LAMPIRAN

Controller

AdminController.php

```
<?php

namespace App\Http\Controllers;

use App\Models\Informasi;
use App\Models\InformasiPakan;
use App\Models\Konsultasi;
use App\Models\Ternak;
use App\Models\User;
use Illuminate\Http\Request;
use
Illuminate\Support\Facades\Auth;
use
Illuminate\Support\Facades\Hash;
use Illuminate\Support\Str;
use Illuminate\Validation\Rule;

class AdminController extends
Controller
{
    public function index()
    {
        $totalTernak =
Ternak::all()->count();
        $ternaks =
Ternak::with('pemilik')->get();
        return
view('admin.dashboard',
compact('totalTernak',
'ternaks'));
    }
    public function userView()
    {
        $userList = User::all();
        $totalUser = User::all()-
>count();
        $userAktif =
User::where('status', 'aktif')-
>count();
        $userNonAktif =
User::where('status',
'nonaktif')->count();
        $userSuspend =
User::where('status', 'suspend')-
>count();
        return view('admin.user',
compact('userList', 'totalUser',
'userAktif', 'userNonAktif',
'userSuspend'));
    }

    public function
storeUser(Request $request)
    {
        try {

            $validated =
$request->validate([
                'nama' =>
'required|string|max:255',
                'email' =>
required|email|unique:users,email
',
                'password' =>
'required|string|min:8|confirmed'
,
                'role' =>
'required|in:admin,peternak,user'
,
                'phone' =>
'nullable|string|max:20',
                'status' =>
'required|in:aktif,nonaktif,suspe
nd'
            ]);

            $user = new User();
            $user->nama =
$validated['nama'];
            $user->email =
$validated['email'];
            $user->password =
Hash::make($validated['password']
);
            $user->role =
$validated['role'];
            $user->phone =
$validated['phone'];
            $user->status =
$validated['status'];
            $user->save();


            if ($request->ajax()
|| $request->wantsJson()) {
                return
response()->json([
                    'success' =>
true,
                    'message' =>
'User berhasil ditambahkan!',
                    'data' =>
$user
                ]);
            }

            return redirect()-
>route('admin.userView')-
>with('success', 'User berhasil
ditambahkan!');
        } catch
(\Illuminate\Validation\Validatio
nException $e) {
            if ($request->ajax()
|| $request->wantsJson()) {
                return
response()->json([
                    'success' =>
false,
                    'message' =>
'Validasi gagal',
                    'errors' =>
$e->errors()
```

```
                ], 422);
            }

            return back()->withErrors($e->errors())->withInput();
        } catch (\Exception $e) {
            if ($request->ajax() || $request->wantsJson()) {
                return response()->json([
                    'success' => false,
                    'message' => 'Terjadi kesalahan saat menyimpan data: ' . $e->getMessage()
                ], 500);
            }

            return back()->with('error', 'Terjadi kesalahan saat menyimpan data')->withInput();
        }
    }
    public function updateUser(Request $request, $idUser)
    {
        try {
            $user = User::findOrFail($idUser);
            $validated = $request->validate([
                'nama' => 'required|string|max:255',
                'email' => [
                    'required',
                    'email',
                    Rule::unique('users', 'email')->ignore($idUser, 'idUser')
                ],
                'role' => 'required|in:admin,peternak,user',
                'phone' => 'nullable|string|max:20',
                'status' => 'required|string|max:255'
            ]);

            $user->nama = $validated['nama'];
            $user->email = $validated['email'];
            $user->role = $validated['role'];
            $user->phone = $validated['phone'];
            $user->status = $validated['status'];

            if ($request->has('reset_password') && $request->reset_password) {
                $newPassword = Str::random(8);
                $user->password = Hash::make($newPassword);
            }

            $user->save();

            // Cek apakah request AJAX
            if ($request->ajax() || $request->wantsJson()) {
                return response()->json([
                    'success' => true,
                    'message' => 'User berhasil diupdate!',
                    'data' => $user
                ]);
            }

            return redirect()->route('admin.userView')->with('success', 'User berhasil diupdate!');
        } catch (\Illuminate\Validation\ValidationException $e) {
            if ($request->ajax() || $request->wantsJson()) {
                return response()->json([
                    'success' => false,
                    'message' => 'Validasi gagal',
                    'errors' => $e->errors()
                ], 422);
            }

            return back()->withErrors($e->errors())->withInput();
        } catch (\Exception $e) {
            if ($request->ajax() || $request->wantsJson()) {
                return response()->json([
                    'success' => false,
                    'message' => 'Terjadi kesalahan saat mengupdate data: ' . $e->getMessage()
                ], 500);
            }
```

```
            return back()->with('error', 'Terjadi kesalahan saat mengupdate data')->withInput();
        }
    }


    public function deleteUser($idUser)
    {
        try {
            $user = User::find($idUser);
            if (!$user) {
                if (request()->ajax() || request()->wantsJson()) {
                    return response()->json([
                        'success' => false,
                        'message' => 'User tidak ditemukan atau sudah dihapus sebelumnya.'
                    ], 404);
                }

                return back()->with('error', 'User tidak ditemukan atau sudah dihapus sebelumnya.');
            }

            $userName = $user->nama;


            $user->delete();


            // Cek apakah request AJAX
            if (request()->ajax() || request()->wantsJson()) {
                return response()->json([
                    'success' => true,
                    'message' => "User {$userName} berhasil dihapus!"
                ]);
            }

            return redirect()->route('admin.userView')->with('success', "User {$userName} berhasil dihapus!");
        } catch (\Exception $e) {
            if (request()->ajax() || request()->wantsJson()) {
                return response()->json([
                    'success' => false,
                    'message' => 'Terjadi kesalahan saat menghapus data. Silakan coba lagi.'
                ], 500);
            }

            return back()->with('error', 'Terjadi kesalahan saat menghapus data. Silakan coba lagi.');
        }
    }
    public function storeInformasi(Request $request)
    {
        $user = new User();
        $user->name = $request->name;
        $user->email = $request->email;
        // $user->password = Hash::make($request->password);
        $user->save();
        return redirect()->route('admin.userView');
    }
    public function storePakan(Request $request)
    {
        $user = new User();
        $user->name = $request->name;
        $user->email = $request->email;
        // $user->password = Hash::make($request->password);
        $user->save();
        return redirect()->route('admin.userView');
    }

    public function ternak()
    {
        $ternakList = Ternak::where('idPemilik', Auth::user()->idUser)->get();
        $totalTernak = $ternakList->count();
        $ternakSehat = Ternak::where('idPemilik', Auth::user()->idUser)->where('status', 'sehat')->count();
        $ternakSakit = Ternak::where('idPemilik', Auth::user()->idUser)-
```

```
>where('status', 'sakit')-
>count();
        $ternakPerawatan =
Ternak::where('idPemilik',
Auth::user()->idUser)-
>where('status', 'perawatan')-
>count();
        $konsultasiSaya =
Konsultasi::where('idPeternak',
Auth::user()->idUser)->count();
        return
view('admin.ternak', compact(
            'ternakList',
            'totalTernak',
            'ternakSehat',
            'ternakSakit',
            'ternakPerawatan',
            'konsultasiSaya'
        ));
    }

    public function informasi()
    {
        $informasiList =
Informasi::latest()-
>paginate(10);
        $totalInformasi =
Informasi::count();
        $informasiPublished =
Informasi::count();
        $totalViews =
$informasiList->count() * 100;
        $penyuluhs =
User::where('role', 'Penyuluh')-
>get();

        return
view('admin.informasi', compact(
            'informasiList',
            'totalInformasi',
            'informasiPublished',
            'totalViews',
            'penyuluhs'
        ));
    }

    public function pakan()
    {
        $informasiPakan =
InformasiPakan::count();
        $pakanList =
InformasiPakan::latest()-
>paginate(10);
        $draftPakan =
InformasiPakan::where('is_publish
ed', 0)->count();
        $publishedPakan =
InformasiPakan::where('is_publish
ed', 1)->count();
        return
view('admin.pakan',
compact('pakanList',
'informasiPakan', 'draftPakan',
'publishedPakan'));
    }
}
```

AdminSesiController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use
Illuminate\Support\Facades\Auth;
use
Illuminate\Support\Facades\Hash;
use App\Models\User;

class AdminSesiController extends
Controller
{
    public function
adminLoginView()
    {
        return
view('auth.login');
    }

    public function
registerView()
    {
        return
view('auth.register');
    }

    public function login(Request
$request)
    {
        $request->validate([
            'email' =>
'required|email',
            'password' =>
'required|min:6',
        ], [
            'email.required' =>
'Silahkan Masukkan Email Anda',
            'email.email' =>
'Format email yang Anda masukkan
tidak valid',
            'password.required'
=> 'Silahkan Masukkan Password
Anda',
            'password.min' =>
'Password minimal terdiri dari 6
karakter',
        ]);

        $infologin = [
            'email' => $request-
>email,
            'password' =>
$request->password,
        ];
```

```
        if
(Auth::attempt($infologin,
$request->has('remember'))) {
            $user = Auth::user();

            if ($user->role ==
"Peternak") {
                return
redirect('/peternak/dashboard');
            } elseif ($user->role
== "Penyuluh") {
                return
redirect('/penyuluh/dashboard');
            } elseif ($user->role
== 'Admin') {
                return
redirect('/admin/dashboard');
            }
        }
        return
redirect('/admin/login')-
>withErrors(['login' => 'Login
Gagal, Email atau Password tidak
sesuai!'])->withInput();
    }
    public function
register(Request $request)
    {
        $request->validate([
            'nama' =>
'required|string|max:255',
            'email' =>
'required|email|unique:users,emai
l',
            'password' =>
'required|string|min:6|confirmed'
,
            'phone' =>
'required|numeric',
            'role' =>
'required|string|in:Peternak,Peny
uluh',
        ], [
            'nama.required' =>
'Nama Wajib Diisi',
            'email.required' =>
'Silahkan Masukkan Email Anda',
            'email.email' =>
'Format email yang Anda masukkan
tidak valid',
            'email.unique' =>
'Email sudah terdaftar',
            'password.required'
=> 'Silahkan Masukkan Password
Anda',
            'password.min' =>
'Password minimal terdiri dari 6
karakter',
            'password.confirmed'
=> 'Konfirmasi password tidak
sesuai',
            'phone.required' =>
'Nomor telepon wajib diisi',
            'role.required' =>
'    hkan Pilih Role Anda',
        ]);

        $user = User::create([
            'nama' => $request-
>nama,
            'email' => $request-
>email,
            'password' =>
Hash::make($request->password),
            'phone' => $request-
>phone,
            'role' => $request-
>role
        ]);

        return
redirect('/admin/login')-
>with('success', 'Registrasi
berhasil! Silahkan Login Ke Akun
Anda.');
    }
    public function
logout(Request $request)
    {
        Auth::logout();
        $request->session()-
>invalidate();
        $request->session()-
>regenerateToken();
        return
redirect('/admin/login');
    }
}
```

InformasiController.php

```
<?php

namespace App\Http\Controllers;

use App\Models\Informasi;
use Illuminate\Http\Request;
use
Illuminate\Support\Facades\Storag
e;
use Str;

class InformasiController extends
Controller
{
    public function index()
    {
        $informasiList =
Informasi::latest()-
>paginate(10);
        $totalInformasi =
Informasi::count();
        return
view('informasi.index',
compact('informasiList',
'totalInformasi'));
```

```
    }

    public function store(Request
$request)
    {
        $request->validate([
            'judul' =>
'required|string|max:255',
            'deskripsi' =>
'required|string|min:10',
            'tanggal_kegiatan' =>
'required|date',
            'kategori' =>
'required|string|in:berita,pengum
uman,tips,panduan',
            'lokasi' =>
'required|string|max:255',
            'idPenyuluh' =>
'required|exists:users,idUser',
            'foto' =>
'nullable|image|mimes:jpg,jpeg,pn
g|max:10240', // 10MB max
        ], [
            'judul.required' =>
'Judul wajib diisi',
            'judul.max' => 'Judul
maksimal 255 karakter',
            'deskripsi.required'
=> 'Deskripsi wajib diisi',
            'deskripsi.min' =>
'Deskripsi minimal 10 karakter',
            'kategori.required'
=> 'Kategori wajib diisi',
            'kategori.in' =>
'Kategori tidak valid',
            'tanggal_kegiatan.req
uired' => 'Tanggal kegiatan wajib
diisi',
            'tanggal_kegiatan.dat
e' => 'Format tanggal tidak
valid',
            'lokasi.required' =>
'Lokasi wajib diisi',
            'lokasi.max' =>
'Lokasi maksimal 255 karakter',
            'idPenyuluh.required'
=> 'Penyuluh wajib dipilih',
            'idPenyuluh.exists'
=> 'Penyuluh tidak ditemukan',
            'foto.image' => 'File
harus berupa gambar',
            'foto.mimes' =>
'Format foto harus jpg, jpeg,
atau png',
            'foto.max' => 'Ukuran
foto maksimal 10MB',
        ]);

        try {
            $fotoPath = null;

