berhasil diubah");
125 }
```

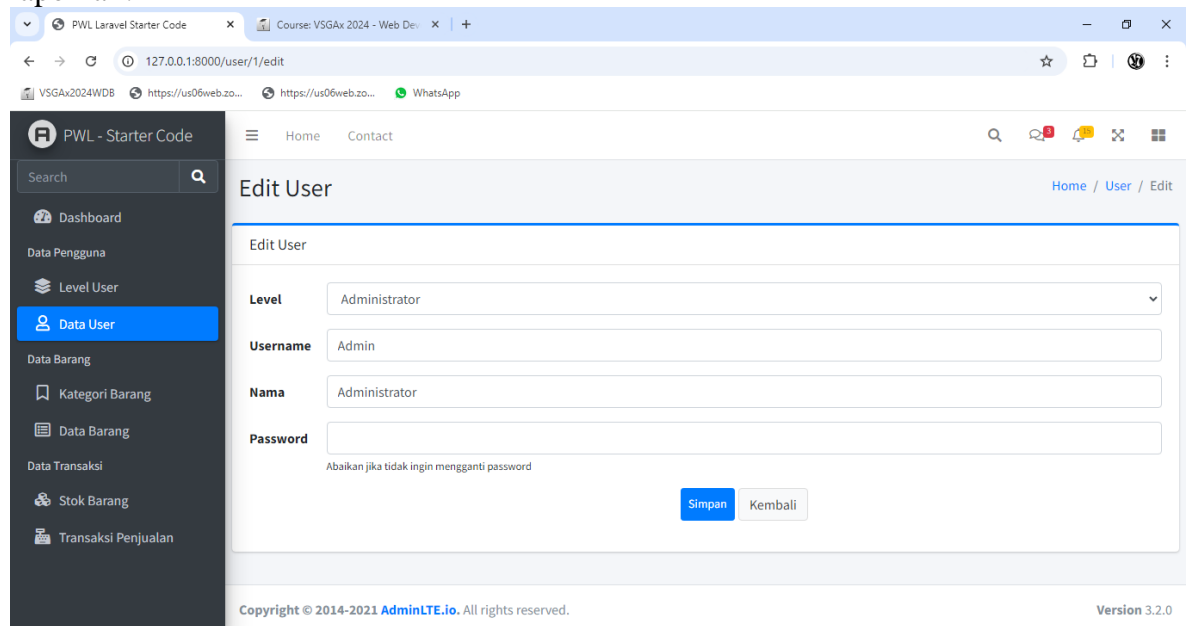
19. Selanjutnya, kita buat view untuk melakukan proses edit data user di resources/views/user/edit.blade.php

```

resources > views > user > edit.blade.php > div.card.card-outline.card-primary > div.card-body > form.form-horizontal > div.form-group.row > div.col-11 > small
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 action="{{ url('/user/' . $user->user_id) }}" method="post" class="form-horizontal">
18                     @csrf
19                     {!! method_field('PUT') !!}
20                     <div class="form-group row">
21                         <label for="level_id" class="col-1 control-label col-form-label">Level</label>
22                         <div class="col-11">
23                             <select name="level_id" id="level_id" class="form-control">
24                                 <option value="">-- Pilih Level --</option>
25                                 @foreach ($level as $item)
26                                     <option value="{{ $item->level_id }}" @if ($item->level_id == $user->level_id) selected @endif>
27                                         {{ $item->level_nama }}
28                                     </option>
29                                 @endforeach
30                             </select>
31                             @error('level_id')
32                                 <small class="form-text text-danger">{{ $message }}</small>
33                             @enderror
34                         </div>
35                     </div>
36                     <div class="form-group row">
37                         <label for="username" class="col-1 control-label col-form-label">Username</label>
38                         <div class="col-11">
39                             <input type="text" name="username" id="username" class="form-control" value="{{ old('username',
40                                 $user->username) }}" required>
41                             @error('username')
42                                 <small class="form-text text-danger">{{ $message }}</small>
43                             @enderror
44                         </div>
45                     </div>
46                     <div class="form-group row">
47                         <label for="nama" class="col-1 control-label col-form-label">Nama</label>
48                         <div class="col-11">
49                             <input type="text" name="nama" id="nama" class="form-control" value="{{ old('nama', $user->nama) }}"
50                                 required>
51                             @error('nama')
52                                 <small class="form-text text-danger">{{ $message }}</small>
53                             @enderror
54                         </div>
55                     </div>
56                     <div class="form-group row">
57                         <label for="password" class="col-1 control-label col-form-label">Password</label>
58                         <div class="col-11">
59                             <input type="text" name="password" id="password" class="form-control">
60                             @error('password')
61                                 <small class="form-text text-danger">{{ $message }}</small>
62                             @else
63                                 <small class="form-text text-mute">Abaikan jika tidak ingin mengganti password</small>
64                             @enderror
65                         </div>
66                     </div>
67                     <div class="form-group row">
68                         <label for="password" class="col-1 control-label col-form-label"></label>
69                         <div class="col-11 d-flex justify-content-center">
70                             <button type="submit" class="btn btn-primary btn-sm">Simpan</button>
71                             <a href="{{ url('user') }}" class="btn btn-default ml-1">Kembali</a>
72                         </div>
73                     </div>
74                 </form>
75             @endempty
76         </div>
77     @endsection
78
79 @push('css')
80 @endpush
81
82 @push('js')
83 @endpush

```

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



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

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

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

