

Pemrograman Web Lanjut

Jobsheet 6 - Ajax Form (AdminLTE) dan Client Validation

Dosen Pengampu : Dimas Wahyu Wibowo, S.T., M.T.



Nama : Athallah Ayudya Paramesti
NIM : 2341760061
Program Studi : D-IV Sistem Informasi Bisnis

**Jurusan Teknologi Informasi
Politeknik Negeri Malang
2025**

Praktikum 1: Modal Ajax Tambah Data (Data User)

1. Buat form tambah data baru dengan menerapkan modal dan proses ajax.
2. Pertama yang kita siapkan adalah menambahkan **library jQuery Validation** dan **Sweetalert** ke aplikasi web kita. Caranya kita tambahkan link kedua library tersebut ke `template.blade.php`, library sudah disediakan oleh adminLTE.

```
<!-- jquery-validation -->
| <script src="{{ asset('adminlte/plugins/jquery-validation/jquery.validate.min.js') }}"></script>
| <script src="{{ asset('adminlte/plugins/jquery-validation/additional-methods.min.js') }}"></script>
| <!-- SweetAlert2 -->
| <link rel="stylesheet" href="{{ asset('adminlte/plugins/sweetalert2-theme-bootstrap-4/bootstrap-4.min.css') }}">
```

3. Selanjutnya Kita modifikasi view `user/index.blade.php`, kita tambahkan tombol untuk membuat form popup ajax

```
<a class="btn btn-sm btn-primary mt-1" href="{{ url('user/create') }}">Tambah</a>
| <button onclick="modalAction('{{ url('/user/create_ajax') }}')" class="btn btn-sm btn-success mt-1">Tambah Ajax</button>
```

Kita tambahkan kode berikut, untuk membuat form modal tambah data user dengan ajax

```
<button onclick="modalAction('{{ url('/user/create_ajax') }}')" class="btn btn-sm btn-success mt-1">Tambah Ajax</button>
```

4. Selanjutnya kita tambahkan kode berikut pada **akhir** `@section('content')` pada view `user/index.blade.php`

```
| <div id="myModal" class="modal fade animate shake" tabindex="-1" role="dialog" data-backdrop="static" data-keyboard="false" data-width="75%" aria-hidden="true"></div>
| @endsection
```

5. Kemudian kita tambahkan kode berikut pada awal `@push('js')` pada view `user/index.blade.php`

```
@push('js')
| <script>
|   function modalAction(url = ''){
|     $('#myModal').load(url, function(){
|       $('#myModal').modal('show');
|     });
|   }
| </script>
```

Sehingga tampilan kode program akan seperti berikut

```

<div id="myModal" class="modal fade animate shake" tabindex="-1" role="dialog" data-backdrop="static" data-keyboard="false" data-width="75%" aria-hidden="true"></div>
@endsection

@push('css')
@endpush

@push('js')
<script>
function modalAction(url = ''){
    $('#myModal').load(url, function(){
        $('#myModal').modal('show');
    });
}
$(document).ready(function() {
    var dataUser = $('#table_user').DataTable({
        // serverSide: true, jika ingin menggunakan server side processing
        serverSide: true,
        ajax: {

```

6. Selanjutkan modifikasi `route/web.php` untuk mengakomodir operasi ajax

```

Route::get('/create_ajax', [UserController::class, 'create_ajax']);
Route::post('/ajax', [UserController::class, 'store_ajax']);

```

7. Kemudian tambahkan fungsi `create_ajax()` pada file `UserController.php`

```

public function create_ajax(){
    $level = LevelModel::select('level_id', 'level_nama')->get();

    return view('user.create_ajax')
        |   |   |   ->with('level', $level);
}