            // Handle file upload
            if ($request-
>hasFile('foto')) {
                $file = $request-
>file('foto');
                $filename =
time() . '_' .
\Illuminate\Support\Str::random(1
0) . '.' . $file-
>getClientOriginalExtension();
                $path = $file-
>storeAs('informasi', $filename,
'public');
                $fotoPath =
$filename;
            }

            // Create informasi
            $informasi =
Informasi::create([
                'judul' =>
$request->input('judul'),
                'deskripsi' =>
$request->input('deskripsi'),
                'tanggal_kegiatan
' => $request-
>input('tanggal_kegiatan'),
                'lokasi' =>
$request->input('lokasi'),
                'kategori' =>
$request->input('kategori'),
                'idPenyuluh' =>
$request->input('idPenyuluh'),
                'foto' =>
$fotoPath,
            ]);

            // Return JSON
response for AJAX request
            if ($request->ajax()
|| $request->wantsJson()) {
                return
response()->json([
                    'success' =>
true,
                    'message' =>
'Informasi berhasil disimpan',
                    'data' =>
$informasi
                ]);
            }

            return redirect()-
>route('admin.informasi.index')-
>with('success', 'Informasi
berhasil disimpan.');
        } catch (\Exception $e) {
            // Delete uploaded
file if exists and there's an
error
            if ($fotoPath &&
Storage::disk('public')-
>exists('informasi/' .
$fotoPath)) {
```

```
                Storage::disk('public')->delete('informasi/' . $fotoPath);
            }

            if ($request->ajax() || $request->wantsJson()) {
                return response()->json([
                    'success' => false,
                    'message' => 'Terjadi kesalahan saat menyimpan data: ' . $e->getMessage()
                ], 500);
            }
        }
    }
    public function destroy($idInformasi)
    {
        try {
            $informasi = Informasi::findOrFail($idInformasi);

            if ($informasi->foto && Storage::disk('public')->exists('informasi/' . $informasi->foto)) {
                Storage::disk('public')->delete('informasi/' . $informasi->foto);
            }

            $informasi->delete();

            if (request()->ajax() || request()->wantsJson()) {
                return response()->json([
                    'success' => true,
                    'message' => 'Informasi berhasil dihapus'
                ]);
            }

            return redirect()->route('admin.informasi')->with('success', 'Informasi berhasil dihapus.');
        } catch (\Exception $e) {
            if (request()->ajax() || request()->wantsJson()) {
                return response()->json([
                    'success' => false,
                    'message' => 'Terjadi kesalahan saat menghapus data: ' . $e->getMessage()
                ], 500);
            }

            return redirect()->back()->with('error', 'Terjadi kesalahan saat menghapus data');
        }
    }
    public function update(Request $request, $idInformasi)
    {
        $informasi = Informasi::findOrFail($idInformasi);

        $request->validate([
            'judul' => 'required|string|max:255',
            'deskripsi' => 'required|string|min:10',
            'tanggal_kegiatan' => 'required|date',
            'kategori' => 'required|string|in:berita,pengumuman,tips,panduan',
            'lokasi' => 'required|string|max:255',
            'idPenyuluh' => 'required|exists:users,idUser',
            'foto' => 'nullable|image|mimes:jpg,jpeg,png|max:10240',
        ], [
            'judul.required' => 'Judul wajib diisi',
            'judul.max' => 'Judul maksimal 255 karakter',
            'deskripsi.required' => 'Deskripsi wajib diisi',
            'deskripsi.min' => 'Deskripsi minimal 10 karakter',
            'kategori.required' => 'Kategori wajib diisi',
            'kategori.in' => 'Kategori tidak valid',
            'tanggal_kegiatan.required' => 'Tanggal kegiatan wajib diisi',
            'tanggal_kegiatan.date' => 'Format tanggal tidak valid',
            'lokasi.required' => 'Lokasi wajib diisi',
            'lokasi.max' => 'Lokasi maksimal 255 karakter',
            'idPenyuluh.required' => 'Penyuluh wajib dipilih',
            'idPenyuluh.exists' => 'Penyuluh tidak ditemukan',
            'foto.image' => 'File harus berupa gambar',
```

```
            'foto.mimes' =>
'Format foto harus jpg, jpeg,
atau png',
            'foto.max' => 'Ukuran
foto maksimal 10MB',
        ]);

        try {
            $fotoPath =
$informasi->foto;

            // Handle file upload
            if ($request-
>hasFile('foto')) {
                // Delete old
file if exists
                if ($fotoPath &&
Storage::disk('public')-
>exists('informasi/' .
$fotoPath)) {
                    Storage::disk
('public')->delete('informasi/' .
$fotoPath);
                }

                $file = $request-
>file('foto');
                $filename =
time() . '_' .
\Illuminate\Support\Str::random(1
0) . '.' . $file-
>getClientOriginalExtension();
                $path = $file-
>storeAs('informasi', $filename,
'public');
                $fotoPath =
$filename;
            }

            // Update informasi
            $informasi->update([
                'judul' =>
$request->input('judul'),
                'deskripsi' =>
$request->input('deskripsi'),
                'tanggal_kegiatan
' => $request-
>input('tanggal_kegiatan'),
                'lokasi' =>
$request->input('lokasi'),
                'kategori' =>
$request->input('kategori'),
                'idPenyuluh' =>
$request->input('idPenyuluh'),
                'foto' =>
$fotoPath,
            ]);

            // Return JSON
response for AJAX request
            if ($request->ajax()
|| $request->wantsJson()) {
                return
response()->json([
                    'success' =>
true,
                    'message' =>
'Informasi berhasil diperbarui',
                    'data' =>
$informasi
                ]);
            }

            return redirect()-
>route('admin.informasi.index')-
>with('success', 'Informasi
berhasil diperbarui.');
        } catch (\Exception $e) {
            if ($request->ajax()
|| $request->wantsJson()) {
                return
response()->json([
                    'success' =>
false,
                    'message' =>
'Terjadi kesalahan saat
memperbarui data: ' . $e-
>getMessage()
                ], 500);
            }

            return redirect()-
>back()
                ->withInput()
                ->with('error',
'Terjadi kesalahan saat
memperbarui data');
        }
    }
    public function
show($idInformasi)
    {
        $informasi =
Informasi::findOrFail($idInformas
i);
        return
view('informasi.detail-
informasi',
compact('informasi'));
    }
}
```

InformasiPakanController.php

```
<?php

namespace App\Http\Controllers;

use App\Models\InformasiPakan;
use Illuminate\Http\Request;
use
Illuminate\Support\Facades\Storag
e;
```

```
class InformasiPakanController
extends Controller
{
    public function store(Request
$request)
    {
        $request->validate([
            'judul' =>
'required|string|max:255',
            'deskripsi' =>
'required|string|min:10',
            'foto' =>
'nullable|image|mimes:jpg,jpeg,pn
g|max:10240', // 10MB max
            'sumber' =>
'nullable|string',
            'jenis_pakan' =>
'required|string|in:hijauan,konse
ntrat,fermentasi,organik,limbah',
            'is_published' =>
'required|integer|in:0,1',

        ], [
            'judul.required' =>
'Judul wajib diisi',
            'judul.max' => 'Judul
maksimal 255 karakter',
            'deskripsi.required'
=> 'Deskripsi wajib diisi',
            'deskripsi.min' =>
'Deskripsi minimal 10 karakter',
            'foto.image' => 'File
harus berupa gambar',
            'foto.mimes' =>
'Format foto harus jpg, jpeg,
atau png',
            'foto.max' => 'Ukuran
foto maksimal 10MB',
            'sumber.string' =>
'Sumber harus berupa teks',
            'is_published.in' =>
'Status publikasi harus benar (0
atau 1)',
        ]);

        try {
            $fotoPath = null;
            if ($request-
>hasFile('foto')) {
                $file = $request-
>file('foto');
                $filename =
time() . '_' .
\Illuminate\Support\Str::random(1
0) . '.' . $file-
>getClientOriginalExtension();
                $path = $file-
>storeAs('informasi-pakan',
$filename, 'public');
                $fotoPath =
$filename;
            }

            $pakan =
InformasiPakan::create([
                'judul' =>
$request->input('judul'),
                'deskripsi' =>
$request->input('deskripsi'),
                'foto' =>
$fotoPath,
                'sumber' =>
$request->input('sumber'),
                'jenis_pakan' =>
$request->input('jenis_pakan'),
                'is_published' =>
$request->input('is_published'),
            ]);

            if ($request->ajax()
|| $request->wantsJson()) {
                return
response()->json([
                    'success' =>
true,
                    'message' =>
'Informasi berhasil disimpan',
                    'data' =>
$pakan
                ]);
            }

            return redirect()-
>route('admin.pakan')-
>with('success', 'Pakan berhasil
disimpan.');
        } catch (\Exception $e) {
            // Delete uploaded
file if exists and there's an
error
            if ($fotoPath &&
Storage::disk('public')-
>exists('pakan/' . $fotoPath)) {
                Storage::disk('pu
blic')->delete('pakan/' .
$fotoPath);
            }
            if ($request->ajax()
|| $request->wantsJson()) {
                return
response()->json([
                    'success' =>
false,
                    'message' =>
'Terjadi kesalahan saat menyimpan
data: ' . $e->getMessage()
                ], 500);
            }
        }
    }
    public function
destroy($idPakan)
    {
        try {
```

```
            $informasi =
InformasiPakan::findOrFail($idPak
an);

            if ($informasi->foto
&& Storage::disk('public')-
>exists('pakan/' . $informasi-
>foto)) {
                Storage::disk('pu
blic')->delete('informasi/' .
$informasi->foto);
            }

            $informasi->delete();

            if (request()->ajax()
|| request()->wantsJson()) {
                return
response()->json([
                    'success' =>
true,
                    'message' =>
'Informasi Pakan berhasil
dihapus'
                ]);
            }

            return redirect()-
>route('admin.pakan')-
>with('success', 'Informasi Pakan
berhasil dihapus.');
        } catch (\Exception $e) {
            if (request()->ajax()
|| request()->wantsJson()) {
                return
response()->json([
                    'success' =>
false,
                    'message' =>
'Terjadi kesalahan saat menghapus
data: ' . $e->getMessage()
                ], 500);
            }

            return redirect()-
>back()->with('error', 'Terjadi
kesalahan saat menghapus data');
        }
    }
    public function
update(Request $request,
$idPakan)
    {
        $informasi =
InformasiPakan::findOrFail($idPak
an);

        $request->validate([
            'judul' =>
'required|string|max:255',
            'deskripsi' =>
'required|string|min:10',
            'foto' =>
'nullable|image|mimes:jpg,jpeg,pn
g|max:10240', // 10MB max
            'sumber' =>
'nullable|string',
            'jenis_pakan' =>
'required|string|in:hijauan,konse
ntrat,fermentasi,organik,limbah',
            'is_published' =>
'required|integer|in:0,1',

        ], [
            'judul.required' =>
'Judul wajib diisi',
            'judul.max' => 'Judul
maksimal 255 karakter',
            'deskripsi.required'
=> 'Deskripsi wajib diisi',
            'deskripsi.min' =>
'Deskripsi minimal 10 karakter',
            'foto.image' => 'File
harus berupa gambar',
            'foto.mimes' =>
'Format foto harus jpg, jpeg,
atau png',
            'foto.max' => 'Ukuran
foto maksimal 10MB',
            'sumber.string' =>
'Sumber harus berupa teks',
            'is_published.in' =>
'Status publikasi harus benar (0
atau 1)',
        ]);

        try {
            $fotoPath =
$informasi->foto;

            // Handle file upload
            if ($request-
>hasFile('foto')) {
                // Delete old
file if exists
                if ($fotoPath &&
Storage::disk('public')-
>exists('pakan/' . $fotoPath)) {
                    Storage::disk
('public')->delete('pakan/' .
$fotoPath);
                }

                $file = $request-
>file('foto');
                $filename =
time() . '_' .
\Illuminate\Support\Str::random(1
0) . '.' . $file-
>getClientOriginalExtension();
                $path = $file-
>storeAs('informasi-pakan',
$filename, 'public');
                $fotoPath =
$filename;
```

```
            }

            $informasi->update([
                'judul' =>
$request->input('judul'),
                'deskripsi' =>
$request->input('deskripsi'),
                'sumber' =>
$request->input('sumber'),
                'jenis_pakan' =>
$request->input('jenis_pakan'),
                'is_published' =>
$request->input('is_published'),
                'foto' =>
$fotoPath,
            ]);

            if ($request->ajax()
|| $request->wantsJson()) {
                return
response()->json([
                    'success' =>
true,
                    'message' =>
'Informasi Pakan berhasil
diperbarui',
                    'data' =>
$informasi
                ]);
            }

            return redirect()-
>route('admin.pakan')-
>with('success', 'Informasi Pakan
berhasil diperbarui.');
        } catch (\Exception $e) {
            if ($request->ajax()
|| $request->wantsJson()) {
                return
response()->json([
                    'success' =>
false,
                    'message' =>
'Terjadi kesalahan saat
memperbarui data: ' . $e-
>getMessage()
                ], 500);
            }

            return redirect()-
>back()
                ->withInput()
                ->with('error',
'Terjadi kesalahan saat
memperbarui data');
        }
    }
    public function index()
    {
        $pakanList =
InformasiPakan::where('is_publish
ed', 1)->paginate(10);

        return
view('pakan.index',
compact('pakanList'));
    }

    public function
show($idPakan)
    {
        $pakan =
InformasiPakan::findOrFail($idPak
an);
        return
view('pakan.detail-pakan',
compact('pakan'));
    }
}
```

PenyuluhController.php