```
127 public function destroy(string $id)
128 {
129     $check = UserModel::find($id);
130     if (!$check) {
131         return redirect('/user')->with('error', 'Data User tidak ditemukan');
132     }
133
134     try {
135         UserModel::destroy($id);
136         return redirect('/user')->with('success', 'Data User berhasil dihapus');
137     } catch (\Illuminate\Database\QueryException $e) {
138         return redirect('/user')->with('error', 'Data User gagal dihapus karena masih terdapat tabel lain yang terkait dengan data ini');
139     }
140 }
```

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

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

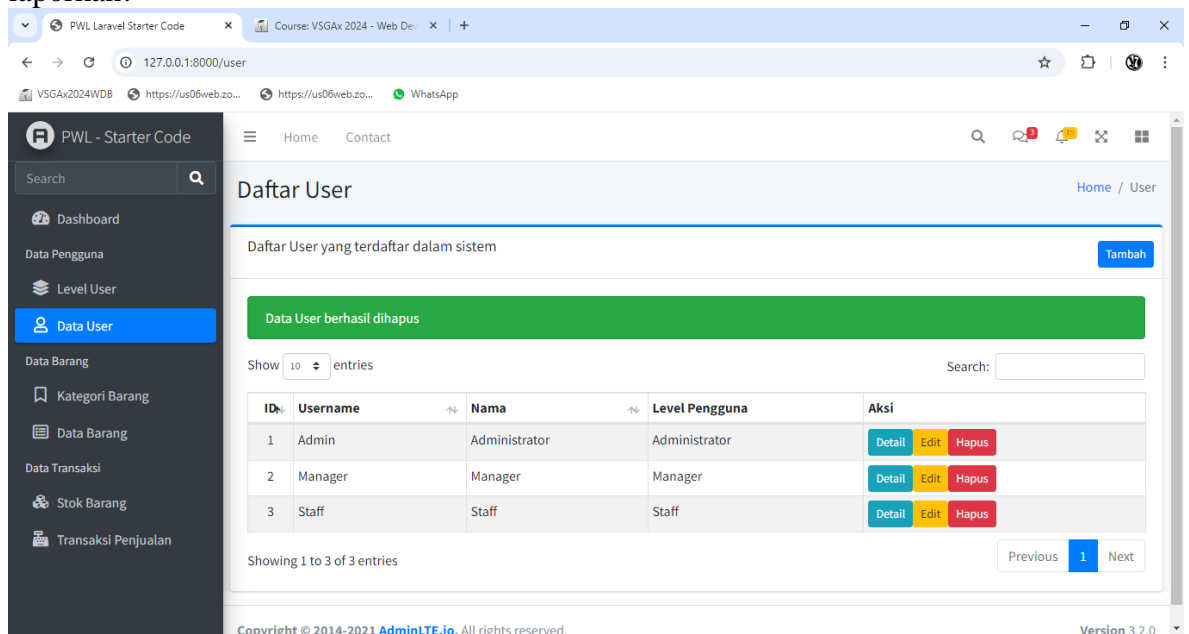
Menjadi