```

8. Setelah itu, kita buat view baru `user/create_ajax.blade.php` untuk menampilkan form dengan ajax

```

<form action="{{ url('/user/ajax') }}" method="post" id="form-tambah">
    @csrf
    <div id="modal-master" class="modal-dialog modal-lg" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="exampleModalLabel">Tambah Data User</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>
            </div>
            <div class="modal-body">
                <div class="form-group">
                    <label>Level Pengguna</label>
                    <select name="level_id" id="level_id" class="form-control" required>
                        <option value="">- Pilih Level -</option>
                        @foreach($level as $l)
                            <option value="{{ $l->level_id }}>{{ $l->level_nama }}</option>
                        @endforeach
                    </select>
                    <small id="error-level_id" class="error-text form-text text-danger"></small>
                </div>
                <div class="form-group">
                    <label>Username</label>
                    <input value="" type="text" name="username" id="username" class="form-control" required>
                    <small id="error-username" class="error-text form-text text-danger"></small>
                </div>
                <div class="form-group">
                    <label>Nama</label>
                    <input value="" type="text" name="nama" id="nama" class="form-control" required>
                    <small id="error-nama" class="error-text form-text text-danger"></small>
                </div>
                <div class="form-group">
                    <label>Password</label>
                    <input value="" type="password" name="password" id="password" class="form-control" required>
                    <small id="error-password" class="error-text form-text text-danger"></small>
                </div>
            </div>
            <div class="modal-footer">
                <button type="button" data-dismiss="modal" class="btn btn-warning">Batal</button>
                <button type="submit" class="btn btn-primary">Simpan</button>
            </div>
        </div>
    </div>
</form>
<script>
$(document).ready(function() {
    $("#form-tambah").validate({
        rules: {
            level_id: {required: true, number: true},
            username: {required: true, minlength: 3, maxlength: 20},
            nama: {required: true, minlength: 3, maxlength: 100},
            password: {required: true, minlength: 6, maxlength: 20}
        },
        submitHandler: function(form) {
            $.ajax({
                url: form.action,
                type: form.method,
                data: $(form).serialize(),
                success: function(response) {
                    if(response.status){
                        $('#myModal').modal('hide');
                        Swal.fire({
                            icon: 'success',
                            title: 'Berhasil',
                            text: response.message
                        });
                        dataUser.ajax.reload();
                    }else{
                        $('.error-text').text('');
                        $.each(response.msgField, function(prefix, val) {
                            $('#error-' + prefix).text(val[0]);
                        });
                    }
                }
            })
        }
    })
})

```

```

        Swal.fire({
            icon: 'error',
            title: 'Terjadi Kesalahan',
            text: response.message
        });
    }
});

return false;
},
errorElement: 'span',
errorPlacement: function (error, element) {
    error.addClass('invalid-feedback');
    element.closest('.form-group').append(error);
},
highlight: function (element, errorClass, validClass) {
    $(element).addClass('is-invalid');
},
unhighlight: function (element, errorClass, validClass) {
    $(element).removeClass('is-invalid');
}
});

```

9. Kemudian untuk mengakomodir proses simpan data melalui ajax, kita buat fungsi `store_ajax()` pada `UserController.php`

```

public function store_ajax(Request $request){
    //cek apakah request berupa ajax
    if($request->ajax() || $request->wantsJson()){
        $rules =[

            'level_id' => 'required|integer',
            'username' => 'required|string|min:3|unique:m_user,username',
            'nama'      => 'required|string|max:100',
            'password'  => 'required|min:6'
        ];

        //use Illuminate\Support\Facades\Validator;
        $validator = Validator::make($request->all(), $rules);

        if($validator->fails()){
            return response()->json([
                'status' => false, //response status, false: error/gagal, true:berhasil
                'message' => 'Validasi Gagal',
                'msgField' => $validator->errors(), //pesan error validasi
            ]);
        }

        UserModel::create($request->all());
        return response()->json([
            'status' => true,
            'message' => 'Data user berhasil disimpan'
        ]);
    }
    redirect('/');
}

```

10. OK, sekarang kita coba melakukan proses tambah data user menggunakan form ajax. Amati apa yang terjadi dan dilaporkan pada laporan jobsheet dan commit di github kalian!!!