```
<?php

namespace App\Http\Controllers;

use App\Models\Konsultasi;
use App\Models\Laporan;
use App\Models\Ternak;
use Illuminate\Http\Request;
use
Illuminate\Support\Facades\Auth;

class PenyuluhController extends
Controller
{
    public function index()
    {
        $ternaks =
Ternak::with('pemilik')->get();
        $totalTernak =
Ternak::all()->count();
        $ternakSehat =
Ternak::where('status', 'sehat')-
>count();
        $ternakSakit =
Ternak::where('status', 'sakit')-
>count();
        $konsultasiAktif =
Konsultasi::where('status',
'berlangsung')->count();
        return
view('penyuluh.dashboard',
compact('totalTernak',
'ternakSehat', 'ternakSakit',
'konsultasiAktif', 'ternaks'));
    }

    public function profile()
    {
        $userInfo = Auth::user();

        return
view('penyuluh.profile',
compact('userInfo'));
```

```
    }
    public function ternak()
    {
        $ternakList =
Ternak::with('pemilik')->get();
        $totalTernak =
Ternak::all()->count();
        $ternakSehat =
Ternak::where('status', 'sehat')-
>count();
        $ternakSakit =
Ternak::where('status', 'sakit')-
>count();
        $ternakPerawatan =
Ternak::where('status',
'perawatan')->count();
        return
view('penyuluh.ternak',
compact('ternakList',
'totalTernak', 'ternakSehat',
'ternakSakit',
'ternakPerawatan'));
    }
}
```

LaporanController.php

```
<?php

namespace App\Http\Controllers;

use
App\Http\Controllers\Controller;
use App\Models\Laporan;
use App\Models\User;
use App\Models\Ternak;
use Illuminate\Http\Request;
use
Illuminate\Support\Facades\Auth;
use
Illuminate\Support\Facades\DB;
use
Illuminate\Support\Facades\Log;
use Carbon\Carbon;

class LaporanController extends
Controller
{
    public function store(Request
$request)
    {
        try {
            // Debug log untuk
melihat data yang masuk
            Log::info('Laporan
store called', [
                'user_id' =>
Auth::id(),
                'user' =>
Auth::user(),
                'request_data' =>
$request->all()
            ]);

            $validated =
$request->validate([
                'idPeternak' =>
'required|exists:users,idUser',
                'idTernak' =>
'required|exists:ternak,idTernak'
,
                'tanggal_pemeriks
aan' => 'required|date',
                'berat_badan' =>
'nullable|numeric|min:0|max:9999.
99',
                'suhu_tubuh' =>
'required|numeric|min:35|max:45',
                'nafsu_makan' =>
'required|in:baik,menurun,tidak_a
da',
                'pernafasan' =>
'required|in:normal,cepat,lambat,
sesak',
                'kulit_bulu' =>
'required|in:normal,kusam,luka,pa
rasit',
                'mata_hidung' =>
'required|in:normal,berair,bengka
k,bernanah',
                'feses' =>
'required|in:normal,encer,keras,b
erdarah',
                'aktivitas' =>
'required|in:aktif,lesu,gelisah,l
emas',
                'tindakan' =>
'required|string|max:1000',
                'rekomendasi' =>
'required|string|max:1000',
            ], [
                'idPeternak.requi
red' => 'Pilih peternak terlebih
dahulu.',
                'idPeternak.exist
s' => 'Peternak yang dipilih
tidak valid.',
                'idTernak.require
d' => 'Pilih ternak terlebih
dahulu.',
                'idTernak.exists'
=> 'Ternak yang dipilih tidak
valid.',
                'tanggal_pemeriks
aan.required' => 'Tanggal
pemeriksaan harus diisi.',
                'tanggal_pemeriks
aan.date' => 'Format tanggal
pemeriksaan tidak valid.',
                'berat_badan.nume
ric' => 'Berat badan harus berupa
angka.',
                'berat_badan.min'
=> 'Berat badan tidak boleh
kurang dari 0.',
```

```
                'berat_badan.max' => 'Berat badan terlalu besar.',
                'suhu_tubuh.required' => 'Suhu tubuh harus diisi.',
                'suhu_tubuh.numeric' => 'Suhu tubuh harus berupa angka.',
                'suhu_tubuh.min' => 'Suhu tubuh minimal 35°C.',
                'suhu_tubuh.max' => 'Suhu tubuh maksimal 45°C.',
                'nafsu_makan.required' => 'Pilih kondisi nafsu makan.',
                'nafsu_makan.in' => 'Kondisi nafsu makan tidak valid.',
                'pernafasan.required' => 'Pilih kondisi pernafasan.',
                'pernafasan.in' => 'Kondisi pernafasan tidak valid.',
                'kulit_bulu.required' => 'Pilih kondisi kulit & bulu.',
                'kulit_bulu.in' => 'Kondisi kulit & bulu tidak valid.',
                'mata_hidung.required' => 'Pilih kondisi mata & hidung.',
                'mata_hidung.in' => 'Kondisi mata & hidung tidak valid.',
                'feses.required' => 'Pilih kondisi feses.',
                'feses.in' => 'Kondisi feses tidak valid.',
                'aktivitas.required' => 'Pilih kondisi aktivitas.',
                'aktivitas.in' => 'Kondisi aktivitas tidak valid.',
                'tindakan.required' => 'Tindakan yang dilakukan harus diisi.',
                'tindakan.max' => 'Tindakan tidak boleh lebih dari 1000 karakter.',
                'rekomendasi.required' => 'Rekomendasi perawatan harus diisi.',
                'rekomendasi.max' => 'Rekomendasi tidak boleh lebih dari 1000 karakter.',
            ]);

            Log::info('Validation passed', ['validated_data' => $validated]);

            // Verifikasi ternak sesuai dengan peternak
            $ternak = Ternak::where('idTernak', $validated['idTernak'])
                ->where('idPemilik', $validated['idPeternak'])
                ->first();

            if (!$ternak) {
                Log::warning('Ternak tidak sesuai dengan peternak', [
                    'idTernak' => $validated['idTernak'],
                    'idPeternak' => $validated['idPeternak']
                ]);

                return back()->withInput()->withErrors([
                    'idTernak' => 'Ternak yang dipilih tidak sesuai dengan peternak.'
                ]);
            }

            // Evaluasi kondisi kesehatan
            $kondisiKesehatan = $this->evaluateHealthCondition($validated);

            Log::info('Health condition evaluated', ['kondisi' => $kondisiKesehatan]);

            $currentUser = Auth::user();
            $currentUserId = $currentUser->idUser;

            Log::info('Current user info', [
                'user' => $currentUser,
                'idUser' => $currentUserId,
                'auth_id' => Auth::id() // This should return idUser value
            ]);

            // Buat laporan menggunakan Eloquent
            $laporan = Laporan::create([
                'idPenyuluh' => $currentUserId,
                'idPeternak' => $validated['idPeternak'],
```

```
                'idTernak' =>
$validated['idTernak'],
                'tanggal_pemeriks
aan' =>
Carbon::parse($validated['tanggal
_pemeriksaan']),
                'berat_badan' =>
$validated['berat_badan'],
                'suhu_tubuh' =>
$validated['suhu_tubuh'],
                'nafsu_makan' =>
$validated['nafsu_makan'],
                'pernafasan' =>
$validated['pernafasan'],
                'kulit_bulu' =>
$validated['kulit_bulu'],
                'mata_hidung' =>
$validated['mata_hidung'],
                'feses' =>
$validated['feses'],
                'aktivitas' =>
$validated['aktivitas'],
                'tindakan' =>
$validated['tindakan'],
                'rekomendasi' =>
$validated['rekomendasi'],
                'status_kesehatan
' => $kondisiKesehatan,
            ]);

            Log::info('Laporan
created successfully', [
                'laporan_id' =>
$laporan->id,
                'laporan' =>
$laporan->toArray()
            ]);

            // Log aktivitas
            $this-
>logActivity('create_health_repor
t', [
                'laporan_id' =>
$laporan->id,
                'ternak_id' =>
$validated['idTernak'],
                'peternak_id' =>
$validated['idPeternak'],
                'status_kesehatan
' => $kondisiKesehatan,
            ]);

            // Redirect dengan
pesan sukses
            return redirect()-
>route('penyuluh.laporan')
                ->with('success',
'Laporan kesehatan berhasil
disimpan!')
                -
>with('laporan_id', $laporan-
>id);

        } catch
(\Illuminate\Validation\Validatio
nException $e) {
            Log::error('Validatio
n failed', [
                'errors' => $e-
>errors(),
                'request_data' =>
$request->all()
            ]);

            return back()-
>withInput()->withErrors($e-
>errors());
        } catch (\Exception $e) {
            Log::error('Error
saving health report', [
                'error' => $e-
>getMessage(),
                'user_id' =>
Auth::id(),
                'request_data' =>
$request->all(),
                'stack_trace' =>
$e->getTraceAsString()
            ]);

            return back()-
>withInput()->withErrors([
                'error' =>
'Terjadi kesalahan saat menyimpan
laporan: ' . $e->getMessage()
            ]);
        }
    }

    private function
evaluateHealthCondition($data)
    {
        $healthScore = 0;
        $totalParams = 7;

        // Evaluasi suhu tubuh
        if ($data['suhu_tubuh']
>= 37.5 && $data['suhu_tubuh'] <=
39.5) {
            $healthScore++;
        }

        // Evaluasi nafsu makan
        if ($data['nafsu_makan']
=== 'baik') {
            $healthScore++;
        }

        // Evaluasi pernafasan
        if ($data['pernafasan']
=== 'normal') {
            $healthScore++;
        }

        // Evaluasi kulit & bulu
```

```
        if ($data['kulit_bulu']
=== 'normal') {
            $healthScore++;
        }

        // Evaluasi mata & hidung
        if ($data['mata_hidung']
=== 'normal') {
            $healthScore++;
        }

        // Evaluasi feses
        if ($data['feses'] ===
'normal') {
            $healthScore++;
        }

        // Evaluasi aktivitas
        if ($data['aktivitas']
=== 'aktif') {
            $healthScore++;
        }

        // Tentukan status
kesehatan berdasarkan skor
        $healthPercentage =
($healthScore / $totalParams) *
100;

        if ($healthPercentage >=
85) {
            return 'sehat';
        } elseif
($healthPercentage >= 60) {
            return
'perlu_perhatian';
        } else {
            return 'sakit';
        }
    }

    private function
logActivity($action, $data = [])
    {
        try {
            $currentUser =
Auth::user();
            $currentUserId =
$currentUser->idUser;

            DB::table('activity_l
ogs')->insert([
                'user_id' =>
$currentUserId,
                'user_type' =>
'penyuluh',
                'action' =>
$action,
                'data' =>
json_encode($data),
                'ip_address' =>
request()->ip(),
                'user_agent' =>
request()->userAgent(),
                'created_at' =>
now(),
            ]);
        } catch (\Exception $e) {
            Log::warning('Failed
to log activity: ' . $e-
>getMessage());
        }
    }

    public function index()
    {
        try {
            $currentUser =
Auth::user();
            $currentUserId =
$currentUser->idUser;

            $laporans =
Laporan::with(['ternak',
'peternak'])
                -
>byPenyuluh($currentUserId)
                ->latest()
                ->paginate(10);

            $peternaks =
User::where('role', 'peternak')
                -
>select('idUser', 'nama')
                ->get();

            $ternaks =
Ternak::select('idTernak',
'namaTernak', 'jenis',
'idPemilik')->get();

            return
view('penyuluh.laporan',
compact('laporans', 'peternaks',
'ternaks'));
        } catch (\Exception $e) {
            Log::error('Error
loading laporan index', [
                'error' => $e-
>getMessage(),
                'user_id' =>
Auth::id()
            ]);

            return back()-
>withErrors(['error' => 'Terjadi
kesalahan saat memuat data.']);
        }
    }

    public function
getDetail($id)
    {
        try {
```

```
            // Debug: Log the request
            \Log::info('Fetching report detail', ['id' => $id, 'user_id' => Auth::id()]);

            $laporan = Laporan::with(['ternak', 'peternak', 'penyuluh'])
                ->where('id', $id)
                ->byPenyuluh(Auth::id())
                ->first();

            if (!$laporan) {
                \Log::warning('Report not found', ['id' => $id, 'user_id' => Auth::id()]);
                return response()->json([
                    'success' => false,
                    'message' => 'Laporan tidak ditemukan'
                ], 404);
            }

            // Calculate health score
            $healthScore = 0;
            $totalParams = 7;

            // Evaluasi suhu tubuh (normal: 37.5-39.5°C)
            if ($laporan->suhu_tubuh >= 37.5 && $laporan->suhu_tubuh <= 39.5) {
                $healthScore++;
            }