```

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

```

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



Praktikum 2 : Implementasi Filtering Datatables

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

```

12     public function index()
13     {
14         $breadcrumb = (object) [
15             'title' => 'Daftar User',
16             'list' => ['Home', 'User']
17         ];
18
19         $page = (object) [
20             'title' => 'Daftar User yang terdaftar dalam sistem'
21         ];
22
23         $level = LevelModel::all();
24
25         return view('user.index', ['breadcrumb' => $breadcrumb, 'page' => $page, 'level' => $level]);
26     }

```

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

```

13         @if (session('error'))
14             <div class="alert alert-danger">{{ session('error') }}</div>
15         @endif
16         <div class="row">
17             <div class="col-md-12">
18                 <div class="form-group row">
19                     <label class="col-1 control-label col-form-label">Filter : </label>
20                     <div class="col-3">
21                         <select name="level_id" id="level_id" class="form-control" required>
22                             <option value="">-- Semua --</option>
23                             @foreach ($level as $item)
24                                 <option value="{{ $item->level_id }}">{{ $item->level_nama }}</option>
25                             @endforeach
26                         </select>
27                         <small class="form-text text-muted">Level Pengguna</small>
28                     </div>
29                 </div>
30             </div>
31         </div>
32         <table class="table table-bordered table-striped table-hover table-sm" id="table_user">
33             <thead>

```

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

```

55         ajax: {
56             "url": "{{ url('user/list') }}",
57             "dataType": "json",
58             "type": "POST",
59             "data": function(d) {
60                 d.level_id = $('#level_id').val();
61             }
62         },

```

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

```

92             searchable: false
93         },
94     ],
95     });
96     $('#level_id').on('change', function() {
97         dataUser.ajax.reload();
98     });
99     });
100 </script>
101 @endpush

```

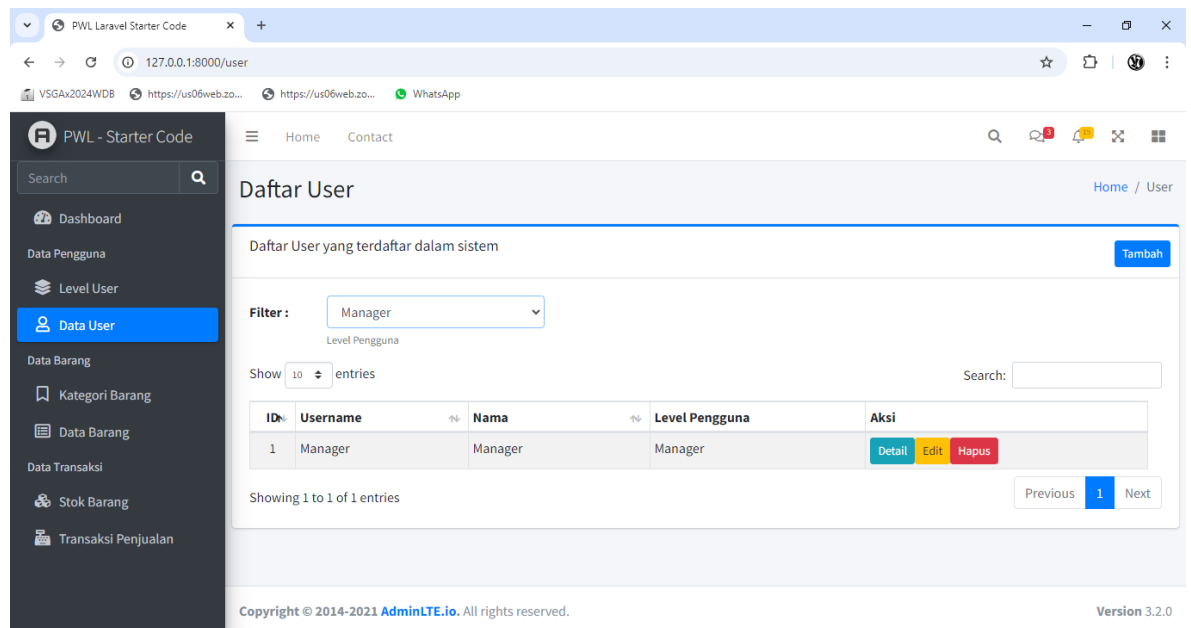
- Tahapan akhir adalah memodifikasi fungsi list() pada UserController.php yang digunakan untuk menampilkan data pada datatable