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 Satu	Customer	[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	manager12	Manager satu satuu	Manager	[Detail] [Edit] [Hapus]

10	customer-3	Pelanggan Tiga	Customer	[Detail] [Edit] [Hapus]
----	------------	----------------	----------	-------------------------

Praktikum 2: Modal Ajax Edit Data (Data User)

1. Pada Praktikum 2 ini, kita akan melakukan coding untuk proses edit menggunakan ajax.

2. Pertama-tama, kita ubah dulu fungsi list() pada UserController.php untuk mengganti tombol edit untuk bisa menggunakan modal

```

    $btn = '<button onclick="modalAction(`'.url('/user/' . $user->user_id .
'/show_ajax`).')" class="btn btn-info btn-sm">Detail</button> ';
    $btn .= '<button onclick="modalAction(`'.url('/user/' . $user->user_id .
'/edit_ajax`).')" class="btn btn-warning btn-sm">Edit</button> ';
    $btn .= '<button onclick="modalAction(`'.url('/user/' . $user->user_id .
'/delete_ajax').')" class="btn btn-danger btn-sm">Hapus</button> ';

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

```

3. Selanjutnya kita modifikasi routes/web.php untuk mengakomodir request edit menggunakan ajax

```

Route::get('{id}/edit_ajax', [UserController::class, 'edit_ajax']); //menampilkan halaman form edit user Ajax
Route::put('{id}/update_ajax', [UserController::class, 'update_ajax']); //menyimpan perubahan data user Ajax

```

4. Kemudian, kita buat fungsi edit_ajax() pada UserController.php

```

// Menampilkan halaman form edit user ajax
public function edit_ajax(string $id){
    $user = UserModel::find($id);
    $level = LevelModel::select('level_id', 'level_nama')->get();

    return view('user.edit_ajax', ['user' => $user, 'level' => $level]);
}

```

5. Kita buat **view baru** pada user/edit_ajax.blade.php untuk menampilkan form view ajax

```

@empty($user)
    <div id="modal-master" class="modal-dialog modal-lg" role="document">
        <div class="modal-content">
            <div class="modal_header">
                <h5 class="modal-title" id="exampleModalLabel">Kesalahan</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>
            </div>
            <div class="modal-body">
                <div class="alert alert-danger">
                    <h5><i class="icon fas fa-ban"></i> Kesalahan!!!</h5>
                    Data yang anda cari tidak ditemukan
                </div>
                <a href="{{ url('/user') }}" class="btn btn-warning">Kembali</a>
            </div>
        </div>
    </div>
@else
    <form action="{{ url('/user/' . $user->user_id . '/update_ajax') }}" method="POST" id="form-edit">
        @csrf
        @method('PUT')
        <div id="modal-master" class="modal-dialog modal-lg" role="document">
            <div class="modal-content">
                <div class="modal_header">
                    <h5 class="modal-title" id="exampleModalLabel">Edit Data User</h5>
                    <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>
                </div>
                <div class="modal-body">
                    <div class="form-group">
                        <label>Level Pengguna</label>
                        <select name="level_id" id="level_id" class="form-control" required>
                            <option value="">- Pilih Level -</option>
                            @foreach($level as $l)
                                <option {{ ($l->level_id == $user->level_id)? 'selected' : '' }}>{{ $l->level_nama }}</option>
                            @endforeach
                        </select>
                        <small id="error-level_id" class="error-text form-text text-danger"></small>
                    </div>
                </div>
            </div>
        </div>
    </form>
@endforeach

```

```


<label>Username</label>
    <input value="{{ $user->username }}" type="text" name="username" id="username" class="form-control" required>
    <small id="error-username" class="error-text form-text text-danger"></small>


<div class="form-group">
    <label>Nama</label>
    <input value="{{ $user->nama }}" type="text" name="nama" id="nama" class="form-control" required>
    <small id="error-nama" class="error-text form-text text-danger"></small>

```

```

        },
        unhighlight: function (element, errorClass, validClass) {
            $(element).removeClass('is-invalid');
        }
    );
});

```

@endempty

- Selanjutnya, kita buat fungsi `update_ajax()` pada `UserController.php` untuk mengakomodir request update data user melalui ajax

```

public function update_ajax(Request $request, $id){
    // cek apakah request dari ajax
    if ($request->ajax() || $request->wantsJson()) {
        $rules = [
            'level_id' => 'required|integer',
            'username' => 'required|max:20|unique:m_user,username,'.$id.',user_id',
            'nama'      => 'required|max:100',
            'password'  => 'nullable|min:6|max:20'
        ];
        // use Illuminate\Support\Facades\Validator;
        $validator = Validator::make($request->all(), $rules);

        if ($validator->fails()) {
            return response()->json([
                'status'  => false,    // respon json, true: berhasil, false: gagal
                'message' => 'Validasi gagal.',
                'msgField' => $validator->errors() // menunjukkan field mana yang error
            ]);
        }
    }

    $check = UserModel::find($id);
    if ($check) {
        if (!$request->filled('password') ) { // jika password tidak diisi, maka hapus dari request
            $request->request->remove('password');
        }

        $check->update($request->all());
        return response()->json([
            'status'  => true,
            'message' => 'Data berhasil diupdate'
        ]);
    } else{
        return response()->json([
            'status'  => false,
            'message' => 'Data tidak ditemukan'
        ]);
    }
    return redirect('/');
}