            // Evaluasi parameter lainnya
            if ($laporan->nafsu_makan === 'baik') $healthScore++;
            if ($laporan->pernafasan === 'normal') $healthScore++;
            if ($laporan->kulit_bulu === 'normal') $healthScore++;
            if ($laporan->mata_hidung === 'normal') $healthScore++;
            if ($laporan->feses === 'normal') $healthScore++;
            if ($laporan->aktivitas === 'aktif') $healthScore++;

            // Text mapping untuk display
            $textMappings = [
                'nafsu_makan' => [
                    'baik' => 'Baik',
                    'menurun' => 'Menurun',
                    'tidak_ada' => 'Tidak Ada'
                ],
                'pernafasan' => [
                    'normal' => 'Normal',
                    'cepat' => 'Cepat',
                    'lambat' => 'Lambat',
                    'sesak' => 'Sesak'
                ],
                'kulit_bulu' => [
                    'normal' => 'Normal',
                    'kusam' => 'Kusam',
                    'luka' => 'Ada Luka',
                    'parasit' => 'Ada Parasit'
                ],
                'mata_hidung' => [
                    'normal' => 'Normal',
                    'berair' => 'Berair',
                    'bengkak' => 'Bengkak',
                    'bernanah' => 'Bernanah'
                ],
                'feses' => [
                    'normal' => 'Normal',
                    'encer' => 'Encer/Diare',
                    'keras' => 'Keras',
                    'berdarah' => 'Berdarah'
                ],
                'aktivitas' => [
                    'aktif' => 'Aktif',
                    'lesu' => 'Lesu',
                    'gelisah' => 'Gelisah',
                    'lemas' => 'Lemas'
                ],
                'status_kesehatan' => [
```

```
                    'sehat' =>
'Sehat',
                    'perlu_perhat
ian' => 'Perlu Perhatian',
                    'sakit' =>
'Sakit'
                ]
            ];

            // Format data untuk
modal
            $data = [
                'id' => $laporan-
>id,
                'peternak_name'
=> $laporan->peternak->nama ?? '-
',
                'peternak_address
' => $laporan->peternak->alamat
?? $laporan->peternak->email ??
'-',
                'ternak_name' =>
$laporan->ternak->namaTernak ??
'-',
                'ternak_type' =>
$laporan->ternak->jenis ?? '-',
                'berat_badan' =>
$laporan->berat_badan,
                'tanggal_pemeriks
aan' => $laporan-
>tanggal_pemeriksaan-
>toISOString(),
                'penyuluh_name'
=> $laporan->penyuluh->nama ?? '-
',
                'suhu_tubuh' =>
$laporan->suhu_tubuh,
                'nafsu_makan' =>
$laporan->nafsu_makan,
                'pernafasan' =>
$laporan->pernafasan,
                'kulit_bulu' =>
$laporan->kulit_bulu,
                'mata_hidung' =>
$laporan->mata_hidung,
                'feses' =>
$laporan->feses,
                'aktivitas' =>
$laporan->aktivitas,
                'tindakan' =>
$laporan->tindakan,
                'rekomendasi' =>
$laporan->rekomendasi,
                'status_kesehatan
' => $laporan->status_kesehatan,
                'health_score' =>
$healthScore,
                'health_percentag
e' => round(($healthScore /
$totalParams) * 100),
                'created_at' =>
$laporan->created_at-
>toISOString(),
                // Text versions
untuk display
                'nafsu_makan_text
' =>
$textMappings['nafsu_makan'][$lap
oran->nafsu_makan] ?? $laporan-
>nafsu_makan,
                'pernafasan_text'
=>
$textMappings['pernafasan'][$lapo
ran->pernafasan] ?? $laporan-
>pernafasan,
                'kulit_bulu_text'
=>
$textMappings['kulit_bulu'][$lapo
ran->kulit_bulu] ?? $laporan-
>kulit_bulu,
                'mata_hidung_text
' =>
$textMappings['mata_hidung'][$lap
oran->mata_hidung] ?? $laporan-
>mata_hidung,
                'feses_text' =>
$textMappings['feses'][$laporan-
>feses] ?? $laporan->feses,
                'aktivitas_text'
=>
$textMappings['aktivitas'][$lapor
an->aktivitas] ?? $laporan-
>aktivitas,
                'status_kesehatan
_text' =>
$textMappings['status_kesehatan']
[$laporan->status_kesehatan] ??
$laporan->status_kesehatan,
            ];

            \Log::info('Report
detail fetched successfully',
['id' => $id]);

            return response()-
>json([
                'success' =>
true,
                'data' => $data
            ]);
        } catch (\Exception $e) {
            \Log::error('Error
fetching health report detail: '
. $e->getMessage(), [
                'report_id' =>
$id,
                'user_id' =>
Auth::id(),
                'stack_trace' =>
$e->getTraceAsString()
            ]);

            return response()-
>json([
```

```
                'success' =>
false,
                'message' =>
'Terjadi kesalahan saat memuat
data laporan: ' . $e-
>getMessage()
            ], 500);
        }
    }

    /**
     * Print health report (HTML
view for printing)
     */
    public function
printReport($id)
    {
        try {
            $laporan =
Laporan::with(['ternak',
'peternak', 'penyuluh'])
                ->where('id',
$id)
                -
>byPenyuluh(Auth::id())
                ->first();

            if (!$laporan) {
                return
redirect()-
>route('penyuluh.laporan.index')
                    -
>withErrors(['error' => 'Laporan
tidak ditemukan.']);
            }

            return
view('penyuluh.laporan.print',
compact('laporan'));
        } catch (\Exception $e) {
            \Log::error('Error
printing health report: ' . $e-
>getMessage(), [
                'report_id' =>
$id,
                'user_id' =>
Auth::id()
            ]);

            return redirect()-
>route('penyuluh.laporan.index')
                -
>withErrors(['error' => 'Terjadi
kesalahan saat mencetak
laporan.']);
        }
    }

    /**
     * Export health report as
PDF
     */
    public function
exportPdf($id)
    {
        try {
            $laporan =
Laporan::with(['ternak',
'peternak', 'penyuluh'])
                ->where('id',
$id)
                -
>byPenyuluh(Auth::id())
                ->first();

            if (!$laporan) {
                return
response()->json([
                    'success' =>
false,
                    'message' =>
'Laporan tidak ditemukan'
                ], 404);
            }

            // If you have PDF
library like DomPDF installed
            // $pdf =
PDF::loadView('penyuluh.laporan.p
df', compact('laporan'));
            // return $pdf-
>download("laporan_kesehatan_{$la
poran->id}.pdf");

            // For now, redirect
to print view
            return redirect()-
>route('penyuluh.laporan.print',
$id);
        } catch (\Exception $e) {
            \Log::error('Error
exporting health report as PDF: '
. $e->getMessage(), [
                'report_id' =>
$id,
                'user_id' =>
Auth::id()
            ]);

            return response()-
>json([
                'success' =>
false,
                'message' =>
'Terjadi kesalahan saat
mengexport PDF'
            ], 500);
        }
    }


    public function destroy($id)
    {
        try {
```

```
            \Log::info('Attemptin
g to delete health report', [
                'report_id' =>
$id,
                'user_id' =>
Auth::id()
            ]);

            // Find laporan
dengan validasi ownership
            $laporan =
Laporan::with(['ternak',
'peternak'])
                ->where('id',
$id)
                -
>byPenyuluh(Auth::id()) //
Pastikan hanya penyuluh yang
membuat bisa hapus
                ->first();

            if (!$laporan) {
                \Log::warning('He
alth report not found or
unauthorized', [
                    'report_id'
=> $id,
                    'user_id' =>
Auth::id()
                ]);

                return
response()->json([
                    'success' =>
false,
                    'message' =>
'Laporan tidak ditemukan atau
Anda tidak memiliki akses untuk
menghapus laporan ini.'
                ], 404);
            }

            // Backup info untuk
log sebelum dihapus
            $reportInfo = [
                'id' => $laporan-
>id,
                'ternak_name' =>
$laporan->ternak->namaTernak ??
'Unknown',
                'peternak_name'
=> $laporan->peternak->name ??
'Unknown',
                'tanggal_pemeriks
aan' => $laporan-
>tanggal_pemeriksaan->format('Y-
m-d H:i:s'),
                'status_kesehatan
' => $laporan->status_kesehatan
            ];

            $laporan->delete();

            \Log::info('Health
report deleted successfully', [
                'deleted_report'
=> $reportInfo,
                'deleted_by' =>
Auth::id(),
                'deleted_at' =>
now()
            ]);

            // Log activity
            $this-
>logActivity('delete_health_repor
t', [
                'deleted_report_i
d' => $id,
                'report_info' =>
$reportInfo
            ]);

            return response()-
>json([
                'success' =>
true,
                'message' =>
'Laporan kesehatan berhasil
dihapus.',
                'deleted_report'
=> [
                    'id' =>
$reportInfo['id'],
                    'ternak_name'
=> $reportInfo['ternak_name'],
                    'tanggal_peme
riksaan' =>
$reportInfo['tanggal_pemeriksaan'
]
                ]
            ]);
        } catch (\Exception $e) {
            \Log::error('Error
deleting health report: ' . $e-
>getMessage(), [
                'report_id' =>
$id,
                'user_id' =>
Auth::id(),
                'stack_trace' =>
$e->getTraceAsString()
            ]);

            return response()-
>json([
                'success' =>
false,
                'message' =>
'Terjadi kesalahan saat menghapus
laporan. Silakan coba lagi.'
            ], 500);
        }
    }
```

```
    public function
bulkDestroy(Request $request)
    {
        try {
            $validated =
$request->validate([
                'ids' =>
'required|array|min:1',
                'ids.*' =>
'integer|exists:laporan_kesehatan
,id'
            ]);

            $ids =
$validated['ids'];

            \Log::info('Attemptin
g bulk delete of health reports',
[
                'report_ids' =>
$ids,
                'user_id' =>
Auth::id()
            ]);

            $laporans =
Laporan::with(['ternak',
'peternak'])
                ->whereIn('id',
$ids)
                -
>byPenyuluh(Auth::id())
                ->get();

            if ($laporans-
>count() === 0) {
                return
response()->json([
                    'success' =>
false,
                    'message' =>
'Tidak ada laporan yang dapat
dihapus atau Anda tidak memiliki
akses.'
                ], 404);
            }

            if ($laporans-
>count() !== count($ids)) {
                return
response()->json([
                    'success' =>
false,
                    'message' =>
'Beberapa laporan tidak ditemukan
atau tidak dapat diakses.'
                ], 403);
            }

            // Backup info
sebelum dihapus
            $deletedReports =
$laporans->map(function
($laporan) {
                return [
                    'id' =>
$laporan->id,
                    'ternak_name'
=> $laporan->ternak->namaTernak
?? 'Unknown',
                    'tanggal_peme
riksaan' => $laporan-
>tanggal_pemeriksaan->format('Y-
m-d H:i:s')
                ];
            });

            // Delete all reports
            Laporan::whereIn('id'
, $ids)
                -
>byPenyuluh(Auth::id())
                ->delete();

            \Log::info('Bulk
delete completed successfully', [
                'deleted_count'
=> $laporans->count(),
                'deleted_reports'
=> $deletedReports->toArray(),
                'deleted_by' =>
Auth::id()
            ]);

            // Log activity
            $this-
>logActivity('bulk_delete_health_
reports', [
                'deleted_count'
=> $laporans->count(),
                'deleted_ids' =>
$ids
            ]);

            return response()-
>json([
                'success' =>
true,
                'message' =>
"Berhasil menghapus {$laporans-
>count()} laporan kesehatan.",
                'deleted_count'
=> $laporans->count(),
                'deleted_reports'
=> $deletedReports
            ]);
        } catch
(\Illuminate\Validation\Validatio
nException $e) {
            return response()-
>json([
                'success' =>
false,
```

```
                'message' =>
'Data yang dikirim tidak valid.',
                'errors' => $e-
>errors()
            ], 422);
        } catch (\Exception $e) {
            \Log::error('Error in
bulk delete: ' . $e-
>getMessage(), [
                'request_data' =>
$request->all(),
                'user_id' =>
Auth::id(),
                'stack_trace' =>
$e->getTraceAsString()
            ]);

            return response()-
>json([
                'success' =>
false,
                'message' =>
'Terjadi kesalahan saat menghapus
laporan. Silakan coba lagi.'
            ], 500);
        }
    }

    public function restore($id)
    {
        try {

            $laporan =
Laporan::withTrashed()
                ->where('id',
$id)
                -
>byPenyuluh(Auth::id())
                ->first();

            if (!$laporan) {
                return
response()->json([
                    'success' =>
false,
                    'message' =>
'Laporan tidak ditemukan.'
                ], 404);
            }

            if (!$laporan-
>trashed()) {
                return
response()->json([
                    'success' =>
false,
                    'message' =>
'Laporan tidak dalam kondisi
terhapus.'
                ], 400);
            }

            $laporan->restore();

            \Log::info('Health
report restored successfully', [
                'report_id' =>
$id,
                'restored_by' =>
Auth::id()
            ]);

            return response()-
>json([
                'success' =>
true,
                'message' =>
'Laporan berhasil dipulihkan.'
            ]);
        } catch (\Exception $e) {
            \Log::error('Error
restoring health report: ' . $e-
>getMessage(), [
                'report_id' =>
$id,
                'user_id' =>
Auth::id()
            ]);

            return response()-
>json([
                'success' =>
false,
                'message' =>
'Terjadi kesalahan saat
memulihkan laporan.'
            ], 500);
        }
    }
}
```

KonsultasiController.php