```

30         $users = UserModel::select('user_id', 'username', '
31
32         if ($request->level_id) {
33             $users->where('level_id', $request->level_id);
34         }
35
36         return DataTables::of($users)->addIndexColumn()->ad

```

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



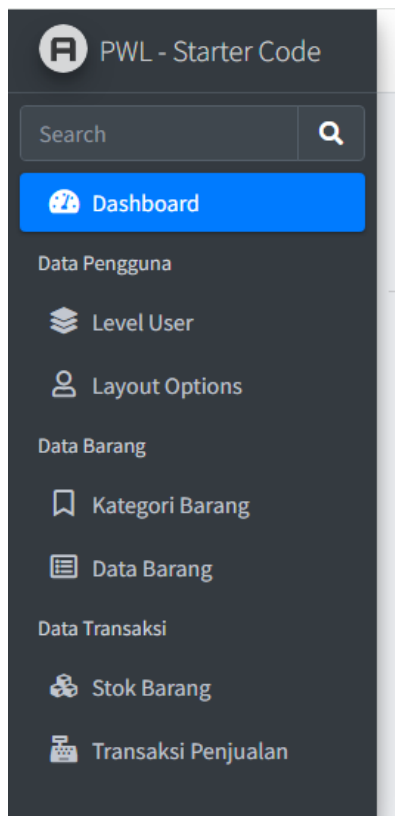
Pertanyaan

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 : @include(), @extend(), @section(), @push(), @yield(), dan @stack()
4. Apa fungsi dan tujuan dari variable \$activeMenu ?

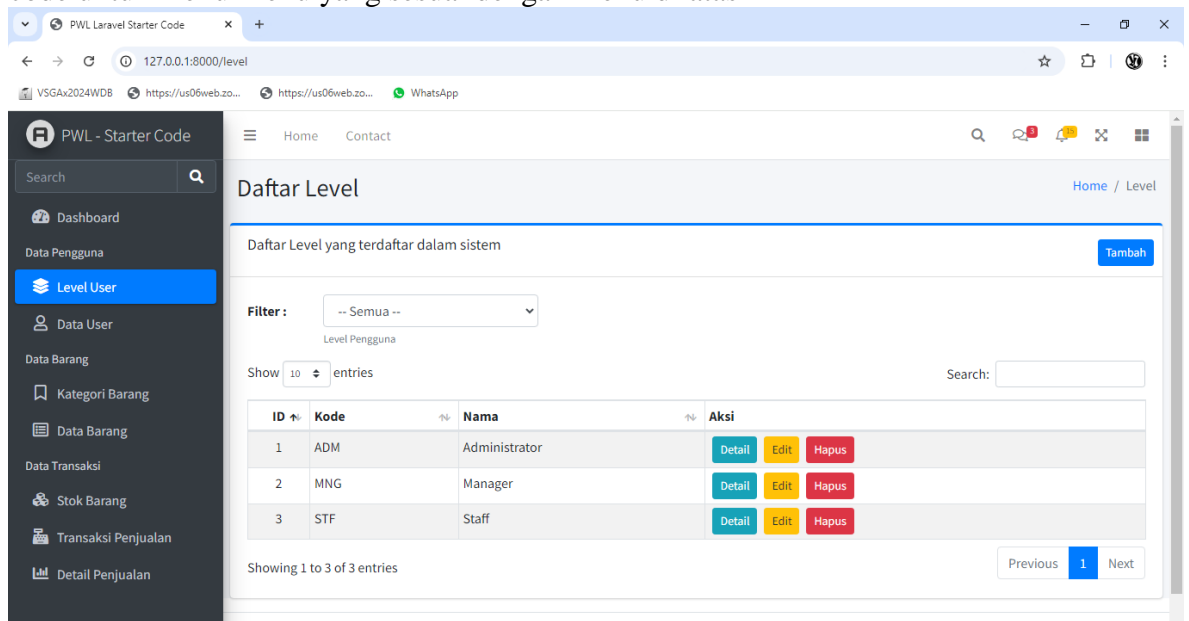
Jawab

1. Frontend Template menyediakan struktur dan desain antarmuka pengguna untuk situs web atau aplikasi. Template ini berfokus pada apa yang dilihat dan berinteraksi oleh pengguna sedangkan Backend Template menyediakan struktur dan logika pemrosesan data di sisi server. Template ini mengatur bagaimana data diproses, disimpan, dan diambil dari basis data.
2. Ya, karena layoting yang baik membantu pengguna menemukan informasi dengan mudah dan cepat, membuat pengguna merasa nyaman dan betah di situs, situs web lebih mudah digunakan, terlihat profesional, dan membantu dalam pencapaian tujuan bisnis.
3. @include digunakan untuk menyertakan tampilan Blade lainnya ke dalam tampilan yang sedang dikerjakan.
@extend digunakan untuk menunjukkan bahwa tampilan Blade tersebut adalah turunan dari tampilan lain.
@section digunakan untuk mendefinisikan bagian konten yang dapat diisi pada tampilan yang menggunakan layout.
@push digunakan untuk menambahkan konten ke dalam tumpukan tertentu, yang dapat dipanggil nanti menggunakan @stack.
@yield digunakan untuk menampilkan konten dari @section yang sesuai di layout utama. Ini adalah placeholder yang akan diisi oleh konten dari tampilan turunan.