```

- Sekarang kita coba bagian edit user, amati proses nya. Jangan lupa laporkan dan commit ke repository git kalian !

Daftar User					
Daftar user yang terdaftar dalam sistem					
Tambah Tambah Ajax					
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 Satu	Customer	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	manager12	Manager satu satu	Manager	Detail Edit Hapus	
9	customer-2	Pelanggan Dua	Customer	Detail Edit Hapus	

Edit Data User

Level Pengguna

Manager

Username

manager12

Nama

Manager satu satu

Password

Abaikan jika tidak ingin ubah password

Batal [Simpan](#)

9	customer-2	Pelanggan Dua	Customer	Detail	Edit	Hapus
---	------------	---------------	----------	------------------------	----------------------	-----------------------

Edit Data User

Level Pengguna

Manager

Username

manager11

Nama

Manager satu satu

Password

Abaikan jika tidak ingin ubah password

Batal [Simpan](#)

9	customer-2	Pelanggan Dua	Customer	Detail	Edit	Hapus
---	------------	---------------	----------	------------------------	----------------------	-----------------------

8	manager11	Manager satu satu	Manager	Detail	Edit	Hapus
---	-----------	-------------------	---------	------------------------	----------------------	-----------------------

Praktikum 3: Modal Ajax Hapus Data (Data User)

1. Pada Praktikum 3 ini, kita akan melakukan coding untuk proses hapus menggunakan ajax.
2. Pertama-tama, kita ubah dulu `routes/web.php` untuk mengakomodir request halaman konfirmasi untuk menghapus data

```
Route::get('/{id}/delete_ajax', [UserController::class, 'confirm_ajax']); // untuk tampilkan form confirm delete user ajax
Route::delete('/{id}/delete_ajax', [UserController::class, 'delete_ajax']); //untuk hapus data user Ajax
```

3. Kemudian kita buat fungsi `confirm_ajax()` pada `UserController.php`

```
public function confirm_ajax(string $id){
    $user = UserModel::find($id);

    return view('user.confirm_ajax', ['user' => $user]);
}
```