```
<?php

namespace App\Http\Controllers;

use App\Models\Konsultasi;
use App\Models\Ternak;
use App\Models\User;
use Illuminate\Http\Request;
use Illuminate\Support\Carbon;
use
Illuminate\Support\Facades\Auth;
use
Illuminate\Support\Facades\DB;
use
Illuminate\Support\Facades\Log;

class KonsultasiController
extends Controller
{
    public function index()
    {
```

```
        $user = Auth::user();
        $konsultasis =
Konsultasi::with(['penyuluh',
'ternak', 'peternak'])
            ->where('idPeternak',
$user->idUser)
            -
>orderBy('created_at', 'desc')
            ->paginate(10);

        $totalKonsultasi =
Konsultasi::where('idPeternak',
Auth::id())->count();
        $konsultasiBerlangsung =
Konsultasi::where('idPeternak',
Auth::id())->where('status',
'berlangsung')->count();
        $konsultasiSelesai =
Konsultasi::where('idPeternak',
Auth::id())->where('status',
'selesai')->count();
        $konsultasiPending =
Konsultasi::where('idPeternak',
Auth::id())->where('status',
'pending')->count();
        $penyuluhList =
User::where('role', 'Penyuluh')-
>get();
        $ternakList =
Ternak::where('idPemilik',
Auth::id())->get();
        return
view('peternak.konsultasi',
compact('konsultasis',
'totalKonsultasi',
'konsultasiBerlangsung',
'konsultasiSelesai',
'konsultasiPending',
'penyuluhList', 'ternakList'));
    }
    public function store(Request
$request)
    {
        $user = Auth::user();

        if ($user->role !==
'Peternak') {
            return response()-
>json([
                'success' =>
false,
                'message' =>
'Unauthorized'
            ], 403);
        }

        $validated = $request-
>validate([
            'idPeternak'     =>
'required|exists:users,idUser',
            'idPenyuluh'     =>
'required|exists:users,idUser',
            'idTernak'       =>
'required|exists:ternak,idTernak'
,
            'judul_konsultasi' =>
'required|string|max:255',
            'kategori'       =>
'required|in:kesehatan,nutrisi,br
eeding,perawatan,umum',
            'status'         =>
'required|in:pending,berlangsung,
selesai,dibatalkan',
            'deskripsi'      =>
'required|string',
            'foto_ternak'     =>
'nullable|image|mimes:jpg,jpeg,pn
g|max:2048',
        ]);

        $fotoPath = null;
        if ($request-
>hasFile('foto_ternak')) {
            $file = $request-
>file('foto_ternak');
            $filename = time() .
'_' . $file-
>getClientOriginalName();
            $path = $file-
>storeAs('foto_ternak',
$filename, 'public');
            $fotoPath =
$filename;
        }

        $konsultasi =
Konsultasi::create([
            'idPeternak' =>
$validated['idPeternak'],
            'idPenyuluh' =>
$validated['idPenyuluh'],
            'idTernak' =>
$validated['idTernak'],
            'judul_konsultasi' =>
$validated['judul_konsultasi'],
            'kategori' =>
$validated['kategori'],
            'status' =>
$validated['status'],
            'deskripsi' =>
$validated['deskripsi'],
            'foto_ternak' =>
$fotoPath,
        ]);

        if ($request->ajax()) {
            return response()-
>json([
                'success' =>
true,
                'message' =>
'Data Konsultasi berhasil
disimpan.',
                'data' =>
$konsultasi
```

```
            ]);
        }

        return redirect()->route('peternak.konsultasi')->with('success', 'Data konsultasi berhasil disimpan.');
    }

    public function destroy($id)
    {
        $konsultasi = Konsultasi::findOrFail($id);
        $konsultasi->delete();
        return redirect()->route('konsultasi.index')->with('success', 'Konsultasi berhasil dihapus.');
    }

    // Penyuluh Controller

    public function konsultasiEditView($id)
    {
        $konsultasi = Konsultasi::findOrFail($id);
        return view('penyuluh.edit.konsultasi', compact('konsultasi'));
    }

    public function storeResponPenyuluh(Request $request, $id)
    {
        $request->validate([
            'respon' => 'required|string',
        ]);

        $konsultasi = Konsultasi::findOrFail($id);
        $konsultasi->respon = $request->respon;
        $konsultasi->save();

        return redirect()->route('penyuluh.konsultasi')->with('success', 'Respon berhasil ditambahkan.');
    }
    public function getDetail($id)
    {
        try {
            $konsultasi = Konsultasi::with(['peternak', 'ternak', 'penyuluh'])
                ->where('idKonsultasi', $id)
                ->first();

            if (!$konsultasi) {
                return response()->json([
                    'success' => false,
                    'message' => 'Konsultasi tidak ditemukan'
                ], 404);
            }

            // Proses foto ternak
            $fotoTernak = null;
            if ($konsultasi->ternak && $konsultasi->ternak->fotoTernak) {
                $fotoTernak = $konsultasi->ternak->fotoTernak;

                // Pastikan path foto benar
                if (!str_starts_with($fotoTernak, '/')) {
                    $fotoTernak = '/storage/foto_ternak/' . $fotoTernak;
                }
            }

            return response()->json([
                'success' => true,
                'data' => [
                    'idKonsultasi' => $konsultasi->idKonsultasi,
                    'judul_konsultasi' => $konsultasi->judul_konsultasi,
                    'kategori' => $konsultasi->kategori,
                    'deskripsi' => $konsultasi->deskripsi,
                    'status' => $konsultasi->status,
                    'foto_ternak' => $konsultasi->foto_ternak,
                    'created_at' => $konsultasi->created_at->format('d/m/Y H:i'),
                    'peternak' => [
                        'nama' => $konsultasi->peternak->nama ?? '',
                        'email' => $konsultasi->peternak->email ?? '',
                        'phone' => $konsultasi->peternak->phone ?? ''
                    ],
                    'ternak' => [
```

```
                        'nama_ternak' => $konsultasi->ternak->namaTernak ?? '',
                        'jenis' => $konsultasi->ternak->jenis ?? '',
                        'umur' => $konsultasi->ternak->umur ?? null,
                        'fotoTernak' => $fotoTernak // Path foto yang sudah diproses
                    ],
                    'penyuluh' => [
                        'nama' => $konsultasi->penyuluh->nama ?? '',
                        'email' => $konsultasi->penyuluh->email ?? ''
                    ]
                ]
            ]);
        } catch (\Exception $e) {
            Log::error("Error getting konsultasi detail: " . $e->getMessage());

            return response()->json([
                'success' => false,
                'message' => 'Terjadi kesalahan saat mengambil detail konsultasi'
            ], 500);
        }
    }
    public function penyuluhView()
    {
        $konsultasis = Konsultasi::where('idPenyuluh', Auth::id())->get();
        $detailKonsultasi = Konsultasi::with(['peternak', 'ternak'])
            ->where('idPenyuluh', Auth::id())
            ->first();
        if ($detailKonsultasi && $detailKonsultasi->ternak) {
            $tanggalLahir = Carbon::parse($detailKonsultasi->ternak->tanggalLahir);
            $detailKonsultasi->ternak->umur = $tanggalLahir->diffForHumans(null, true); // contoh: "2 tahun 3 bulan"
        }
        $totalKonsultasi = Konsultasi::where('idPenyuluh', Auth::id())->count();
        $totalKonsultasiHariIni = Konsultasi::where('idPenyuluh', Auth::id())->whereDate('created_at', Carbon::today())->count();
        $konsultasiBerlangsung = Konsultasi::where('idPenyuluh', Auth::id())->where('status', 'berlangsung')->count();
        $konsultasiSelesai = Konsultasi::where('idPenyuluh', Auth::id())->where('status', 'selesai')->count();
        $konsultasiPending = Konsultasi::where('idPenyuluh', Auth::id())->where('status', 'pending')->count();
        return view('penyuluh.konsultasi', compact('konsultasis', 'totalKonsultasi', 'konsultasiBerlangsung', 'konsultasiSelesai', 'konsultasiPending', 'totalKonsultasiHariIni', 'detailKonsultasi'));
    }
    public function updateStatus(Request $request, $id)
    {
        try {

            $request->validate([
                'status' => 'required|in:pending,berlangsung,selesai,dibatalkan'
            ]);

            $konsultasi = Konsultasi::where('idKonsultasi', $id)->first();

            if (!$konsultasi) {
                return response()->json([
                    'success' => false,
                    'message' => 'Konsultasi tidak ditemukan'
                ], 404);
            }

            if (auth()->user()->idUser !== $konsultasi->idPenyuluh && auth()->user()->role !== 'admin') {
                return response()->json([
```

```
                    'success' =>
false,
                    'message' =>
'Anda tidak memiliki akses untuk
mengubah status konsultasi ini'
                ], 403);
            }

            $oldStatus =
$konsultasi->status;
            $newStatus =
$request->status;

            $validTransitions =
$this-
>getValidStatusTransitions($oldSt
atus);

            if
(!in_array($newStatus,
$validTransitions)) {
                return
response()->json([
                    'success' =>
false,
                    'message' =>
"Tidak dapat mengubah status dari
'{$oldStatus}' menjadi
'{$newStatus}'"
                ], 400);
            }

            // Update status
            DB::beginTransaction(
);

            $konsultasi->status =
$newStatus;

            if ($newStatus ===
'berlangsung' && !$konsultasi-
>idPenyuluh) {
                $konsultasi-
>idPenyuluh = auth()->user()-
>idUser;
            }

            $konsultasi->save();

            Log::info("Status
konsultasi {$id} diubah dari
{$oldStatus} menjadi {$newStatus}
oleh user " . auth()->user()-
>idUser);

            DB::commit();

            return response()-
>json([
                'success' =>
true,
                'message' =>
"Status konsultasi berhasil
diubah menjadi '{$newStatus}'",
                'data' => [
                    'id' =>
$konsultasi->idKonsultasi,
                    'old_status'
=> $oldStatus,
                    'new_status'
=> $newStatus,
                    'updated_at'
=> $konsultasi->updated_at-
>format('Y-m-d H:i:s')
                ]
            ]);
        } catch
(\Illuminate\Validation\Validatio
nException $e) {
            return response()-
>json([
                'success' =>
false,
                'message' =>
'Data tidak valid',
                'errors' => $e-
>errors()
            ], 422);
        } catch (\Exception $e) {
            DB::rollback();

            Log::error("Error
updating konsultasi status: " .
$e->getMessage());

            return response()-
>json([
                'success' =>
false,
                'message' =>
'Terjadi kesalahan sistem: ' .
$e->getMessage()
            ], 500);
        }
    }
    private function
getValidStatusTransitions($curren
tStatus)
    {
        $transitions = [
            'pending' =>
['berlangsung', 'dibatalkan'],
            'berlangsung' =>
['selesai', 'dibatalkan'],
            'selesai' => [],
            'dibatalkan' =>
['pending']
        ];

        return
$transitions[$currentStatus] ??
[];
    }
}
```

PakanController.php

```
<?php

namespace App\Http\Controllers;

use App\Models\Pakan;
use App\Models\Ternak;
use Illuminate\Http\Request;
use
Illuminate\Support\Facades\Auth;

class PakanController extends
Controller
{
    public function index()
    {
        $ternaks =
Ternak::with('pemilik')->get();
        $pakans =
Pakan::with('ternak', 'penyuluh')
            -
>orderBy('tanggalRekomendasi',
'desc')
            ->paginate(12);
        return
view('penyuluh.pakan',
compact('pakans', 'ternaks'));
    }

    public function
pakanStoreView()
    {
        $ternaks =
Ternak::with('pemilik')->get();
        return
view('penyuluh.tambah.pakan',
compact('ternaks'));
    }

    public function store(Request
$request)
    {

        $request->validate([
            'idTernak' =>
'required|exists:ternak,idTernak'
,
            'tanggalRekomendasi'
=>
'required|date|after_or_equal:tod
ay',
            'jenisPakan' =>
'required|string|max:255',
            'kategori' =>
'required|in:hijauan,konsentrat,s
uplemen,vitamin',
            'jumlah' =>
'required|numeric|min:0.1',
            'satuan' =>
'required|string|max:50',
            'deskripsi' =>
'required|string|min:10',
        ], [
            'idTernak.required'
=> 'Pilih ternak yang akan diberi
rekomendasi pakan.',
            'idTernak.exists' =>
'Ternak yang dipilih tidak
valid.',
            'tanggalRekomendasi.r
equired' => 'Tanggal rekomendasi
harus diisi.',
            'tanggalRekomendasi.d
ate' => 'Format tanggal tidak
valid.',
            'tanggalRekomendasi.a
fter_or_equal' => 'Tanggal
rekomendasi tidak boleh kurang
dari hari ini.',
            'jenisPakan.required'
=> 'Jenis pakan harus diisi.',
            'jenisPakan.max' =>
'Jenis pakan maksimal 255
karakter.',
            'kategori.required'
=> 'Kategori pakan harus
dipilih.',
            'kategori.in' =>
'Kategori pakan harus salah satu
dari: hijauan, konsentrat,
suplemen, vitamin.',
            'jumlah.required' =>
'Jumlah pakan harus diisi.',
            'jumlah.numeric' =>
'Jumlah pakan harus berupa
angka.',
            'jumlah.min' =>
'Jumlah pakan minimal 0.1.',
            'satuan.required' =>
'Satuan pakan harus diisi.',
            'satuan.max' =>
'Satuan pakan maksimal 50
karakter.',
            'deskripsi.required'
=> 'Deskripsi cara pemberian
harus diisi.',
            'deskripsi.min' =>
'Deskripsi minimal 10 karakter.',
        ]);

        try {

            $pakan =
Pakan::create([
                'idTernak' =>
$request->idTernak,
                'idPenyuluh' =>
Auth::id(),
                'tanggalRekomenda
si' => $request-
>tanggalRekomendasi,
                'jenisPakan' =>
$request->jenisPakan,
                'kategori' =>
$request->kategori,
```