@stack digunakan untuk menampilkan semua konten yang telah ditambahkan ke tumpukan menggunakan @push.
4. Variabel \$activeMenu berisi nilai yang menunjukkan menu mana yang sedang aktif. Misalnya, jika \$activeMenu diatur ke 'dashboard', maka item menu dashboard akan ditandai sebagai aktif.

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 atas



PWL - Starter Code

Search

Dashboard

Data Pengguna

Level User

Data User

Data Barang

Kategori Barang

Data Barang

Data Transaksi

Stok Barang

Transaksi Penjualan

Detail Penjualan

Home

Contact

Daftar Kategori

Home / Kategori

Daftar Kategori yang terdaftar dalam sistem

Tambah

Filter : -- Semua --

Kategori

Show 10 entries

Search:

ID	Kode	Nama	Aksi
1	ELK	Elektronik	<div>DetailEditHapus</div>
2	PKN	Pakaian	<div>DetailEditHapus</div>
3	MKN	Makanan	<div>DetailEditHapus</div>
4	KSH	Kesehatan	<div>DetailEditHapus</div>
5	KCT	Kecantikan	<div>DetailEditHapus</div>

PWL - Starter Code

Search

Dashboard

Data Pengguna

Level User

Data User

Data Barang

Kategori Barang

Data Barang

Data Transaksi

Stok Barang

Transaksi Penjualan

Detail Penjualan

Home

Contact

Daftar Barang

Home / Barang

Daftar Barang yang terdaftar dalam sistem

Tambah

Filter : -- Semua --

Kategori

Show 10 entries

Search:

ID	Kode	Nama	Harga Beli	Harga Jual	Kategori	Aksi
1	BRG001	Laptop	5000000	6000000	Elektronik	<div>DetailEditHapus</div>
2	BRG002	Smartphone	3000000	3500000	Elektronik	<div>DetailEditHapus</div>
3	BRG003	Jacket	150000	200000	Pakaian	<div>DetailEditHapus</div>
4	BRG004	Shirt	80000	120000	Pakaian	<div>DetailEditHapus</div>
5	BRG005	Chocolate	20000	30000	Makanan	<div>DetailEditHapus</div>
6	BRG006	Coffee	25000	35000	Makanan	<div>DetailEditHapus</div>