4. Selanjutnya kita view untuk konfirmasi hapus data dengan nama `user/confirm_ajax.blade.php`

```
@empty($user)
    <div id="modal-master" class="modal-dialog modal-lg" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="exampleModalLabel">Kesalahan</h5>
                <button type="button" class="close" data-dismiss="modal" aria
label="Close"><span aria-hidden="true">&times;</span></button>
            </div>
            <div class="modal-body">
                <div class="alert alert-danger">
                    <h5><i class="icon fas fa-ban"></i> Kesalahan!!!</h5>
                    Data yang anda cari tidak ditemukan</div>
                    <a href="{{ url('/user') }}" class="btn btn-warning">Kembali</a>
                </div>
            </div>
        </div>
    @else
        <form action="{{ url('/user/' . $user->user_id.'/delete_ajax') }}" method="POST" id="form
delete">
            @csrf
            @method('DELETE')
            <div id="modal-master" class="modal-dialog modal-lg" role="document">
                <div class="modal-content">
                    <div class="modal-header">
                        <h5 class="modal-title" id="exampleModalLabel">Hapus Data User</h5>
                        <button type="button" class="close" data-dismiss="modal" aria
label="Close"><span aria-hidden="true">&times;</span></button>
                    </div>
                    <div class="modal-body">
                        <div class="alert alert-warning">
                            <h5><i class="icon fas fa-ban"></i> Konfirmasi !!!</h5>
                            Apakah Anda ingin menghapus data seperti di bawah ini?
                        </div>
                        <table class="table table-sm table-bordered table-striped">
                            <tr><th class="text-right col-3">Level Pengguna :</th><td class="col-9">{{
$user->level->level_nama }}</td></tr>
                            <tr><th class="text-right col-3">Username :</th><td class="col-9">{{
$user->username }}</td></tr>
                            <tr><th class="text-right col-3">Nama :</th><td class="col-9">{{ $user
}}
```

```

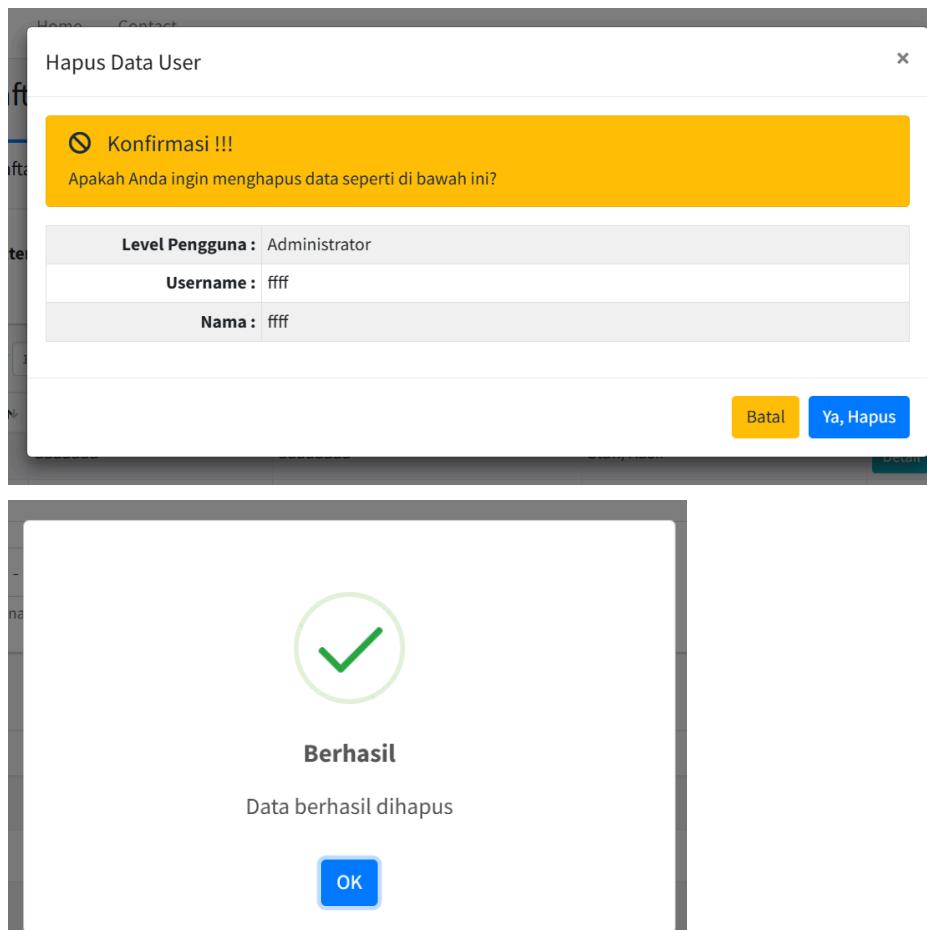
>nama })</td></tr>
    </table>
</div>
<div class="modal-footer">
    <button type="button" data-dismiss="modal" class="btn btn-warning">Batal</button>
    <button type="submit" class="btn btn-primary">Ya, Hapus</button>
</div>
</div>
</form>
<script>
    $(document).ready(function() {
        $("#form-delete").validate({
            rules: {},
            submitHandler: function(form) {
                $.ajax({
                    url: form.action,
                    type: form.method,
                    data: $(form).serialize(),
                    success: function(response) {
                        if(response.status){
                            $('#myModal').modal('hide');
                            Swal.fire({
                                icon: 'success',
                                title: 'Berhasil',
                                text: response.message
                            });
                            dataUser.ajax.reload();
                        }else{
                            $('.error-text').text('');
                            $.each(response.msgField, function(prefix, val) {
                                $('#error-' +prefix).text(val[0]);
                            });
                            Swal.fire({
                                icon: 'error',
                                title: 'Terjadi Kesalahan',
                                text: response.message
                            });
                        }
                    },
                    errorElement: 'span',
                    errorPlacement: function (error, element) {
                        error.addClass('invalid-feedback');
                        element.closest('.form-group').append(error);
                    },
                    highlight: function (element, errorClass, validClass) {
                        $(element).addClass('is-invalid');
                    },
                    unhighlight: function (element, errorClass, validClass) {
                        $(element).removeClass('is-invalid');
                    }
                });
            }
        });
    });
</script>
@endempty