```
                'jumlah' =>
$request->jumlah,
                'satuan' =>
$request->satuan,
                'deskripsi' =>
$request->deskripsi,
            ]);

            return redirect()-
>route('penyuluh.pakan')
                ->with('success',
'Rekomendasi pakan berhasil
disimpan.');
        } catch (\Exception $e) {
            return redirect()-
>back()
                ->withInput()
                ->with('error',
'Gagal menyimpan rekomendasi
pakan. Silakan coba lagi.');
        }
    }

    public function
pakanDetailView($id)
    {
        $pakan =
Pakan::findOrFail($id);
        return
view('penyuluh.detail.pakan',
compact('pakan'));
    }

    public function
pakanEditView($id)
    {
        $pakan =
Pakan::findOrFail($id);
        $ternaks =
Ternak::with('pemilik')->get();
        return
view('penyuluh.edit.pakan',
compact('ternaks', 'pakan'));
    }

    public function edit(Request
$request, $id)
    {
        $pakans =
Pakan::findOrFail($id);

        $request->validate([
            'idTernak' =>
'required|exists:ternak,idTernak'
,
            'tanggalRekomendasi'
=> 'required|date',
            'jenisPakan' =>
'required|string',
            'jumlah' =>
'required|string',
            'saran' =>
'required|string',
        ]);

        $pakans->update([
            'idTernak' =>
$request->idTernak,
            'idPenyuluh' =>
Auth::id(),
            'tanggalRekomendasi'
=> $request->tanggalRekomendasi,
            'jenisPakan' =>
$request->jenisPakan,
            'jumlah' => $request-
>jumlah,
            'saran' => $request-
>saran,
        ]);

        return redirect()-
>route('penyuluh.pakan')-
>with('success', 'Pakan berhasil
diupdate.');
    }


    public function destroy($id)
    {
        try {
            $pakan =
Pakan::findOrFail($id);
            $pakanName = $pakan-
>jenisPakan; // Store name before
deletion

            $pakan->delete();


            if (request()-
>expectsJson()) {
                return
response()->json([
                    'success' =>
true,
                    'message' =>
"Rekomendasi pakan '{$pakanName}'
berhasil dihapus"
                ], 200);
            }


            return redirect()-
>route('penyuluh.pakan')
                ->with('success',
"Rekomendasi pakan '{$pakanName}'
berhasil dihapus");
        } catch
(\Illuminate\Database\Eloquent\Mo
delNotFoundException $e) {
            if (request()-
>expectsJson()) {
                return
response()->json([
```

```
                    'success' =>
false,
                    'message' =>
'Rekomendasi pakan tidak
ditemukan'
                ], 404);
            }

            return redirect()-
>route('penyuluh.pakan')
                ->with('error',
'Rekomendasi pakan tidak
ditemukan');
        } catch (\Exception $e) {
            if (request()-
>expectsJson()) {
                return
response()->json([
                    'success' =>
false,
                    'message' =>
'Terjadi kesalahan saat menghapus
rekomendasi pakan'
                ], 500);
            }

            return redirect()-
>route('penyuluh.pakan')
                ->with('error',
'Terjadi kesalahan saat menghapus
rekomendasi pakan');
        }
    }
}
```

PeternakController.php

```
<?php

namespace App\Http\Controllers;

use App\Models\Konsultasi;
use App\Models\Pakan;
use App\Models\Ternak;
use Illuminate\Http\Request;
use
Illuminate\Support\Facades\Auth;

class PeternakController extends
Controller
{
    public function index()
    {
        $userId = Auth::user()-
>idUser;
        $ternak =
Ternak::where('idPemilik',
$userId)->first();
        $totalTernak =
Ternak::where('idPemilik',
$userId)->count();
        $ternakSehat =
Ternak::where('idPemilik',
Auth::user()->idUser)-
>where('status', 'sehat')-
>count();
        $ternakSakit =
Ternak::where('idPemilik',
Auth::user()->idUser)-
>where('status', 'sakit')-
>count();
        $konsultasiSaya =
Konsultasi::where('idPeternak',
Auth::user()->idUser)->count();
        $recentTernakList =
Ternak::where('idPemilik',
Auth::user()->idUser)-
>orderBy('created_at', 'desc')-
>take(5)->get();
        return
view('peternak.dashboard',
compact('ternak', 'totalTernak',
'ternakSehat', 'ternakSakit',
'konsultasiSaya',
'recentTernakList'));
    }

    public function kesehatan()
    {
        return
view('peternak.kesehatan');
    }

    public function konsultasi()
    {
        return
view('peternak.konsultasi');
    }
    public function laporan()
    {
        return
view('peternak.laporan');
    }
    public function profile()
    {
        $userInfo = Auth::user();
        $ternakSaya =
Ternak::where('idPemilik',
Auth::user()->idUser)->count();
        $konsultasiSaya =
Konsultasi::where('idPeternak',
Auth::user()->idUser)->count();
        return
view('peternak.profile',
compact('userInfo', 'ternakSaya',
'konsultasiSaya'));
    }

    public function pakan(Request
$request)
    {
        $query =
Pakan::with(['ternak',
'penyuluh']);
```

```
        if ($request->filled('kategori')) {
            $query->where('kategori', $request->kategori);
        }

        if ($request->filled('search')) {
            $search = $request->search;
            $query->where(function ($q) use ($search) {
                $q->where('jenisPakan', 'like', '%' . $search . '%')
                    ->orWhere('deskripsi', 'like', '%' . $search . '%')
                    ->orWhereHas('ternak', function ($query) use ($search) {
                        $query->where('nama_ternak', 'like', '%' . $search . '%');
                    });
            });
        }

        $pakans = $query->orderBy('tanggalRekomendasi', 'desc')->paginate(12);
        return view('peternak.pakan', compact('pakans'));
    }
}
```

TernakController.php

```
<?php

namespace App\Http\Controllers;

use App\Models\Konsultasi;
use Illuminate\Http\Request;
use App\Models\Ternak;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Storage;

class TernakController extends Controller
{
    /**
     * Display listing of ternak for authenticated peternak
     */
    public function index()
    {
        $ternakList = Ternak::where('idPemilik', Auth::user()->idUser)->get();
        $totalTernak = $ternakList->count();
        $ternakSehat = Ternak::where('idPemilik', Auth::user()->idUser)->where('status', 'sehat')->count();
        $ternakSakit = Ternak::where('idPemilik', Auth::user()->idUser)->where('status', 'sakit')->count();
        $ternakPerawatan = Ternak::where('idPemilik', Auth::user()->idUser)->where('status', 'perawatan')->count();
        $konsultasiSaya = Konsultasi::where('idPeternak', Auth::user()->idUser)->count();

        return view('peternak.ternak', compact(
            'ternakList',
            'totalTernak',
            'ternakSehat',
            'ternakSakit',
            'ternakPerawatan',
            'konsultasiSaya'
        ));
    }

    /**
     * Show the form for creating a new ternak
     */
    public function create()
    {
        return view('peternak.ternak.tambah');
    }

    /**
     * Store a newly created ternak in storage
     */
    public function store(Request $request)
    {
        $user = Auth::user();

        if ($user->role !== 'Peternak') {
            return response()->json([
                'success' => false,
```

```
                'message' =>
'Unauthorized'
            ], 403);
        }

        $validated = $request-
>validate([
            'namaTernak'     =>
'required|string|max:100',
            'jenis'          =>
'required|string|max:50',
            'jenis_kelamin'  =>
'required|in:Jantan,Betina',
            'tanggalLahir'   =>
'nullable|date',
            'berat'          =>
'nullable|numeric|min:0',
            'status'         =>
'required|in:sehat,sakit,perawata
n',
            'asal'           =>
'nullable|string|max:100',
            'keterangan'     =>
'nullable|string|max:255',
            'fotoTernak'     =>
'nullable|image|mimes:jpg,jpeg,pn
g|max:2048',
        ]);

        $fotoPath = null;
        if ($request-
>hasFile('fotoTernak')) {
            $file = $request-
>file('fotoTernak');
            $filename = time() .
'_' . $file-
>getClientOriginalName();
            $path = $file-
>storeAs('foto_ternak',
$filename, 'public');
            $fotoPath =
$filename;
        }

        $ternak =
Ternak::create([
            'idPemilik'      =>
$user->idUser,
            'namaTernak'     =>
$validated['namaTernak'],
            'jenis'          =>
$validated['jenis'],
            'jenis_kelamin'  =>
$validated['jenis_kelamin'],
            'tanggalLahir'   =>
$validated['tanggalLahir'],
            'berat'          =>
$validated['berat'],
            'status'         =>
$validated['status'],
            'asal'           =>
$validated['asal'],
            'keterangan'     =>
$validated['keterangan'],
            'fotoTernak'     =>
$fotoPath,
        ]);

        if ($request->ajax()) {
            return response()-
>json([
                'success' =>
true,
                'message' =>
'Data ternak berhasil disimpan.',
                'data' => $ternak
            ]);
        }

        return redirect()-
>route('peternak.ternak')-
>with('success', 'Data ternak
berhasil disimpan.');
    }

    /**
     * Display the specified
ternak
     */
    public function show($id)
    {
        $user = Auth::user();
        $ternak =
Ternak::where('idTernak', $id)
            ->where('idPemilik',
$user->idUser)
            ->firstOrFail();

        return
view('peternak.ternak.detail',
compact('ternak'));
    }

    /**
     * Show the form for editing
the specified ternak
     */
    public function edit($id)
    {
        $user = Auth::user();
        $ternak =
Ternak::where('idTernak', $id)
            ->where('idPemilik',
$user->idUser)
            ->firstOrFail();

        return
view('peternak.ternak.edit',
compact('ternak'));
    }

    /**
     * Update the specified
ternak in storage
     */
```

```
    public function
update(Request $request, $id)
    {
        $user = Auth::user();

        if ($user->role !==
'Peternak') {
            return response()-
>json([
                'success' =>
false,
                'message' =>
'Unauthorized'
            ], 403);
        }

        $ternak =
Ternak::where('idTernak', $id)
            ->where('idPemilik',
$user->idUser)
            ->firstOrFail();

        $validated = $request-
>validate([
            'namaTernak'     =>
'required|string|max:100',
            'jenis'          =>
'required|string|max:50',
            'jenis_kelamin'  =>
'required|in:Jantan,Betina',
            'tanggalLahir'   =>
'nullable|date',
            'berat'          =>
'nullable|numeric|min:0',
            'status'         =>
'required|in:sehat,sakit,perawata
n',
            'asal'           =>
'nullable|string|max:100',
            'keterangan'     =>
'nullable|string|max:255',
            'fotoTernak'     =>
'nullable|image|mimes:jpg,jpeg,pn
g|max:2048',
        ]);

        // Handle foto upload
        if ($request-
>hasFile('fotoTernak')) {
            // Delete old photo
if exists
            if ($ternak-
>fotoTernak &&
Storage::disk('public')-
>exists('foto_ternak/' . $ternak-
>fotoTernak)) {
                Storage::disk('pu
blic')->delete('foto_ternak/' .
$ternak->fotoTernak);
            }

            // Upload new photo
            $file = $request-
>file('fotoTernak');
            $filename = time() .
'_' . $file-
>getClientOriginalName();
            $path = $file-
>storeAs('foto_ternak',
$filename, 'public');
            $validated['fotoTerna
k'] = $filename;
        }

        $ternak-
>update($validated);

        if ($request->ajax()) {
            return response()-
>json([
                'success' =>
true,
                'message' =>
'Data ternak berhasil diupdate.',
                'data' =>
$ternak->fresh()
            ]);
        }

        return redirect()-
>route('peternak.ternak')-
>with('success', 'Data ternak
berhasil diupdate.');
    }

    /**
     * Remove the specified
ternak from storage
     */
    public function destroy($id)
    {
        try {
            $user = Auth::user();
            $ternak =
Ternak::where('idTernak', $id)
                -
>where('idPemilik', $user-
>idUser)
                ->firstOrFail();

            // Delete photo if
exists
            if ($ternak-
>fotoTernak &&
Storage::disk('public')-
>exists('foto_ternak/' . $ternak-
>fotoTernak)) {
                Storage::disk('pu
blic')->delete('foto_ternak/' .
$ternak->fotoTernak);
            }

            $ternak->delete();
```

```
            if (request()->ajax()) {
                return response()->json([
                    'success' => true,
                    'message' => 'Data ternak berhasil dihapus.'
                ]);
            }

            return redirect()->route('peternak.ternak')->with('success', 'Data ternak berhasil dihapus.');
        } catch (\Exception $e) {
            if (request()->ajax()) {
                return response()->json([
                    'success' => false,
                    'message' => 'Gagal menghapus data ternak: ' . $e->getMessage()
                ], 500);
            }

            return redirect()->route('peternak.ternak')->with('error', 'Gagal menghapus data ternak.');
        }
    }

    /**
     * Get ternak data for AJAX requests
     */
    public function getTernakData($id)
    {
        try {
            $user = Auth::user();
            $ternak = Ternak::where('idTernak', $id)
                ->where('idPemilik', $user->idUser)
                ->firstOrFail();

            return response()->json([
                'success' => true,
                'data' => $ternak
            ]);
        } catch (\Exception $e) {
            return response()->json([
                'success' => false,
                'message' => 'Data ternak tidak ditemukan.'
            ], 404);
        }
    }

    /**
     * Export ternak data (for future implementation)
     */
    public function export()
    {
        // TODO: Implement export functionality
        return response()->json([
            'success' => false,
            'message' => 'Fitur export belum tersedia.'
        ]);
    }
}
```

Models
Informasi.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Informasi extends Model
{

    protected $primaryKey = 'idInformasi';

    protected $fillable = [
        'judul',
        'deskripsi',
        'tanggal_kegiatan',
        'lokasi',
        'kategori',
        'idPenyuluh',
        'foto'
    ];

}
```

InformasiPakan.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class InformasiPakan extends Model
```

```
{
    protected $table = 'pakan';

    protected $primaryKey =
'idPakan';

    protected $fillable = [
        'idPakan',
        'judul',
        'deskripsi',
        'foto',
        'sumber',
        'jenis_pakan',
        'is_published',
        'created_at',
        'updated_at'
    ];
}
```

Konsultasi.php

```
<?php

namespace App\Models;

use
Illuminate\Database\Eloquent\Mode
l;

class Konsultasi extends Model
{
    protected $table =
'konsultasi_kesehatan';

    protected $primaryKey =
'idKonsultasi';

    protected $fillable = [
        'idPeternak',
        'idPenyuluh',
        'idTernak',
        'judul_konsultasi',
        'kategori',
        'deskripsi',
        'status',
        'foto_ternak',
    ];
    public function penyuluh()
    {
        return $this-
>belongsTo(User::class,
'idPenyuluh', 'idUser');
    }

    public function ternak()
    {
        return $this-
>belongsTo(Ternak::class,
'idTernak', 'idTernak');
    }
    public function peternak()
    {
        return $this-
>belongsTo(User::class,
'idPeternak', 'idUser');
    }
}
```

Laporan.php

```
<?php

namespace App\Models;

use
Illuminate\Database\Eloquent\Fact
ories\HasFactory;
use
Illuminate\Database\Eloquent\Mode
l;
use Carbon\Carbon;

class Laporan extends Model
{
    use HasFactory;

    protected $table =
'laporan_kesehatan';

    protected $fillable = [
        'idPenyuluh',
        'idPeternak',
        'idTernak',
        'tanggal_pemeriksaan',
        'berat_badan',
        'suhu_tubuh',
        'nafsu_makan',
        'pernafasan',
        'kulit_bulu',
        'mata_hidung',
        'feses',
        'aktivitas',
        'tindakan',
        'rekomendasi',
        'status_kesehatan'
    ];

    protected $casts = [
        'tanggal_pemeriksaan' =>
'datetime',
        'berat_badan' =>
'decimal:2',
        'suhu_tubuh' =>
'decimal:1',
    ];

    /**
     * Relationship dengan User
(Penyuluh)
     * Foreign key: idPenyuluh ->
references: idUser on users table
     */
    public function penyuluh()
    {
        return $this-
>belongsTo(User::class,
'idPenyuluh', 'idUser');
```

```
    }

    /**
     * Relationship dengan User
(Peternak)
     * Foreign key: idPeternak ->
references: idUser on users table
     */
    public function peternak()
    {
        return $this-
>belongsTo(User::class,
'idPeternak', 'idUser');
    }

    /**
     * Relationship dengan Ternak
     * Foreign key: idTernak ->
references: idTernak on ternaks
table
     */
    public function ternak()
    {
        return $this-
>belongsTo(Ternak::class,
'idTernak', 'idTernak');
    }

    /**
     * Scope untuk filter
berdasarkan penyuluh
     */
    public function
scopeByPenyuluh($query,
$penyuluhId)
    {
        return $query-
>where('idPenyuluh',
$penyuluhId);
    }

    /**
     * Scope untuk filter
berdasarkan peternak
     */
    public function
scopeByPeternak($query,
$peternakId)
    {
        return $query-
>where('idPeternak',
$peternakId);
    }

    /**
     * Scope untuk filter
berdasarkan ternak
     */
    public function
scopeByTernak($query, $ternakId)
    {
        return $query-
>where('idTernak', $ternakId);
    }

    /**
     * Scope untuk filter
berdasarkan status kesehatan
     */
    public function
scopeByStatus($query, $status)
    {
        return $query-
>where('status_kesehatan',
$status);
    }

    /**
     * Scope untuk filter
berdasarkan periode
     */
    public function
scopeByPeriod($query, $startDate,
$endDate)
    {
        return $query-
>whereBetween('tanggal_pemeriksaa
n', [$startDate, $endDate]);
    }

    /**
     * Scope untuk laporan
terbaru
     */
    public function
scopeLatest($query)
    {
        return $query-
>orderBy('tanggal_pemeriksaan',
'desc');
    }

    /**
     * Accessor untuk format
tanggal pemeriksaan
     */
    public function
getFormattedTanggalPemeriksaanAtt
ribute()
    {
        return $this-
>tanggal_pemeriksaan->format('d F
Y, H:i');
    }

    /**
     * Accessor untuk format
tanggal pemeriksaan pendek
     */
    public function
getShortTanggalPemeriksaanAttribu
te()
    {
        return $this-
>tanggal_pemeriksaan-
>format('d/m/Y');
```

```
    }

    /**
     * Accessor untuk status
kesehatan dalam bahasa Indonesia
     */
    public function
getStatusKesehatanTextAttribute()
    {
        $statuses = [
            'sehat' => 'Sehat',
            'perlu_perhatian' =>
'Perlu Perhatian',
            'sakit' => 'Sakit'
        ];

        return $statuses[$this-
>status_kesehatan] ?? 'Tidak
Diketahui';
    }

    /**
     * Accessor untuk CSS class
status kesehatan
     */
    public function
getStatusKesehatanClassAttribute(
)
    {
        $classes = [
            'sehat' => 'bg-green-
100 text-green-800',
            'perlu_perhatian' =>
'bg-yellow-100 text-yellow-800',
            'sakit' => 'bg-red-
100 text-red-800'
        ];

        return $classes[$this-
>status_kesehatan] ?? 'bg-gray-
100 text-gray-800';
    }

    /**
     * Accessor untuk nafsu makan
dalam bahasa Indonesia
     */
    public function
getNafsuMakanTextAttribute()
    {
        $texts = [
            'baik' => 'Baik',
            'menurun' =>
'Menurun',
            'tidak_ada' => 'Tidak
Ada'
        ];

        return $texts[$this-
>nafsu_makan] ?? 'Tidak
Diketahui';
    }

    /**
     * Accessor untuk pernafasan
dalam bahasa Indonesia
     */
    public function
getPernafasanTextAttribute()
    {
        $texts = [
            'normal' => 'Normal',
            'cepat' => 'Cepat',
            'lambat' => 'Lambat',
            'sesak' => 'Sesak'
        ];

        return $texts[$this-
>pernafasan] ?? 'Tidak
Diketahui';
    }

    /**
     * Accessor untuk kulit bulu
dalam bahasa Indonesia
     */
    public function
getKulitBuluTextAttribute()
    {
        $texts = [
            'normal' => 'Normal',
            'kusam' => 'Kusam',
            'luka' => 'Ada Luka',
            'parasit' => 'Ada
Parasit'
        ];

        return $texts[$this-
>kulit_bulu] ?? 'Tidak
Diketahui';
    }

    /**
     * Accessor untuk mata hidung
dalam bahasa Indonesia
     */
    public function
getMataHidungTextAttribute()
    {
        $texts = [
            'normal' => 'Normal',
            'berair' => 'Berair',
            'bengkak' =>
'Bengkak',
            'bernanah' =>
'Bernanah'
        ];

        return $texts[$this-
>mata_hidung] ?? 'Tidak
Diketahui';
    }

    /**
     * Accessor untuk feses dalam
bahasa Indonesia
```

```
     */
    public function
getFesesTextAttribute()
    {
        $texts = [
            'normal' => 'Normal',
            'encer' => 'Encer',
            'keras' => 'Keras',
            'berdarah' =>
'Berdarah'
        ];

        return $texts[$this-
>feses] ?? 'Tidak Diketahui';
    }

    /**
     * Accessor untuk aktivitas
dalam bahasa Indonesia
     */
    public function
getAktivitasTextAttribute()
    {
        $texts = [
            'aktif' => 'Aktif',
            'lesu' => 'Lesu',
            'gelisah' =>
'Gelisah',
            'lemas' => 'Lemas'
        ];

        return $texts[$this-
>aktivitas] ?? 'Tidak Diketahui';
    }

    /**
     * Method untuk mendapatkan
indikator suhu
     */
    public function
getSuhuIndicatorAttribute()
    {
        if ($this->suhu_tubuh >=
37.5 && $this->suhu_tubuh <=
39.5) {
            return [
                'status' =>
'normal',
                'text' =>
'Normal',
                'class' => 'text-
green-600'
            ];
        } elseif ($this-
>suhu_tubuh < 37.5) {
            return [
                'status' =>
'rendah',
                'text' => 'Rendah
(Hipotermia)',
                'class' => 'text-
blue-600'
            ];
        } else {
            return [
                'status' =>
'tinggi',
                'text' => 'Tinggi
(Demam)',
                'class' => 'text-
red-600'
            ];
        }
    }

    /**
     * Method untuk menghitung
skor kesehatan
     */
    public function
getHealthScoreAttribute()
    {
        $score = 0;
        $maxScore = 7;

        // Suhu normal
        if ($this->suhu_tubuh >=
37.5 && $this->suhu_tubuh <=
39.5) $score++;

        // Kondisi normal lainnya
        if ($this->nafsu_makan
=== 'baik') $score++;
        if ($this->pernafasan ===
'normal') $score++;
        if ($this->kulit_bulu ===
'normal') $score++;
        if ($this->mata_hidung
=== 'normal') $score++;
        if ($this->feses ===
'normal') $score++;
        if ($this->aktivitas ===
'aktif') $score++;

        return [
            'score' => $score,
            'max_score' =>
$maxScore,
            'percentage' =>
round(($score / $maxScore) * 100)
        ];
    }

    /**
     * Static method untuk
statistik laporan
     */
    public static function
getStatistics($penyuluhId = null,
$period = null)
    {
        $query = self::query();

        if ($penyuluhId) {
```

```
            $query->where('idPenyuluh', $penyuluhId);
        }

        if ($period) {
            $query->where('tanggal_pemeriksaan', '>=', Carbon::parse($period));
        }

        return [
            'total_laporan' => $query->count(),
            'sehat' => $query->where('status_kesehatan', 'sehat')->count(),
            'perlu_perhatian' => $query->where('status_kesehatan', 'perlu_perhatian')->count(),
            'sakit' => $query->where('status_kesehatan', 'sakit')->count(),
            'laporan_bulan_ini' => $query->whereMonth('tanggal_pemeriksaan', now()->month)
                                         ->whereYear('tanggal_pemeriksaan', now()->year)
                                         ->count()
        ];
    }

    /**
     * Method untuk export data
     */
    public function toExportArray()
    {
        return [
            'ID' => $this->id,
            'Tanggal Pemeriksaan' => $this->formatted_tanggal_pemeriksaan,
            'Nama Peternak' => $this->peternak->nama ?? '-',
            'Nama Ternak' => $this->ternak->namaTernak ?? '-',
            'Jenis Ternak' => $this->ternak->jenis ?? '-',
            'Berat Badan (kg)' => $this->berat_badan ?? '-',
            'Suhu Tubuh (°C)' => $this->suhu_tubuh,
            'Nafsu Makan' => $this->nafsu_makan_text,
            'Pernafasan' => $this->pernafasan_text,
            'Kulit & Bulu' => $this->kulit_bulu_text,
            'Mata & Hidung' => $this->mata_hidung_text,
            'Feses' => $this->feses_text,
            'Aktivitas' => $this->aktivitas_text,
            'Tindakan' => $this->tindakan,
            'Rekomendasi' => $this->rekomendasi,
            'Status Kesehatan' => $this->status_kesehatan_text,
            'Penyuluh' => $this->penyuluh->nama ?? '-'
        ];
    }
}
```

Pakan.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Pakan extends Model
{
    protected $table = 'rekomendasi_pakan';

    protected $primaryKey = 'idRekomendasi';

    protected $fillable = [
        'idTernak',
        'idPenyuluh',
        'tanggalRekomendasi',
        'jenisPakan',
        'kategori',
        'jumlah',
        'satuan',
        'deskripsi',
    ];

    public function ternak()
    {
        return $this->belongsTo(Ternak::class, 'idTernak');
    }

    public function penyuluh()
    {
        return $this->belongsTo(User::class, 'idPenyuluh');
    }
}
```