```

5. Kemudian kita buat fungsi `delete_ajax()` pada `UserController.php` untuk mengakomodir request hapus data user

```
public function delete_ajax(Request $request, $id){  
    //cek apakah request dari ajax  
    if ($request->ajax() || $request->wantsJson()) {  
        $user = UserModel::find($id);  
        if ($user) {  
            $user->delete();  
            return response()->json([  
                'status' => true,  
                'message' => 'Data berhasil dihapus'  
            ]);  
        } else {  
            return response()->json([  
                'status' => false,  
                'message' => 'Data tidak ditemukan'  
            ]);  
        }  
    }  
    return redirect('/');  
}
```

6. Setelah semua selesai, mari kita coba untuk melakukan percobaan dari koding yang telah kita lakukan.



7. Jangan lupa laporkan ke laporan jobsheet dan lakukan *commit* pada repository git kalian!!!

Tugas Jobsheet

Implementasikan coding untuk Ajax Form dan Client Validation dengan jQuery Validation pada menu berikut ini

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

Laporkan pada laporan jobsheet dan Jangan lupa di commit dan push pada repository git kalian

Tabel m_level

The screenshot shows a web application interface for managing user levels. On the left is a sidebar with navigation links: Dashboard, Data Pengguna, Level User (which is selected and highlighted in blue), Data User, Data Barang, Kategori Barang, Data Barang, Data Transaksi, Stok Barang, Transaksi Penjualan, and Supplier. The main content area has a title 'Daftar Level' and a subtitle 'Daftar level yang terdaftar dalam sistem'. It includes a 'Filter' dropdown set to '- Semua -' and a 'Show' dropdown set to '10 entries'. A search bar and a 'Search:' input field are also present. Below these are four rows of data in a table:

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]
4	CUS	Customer	[Detail] [Edit] [Hapus]

At the bottom, it says 'Showing 1 to 4 of 4 entries' and has 'Previous' and 'Next' buttons. The footer includes 'Copyright © 2014-2021 AdminLTE.io. All rights reserved.' and 'Version 3.2.0'.

The screenshot shows a modal dialog titled 'Tambah Data Level' (Add New Level). It contains two input fields: 'Kode Level' and 'Level Nama', both currently empty. Below the inputs is a 'Filter' dropdown. At the bottom right of the modal are 'Batal' (Cancel) and 'Simpan' (Save) buttons. In the background, the 'Daftar Level' page is visible, showing the same table of user levels. The sidebar on the left remains the same as the previous screenshot.

Laravel 127.0.0.1:8000/level

PWL - Starter Code

Home Contact

Search

Dashboard Data Pengguna Level User Data User Data Barang Kategori Barang Data Transaksi Stok Barang Transaksi Penjualan Supplier

Daftar Level

Daftar level yang terdaftar dalam sistem

Tambah Tambah Ajax

ID Level Kode Level Nama Aksi

ID	Level Kode	Level Nama	Aksi
1	ADM	Administrator	[Detail] [Edit] [Hapus]
2	CUS	Customer	[Detail] [Edit] [Hapus]
3	MNG	Manager	[Detail] [Edit] [Hapus]
4	STF	Staff/Kasir	[Detail] [Edit] [Hapus]

Showing 1 to 4 of 4 entries

Previous 1 Next

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

Version 3.2.0

Laravel 127.0.0.1:8000/level

PWL - Starter Code

Home Contact

Search

Dashboard Data Pengguna Level User Data User Data Barang Kategori Barang Data Transaksi Stok Barang Transaksi Penjualan Supplier

Edit Data Level

Level Kode

Level Nama

Batal Simpan

ID Level Kode Level Nama Aksi

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]
4	CUS	Customer	[Detail] [Edit] [Hapus]

Showing 1 to 4 of 4 entries

Previous 1 Next

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

Version 3.2.0

Laravel 127.0.0.1:8000/level

PWL - Starter Code

Home Contact

Search

Dashboard Data Pengguna Level User Data User Data Barang Kategori Barang Data Transaksi Stok Barang Transaksi Penjualan Supplier

Hapus Data level

Konfirmasi !!!

Apakah Anda ingin menghapus data seperti di bawah ini?

level Kode : ADM
level Nama : Administrator

Batal Ya, Hapus

ID Level Kode Level Nama Aksi

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]
4	CUS	Customer	[Detail] [Edit] [Hapus]

Showing 1 to 4 of 4 entries

Previous 1 Next

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

Version 3.2.0

Tabel m_kategori

Daftar Kategori

ID	Kode Kategori	Nama Kategori	Aksi
1	FOOD	Makanan	[Detail] [Edit] [Hapus]
2	DRINK	Minuman	[Detail] [Edit] [Hapus]
3	BEAUTY	Kecantikan	[Detail] [Edit] [Hapus]
4	HOME	Peralatan Rumah	[Detail] [Edit] [Hapus]
5	BABY	Perlengkapan Bayi	[Detail] [Edit] [Hapus]
6	SPORT	Perlengkapan Olahraga	[Detail] [Edit] [Hapus]

Showing 1 to 6 of 6 entries

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

Version 3.2.0

Tambah Data Kategori

Kode Kategori

Kategori Name

ID	Kode Kategori	Nama Kategori	Aksi
1			[Detail] [Edit] [Hapus]
2			[Detail] [Edit] [Hapus]

Showing 1 to 6 of 6 entries

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

Version 3.2.0

Berhasil

Data kategori berhasil disimpan

OK

7 HEALTY Kesehatan [Detail] [Edit] [Hapus]

Laravel

127.0.0.1:8000/kategori

PWL - Starter Code

Home Contact

Search

Dashboard

Data Pengguna

Level User

Data User

Data Barang

Kategori Barang

Data Barang

Data Transaksi

Stok Barang

Transaksi Penjualan

Supplier

Daftar Kategori

Daftar kategori yang terdaftar dalam sistem

Tambah Tambah Ajax

Detail kategori

ID	Kode Kategori	Nama Kategori
1	FOOD	Makanan
6	SPORT	Perlengkapan Olahraga
7	HEALTY	Kesehatan

Showing 1 to 7 of 7 entries

Previous 1 Next

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

Version 3.2.0

Laravel

127.0.0.1:8000/kategori

PWL - Starter Code

Home Contact

Search

Dashboard

Data Pengguna

Level User

Data User

Data Barang

Kategori Barang

Data Barang

Data Transaksi

Stok Barang

Transaksi Penjualan

Supplier

Edit Data Kategori

Kategori Kode

FOOD

Kategori Nama

Makanan

Batal Simpan

3	BEAUTY	Kecantikan
4	HOME	Peralatan Rumah
5	BABY	Perlengkapan Bayi
6	SPORT	Perlengkapan Olahraga
7	HEALTY	Kesehatan

Showing 1 to 7 of 7 entries

Previous 1 Next

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

Version 3.2.0

Laravel

127.0.0.1:8000/kategori

PWL - Starter Code

Home Contact

Search

Dashboard

Data Pengguna

Level User

Data User

Data Barang

Kategori Barang

Data Barang

Data Transaksi

Stok Barang

Transaksi Penjualan

Supplier

Hapus Data Kategori

Konfirmasi !!!

Apakah Anda ingin menghapus data seperti di bawah ini?

Kategori Kode : HEALTY

Kategori Nama : Kesehatan

Batal Ya, Hapus

3	BEAUTY	Kecantikan
4	HOME	Peralatan Rumah Tangga
5	BABY	Perlengkapan Bayi
6	SPORT	Perlengkapan Olahraga
7	HEALTY	Kesehatan

Showing 1 to 7 of 7 entries

Previous 1 Next

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

Version 3.2.0

Tabel m_supplier

ID	Kode Supplier	Nama Supplier	Alamat Supplier	Aksi
1	SUP001	PT Maju Mundur	Jl. Mawar No. 12, Surabaya	[Detail] [Edit] [Hapus]
2	SUP002	PT Jaya Makmur	Jl. Kawi No. 06, Malang	[Detail] [Edit] [Hapus]

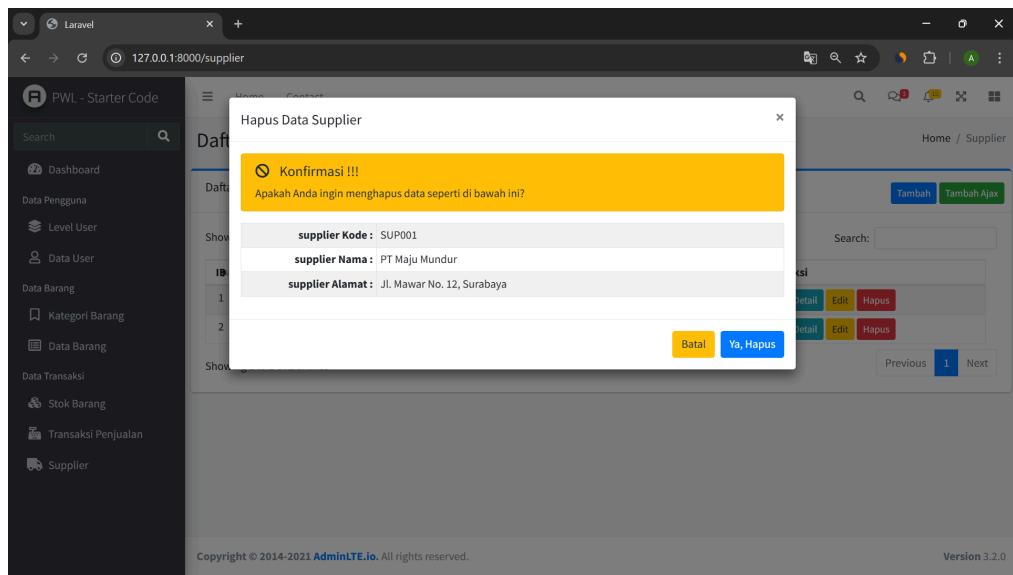
Kode Supplier
Nama Supplier
Alamat Supplier

Batal Simpan

Detail Supplier

Kode Supplier	SUP001
Nama Supplier	PT Maju Mundur
Alamat Supplier	Jl. Mawar No. 12, Surabaya

Tutup



Tabel m_barang

The screenshot shows a Laravel application interface. The main page title is "Data Barang". It displays a table of items with the following columns: ID, Kode Barang, Nama Barang, Nama Kategori, Harga Beli, Harga Jual, and Aksi (Actions). The table has 8 rows of data. The "Aksi" column contains three buttons for each row: "Detail", "Edit", and "Hapus".

ID	Kode Barang	Nama Barang	Nama Kategori	Harga Beli	Harga Jual	Aksi
1	BRG001	Nasi Goreng	Makanan	10000	15000	[Detail] [Edit] [Hapus]
2	BRG002	Mie Goreng	Makanan	9000	14000	[Detail] [Edit] [Hapus]
3	BRG003	Es Teh	Minuman	3000	5000	[Detail] [Edit] [Hapus]
4	BRG004	Jus Jeruk	Minuman	5000	8000	[Detail] [Edit] [Hapus]
5	BRG005	Lipstik	Kecantikan	25000	40000	[Detail] [Edit] [Hapus]
6	BRG006	Face Wash	Kecantikan	15000	25000	[Detail] [Edit] [Hapus]
7	BRG007	Sapu Lidi	Peralatan Rumah Tangga	10000	20000	[Detail] [Edit] [Hapus]
8	BRG008	Pel Lantai	Peralatan Rumah Tangga	20000	35000	[Detail] [Edit] [Hapus]

The screenshot shows a Laravel application interface. A modal window titled "Tambah Data Barang" (Add New Item) is displayed. The form fields include:

- Kategori Barang**: A dropdown menu labeled "Pilih Kategori".
- Barang Kode**: An input field for the item code.
- Barang Nama**: An input field for the item name.
- Harga Beli**: An input field for the purchase price.
- Harga Jual**: An input field for the selling price.

At the bottom of the form are two buttons: "Batal" (Cancel) and "Simpan" (Save).

The background shows the same "Data Barang" table as the previous screenshot, with 8 rows of data.