Perawatan.php

```
<?php
```

```
namespace App\Models;

use
Illuminate\Database\Eloquent\Mode
l;

class Perawatan extends Model
{
    protected $table =
'perawatan';

    protected $primaryKey =
'idPerawatan';

    protected $fillable = [
        'idTernak',
        'idPenyuluh',
        'tanggalPerawatan',
        'jenisPerawatan',
        'deskripsi',
    ];

    public function ternak()
    {
        return $this-
>belongsTo(Ternak::class,
'idTernak');
    }

    public function penyuluh()
    {
        return $this-
>belongsTo(User::class,
'idPenyuluh');
    }
}
```

Ternak.php

```
<?php

namespace App\Models;

use Carbon\Carbon;
use
Illuminate\Database\Eloquent\Mode
l;

class Ternak extends Model
{
    protected $table = 'ternak';

    protected $primaryKey =
'idTernak';

    protected $fillable = [
        'idPemilik',
        'namaTernak',
        'tanggalLahir',
        'jenis',
        'jenis_kelamin',
        'berat',
        'status',
        'asal',
        'fotoTernak',
        'keterangan'
    ];

    public function pemilik()
    {
        return $this-
>belongsTo(User::class,
'idPemilik');
    }
    public function
getUmurAttribute()
    {
        if ($this->tanggalLahir)
{
            return
Carbon::parse($this-
>tanggalLahir)->age;
        }
        return null;
    }
}
```

User.php

```
<?php

namespace App\Models;

// use
Illuminate\Contracts\Auth\MustVer
ifyEmail;
use
Illuminate\Database\Eloquent\Fact
ories\HasFactory;
use
Illuminate\Foundation\Auth\User
as Authenticatable;
use
Illuminate\Notifications\Notifiab
le;

class User extends
Authenticatable
{
    /** @use
HasFactory<\Database\Factories\Us
erFactory> */
    use HasFactory, Notifiable;

    /**
     * The table associated with
the model.
     */
    protected $table = 'users';

    /**
     * The primary key associated
with the table.
     */
    protected $primaryKey =
'idUser';
```

```
    /**
     * The attributes that are
mass assignable.
     *
     * @var list<string>
     */
    protected $fillable = [
        'nama',
        'email',
        'password',
        'phone',
        'role',
        'status'
    ];

    /**
     * The attributes that should
be hidden for serialization.
     *
     * @var list<string>
     */

    protected $casts = [
        'password' => 'string',
    ];
    protected $hidden = [
        'password',
        'remember_token',
    ];

    /**
     * Get the attributes that
should be cast.
     *
     * @return array<string,
string>
     */
    protected function casts():
array
    {
        return [
            'email_verified_at'
=> 'datetime',
            'password' =>
'hashed',
        ];
    }

    /**
     * Relationship dengan Ternak
(sebagai pemilik)
     * Foreign key: idPemilik
pada tabel ternaks -> references:
idUser
     */
    public function ternak()
    {
        return $this-
>hasMany(Ternak::class,
'idPemilik', 'idUser');
    }

    /**
     * Relationship dengan
Konsultasi (generic)
     */
    public function konsultasi()
    {
        return $this-
>hasMany(Konsultasi::class,
'idPeternak', 'idUser');
    }

    /**
     * Relationship dengan
Perawatan
     */
    public function perawatan()
    {
        return $this-
>hasMany(Perawatan::class,
'idPenyuluh', 'idUser');
    }

    /**
     * Relationship dengan
Laporan (sebagai penyuluh)
     * Foreign key: idPenyuluh
pada tabel laporan_kesehatan ->
references: idUser
     */
    public function
laporanSebagaiPenyuluh()
    {
        return $this-
>hasMany(Laporan::class,
'idPenyuluh', 'idUser');
    }

    /**
     * Relationship dengan
Laporan (sebagai peternak)
     * Foreign key: idPeternak
pada tabel laporan_kesehatan ->
references: idUser
     */
    public function
laporanSebagaiPeternak()
    {
        return $this-
>hasMany(Laporan::class,
'idPeternak', 'idUser');
    }

    /**
     * Alias untuk backward
compatibility
     */
    public function laporan()
    {
        return $this-
>laporanSebagaiPenyuluh();
    }

    /**
     * Relationship dengan Pakan
```

```
     */
    public function pakan()
    {
        return $this->hasMany(Pakan::class, 'idPenyuluh', 'idUser');
    }

    /**
     * Relationship dengan Konsultasi sebagai Peternak
     * Foreign key: idPeternak pada tabel konsultasis -> references: idUser
     */
    public function konsultasiSebagaiPeternak()
    {
        return $this->hasMany(Konsultasi::class, 'idPeternak', 'idUser');
    }

    /**
     * Relationship dengan Konsultasi sebagai Penyuluh
     * Foreign key: idPenyuluh pada tabel konsultasis -> references: idUser
     */
    public function konsultasiSebagaiPenyuluh()
    {
        return $this->hasMany(Konsultasi::class, 'idPenyuluh', 'idUser');
    }

    /**
     * Scope untuk filter berdasarkan role
     */
    public function scopeByRole($query, $role)
    {
        return $query->where('role', $role);
    }

    /**
     * Scope untuk peternak
     */
    public function scopePeternak($query)
    {
        return $query->where('role', 'peternak');
    }

    /**
     * Scope untuk penyuluh
     */
    public function scopePenyuluh($query)
    {
        return $query->where('role', 'penyuluh');
    }

    /**
     * Check if user is peternak
     */
    public function isPeternak()
    {
        return $this->role === 'peternak';
    }

    /**
     * Check if user is penyuluh
     */
    public function isPenyuluh()
    {
        return $this->role === 'penyuluh';
    }

    /**
     * Check if user is admin
     */
    public function isAdmin()
    {
        return $this->role === 'admin';
    }

    /**
     * Get user's full name (alias untuk nama)
     */
    public function getFullNameAttribute()
    {
        return $this->nama;
    }

    /**
     * Get user's display name
     */
    public function getDisplayNameAttribute()
    {
        return $this->nama . ' (' . ucfirst($this->role) . ')';
    }
}
```

Migration 0001_01_01_000000_create_users_table.php

```
<?php
```

```
use
Illuminate\Database\Migrations\Mi
gration;
use
Illuminate\Database\Schema\Bluepr
int;
use
Illuminate\Support\Facades\Schema
;

return new class extends
Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('users',
function (Blueprint $table) {
            $table->id('idUser');
            $table-
>string('nama');
            $table-
>string('email')->unique();
            $table-
>string('password');
            $table->enum('role',
['Admin', 'Penyuluh',
'Peternak'])-
>default('Peternak');
            $table-
>rememberToken();
            $table->timestamps();
        });

        Schema::create('password_
reset_tokens', function
(Blueprint $table) {
            $table-
>string('email')->primary();
            $table-
>string('token');
            $table-
>timestamp('created_at')-
>nullable();
        });

        Schema::create('sessions'
, function (Blueprint $table) {
            $table->string('id')-
>primary();
            $table-
>foreignId('user_id')-
>nullable()->index();
            $table-
>string('ip_address', 45)-
>nullable();
            $table-
>text('user_agent')->nullable();
            $table-
>longText('payload');
            $table-
>integer('last_activity')-
>index();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('use
rs');
        Schema::dropIfExists('pas
sword_reset_tokens');
        Schema::dropIfExists('ses
sions');
    }
};
```

0001_01_01_000001_create_cache_table.php

```
<?php

use
Illuminate\Database\Migrations\Mi
gration;
use
Illuminate\Database\Schema\Bluepr
int;
use
Illuminate\Support\Facades\Schema
;

return new class extends
Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('cache',
function (Blueprint $table) {
            $table-
>string('key')->primary();
            $table-
>mediumText('value');
            $table-
>integer('expiration');
        });

        Schema::create('cache_loc
ks', function (Blueprint $table)
{
            $table-
>string('key')->primary();
            $table-
>string('owner');
            $table-
>integer('expiration');
```

```
        });
    }

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

0001_01_01_000002_create_jobs_table.php

```
e<?php

use
Illuminate\Database\Migrations\Mi
gration;
use
Illuminate\Database\Schema\Bluepr
int;
use
Illuminate\Support\Facades\Schema
;

return new class extends
Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('jobs',
function (Blueprint $table) {
            $table->id();
            $table-
>string('queue')->index();
            $table-
>longText('payload');
            $table-
>unsignedTinyInteger('attempts');
            $table-
>unsignedInteger('reserved_at')-
>nullable();
            $table-
>unsignedInteger('available_at');
            $table-
>unsignedInteger('created_at');
        });

        Schema::create('job_batch
es', function (Blueprint $table)
{
            $table->string('id')-
>primary();
            $table-
>string('name');
            $table-
>integer('total_jobs');
            $table-
>integer('pending_jobs');
            $table-
>integer('failed_jobs');
            $table-
>longText('failed_job_ids');
            $table-
>mediumText('options')-
>nullable();
            $table-
>integer('cancelled_at')-
>nullable();
            $table-
>integer('created_at');
            $table-
>integer('finished_at')-
>nullable();
        });

        Schema::create('failed_jo
bs', function (Blueprint $table)
{
            $table->id();
            $table-
>string('uuid')->unique();
            $table-
>text('connection');
            $table-
>text('queue');
            $table-
>longText('payload');
            $table-
>longText('exception');
            $table-
>timestamp('failed_at')-
>useCurrent();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('job
s');
        Schema::dropIfExists('job
_batches');
        Schema::dropIfExists('fai
led_jobs');
    }
};
```

2025_05_20_115833_create_ternak_table.php

```
<?php
```

```
use
Illuminate\Database\Migrations\Mi
gration;
use
Illuminate\Database\Schema\Bluepr
int;
use
Illuminate\Support\Facades\Schema
;

return new class extends
Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('ternak',
function (Blueprint $table) {
            $table-
>id('idTernak');
            $table-
>unsignedBigInteger('idPemilik');
            $table-
>string('namaTernak', 100);
            $table-
>date('tanggalLahir');
            $table-
>integer('berat');
            $table-
>text('fotoTernak')->nullable();
            $table->timestamps();
            $table-
>foreign('idPemilik')-
>references('idUser')-
>on('users')-
>onDelete('cascade');
        });
    }

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

2025_05_20_121035_create_pakan_table.php

```
<?php

use
Illuminate\Database\Migrations\Mi
gration;
use
Illuminate\Database\Schema\Bluepr
int;
use
Illuminate\Support\Facades\Schema
;

return new class extends
Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('rekomenda
si_pakan', function (Blueprint
$table) {
            $table-
>id('idRekomendasi');
            $table-
>unsignedBigInteger('idTernak');
            $table-
>unsignedBigInteger('idPenyuluh')
;
            $table-
>date('tanggalRekomendasi');
            $table-
>string('jenisPakan', 100);
            $table-
>integer('jumlah');
            $table-
>text('saran')->nullable();
            $table->timestamps();
            $table-
>foreign('idTernak')-
>references('idTernak')-
>on('ternak')-
>onDelete('cascade');
            $table-
>foreign('idPenyuluh')-
>references('idUser')-
>on('users')-
>onDelete('cascade');
        });
    }

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

2025_05_20_121629_create_konsultasi_kesehatan_table.php

```
<?php

use
Illuminate\Database\Migrations\Mi
gration;
```

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

return new class extends
Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('konsultasi_kesehatan', function (Blueprint $table) {
            $table->id('idKonsultasi');
            $table->unsignedBigInteger('idPeternak');
            $table->unsignedBigInteger('idPenyuluh');
            $table->unsignedBigInteger('idTernak');
            $table->date('tanggalKonsultasi');
            $table->text('keluhan');
            $table->text('respon')->nullable();
            $table->timestamps();
            $table->foreign('idPeternak')->references('idUser')->on('users')->onDelete('cascade');
            $table->foreign('idPenyuluh')->references('idUser')->on('users')->onDelete('cascade');
            $table->foreign('idTernak')->references('idTernak')->on('ternak')->onDelete('cascade');
        });
    }

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

2025_07_05_093302_create_laporan_kesehatan_table.php

```
<?php

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

return new class extends
Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('laporan_kesehatan', function (Blueprint $table) {
            $table->id();
            $table->unsignedBigInteger('idPenyuluh');
            $table->unsignedBigInteger('idPeternak');
            $table->unsignedBigInteger('idTernak');
            $table->dateTime('tanggal_pemeriksaan');
            $table->decimal('berat_badan', 8, 2)->nullable();
            $table->decimal('suhu_tubuh', 4, 1);
            $table->enum('nafsu_makan', ['baik', 'menurun', 'tidak_ada']);
            $table->enum('pernafasan', ['normal', 'cepat', 'lambat', 'sesak']);
            $table->enum('kulit_bulu', ['normal', 'kusam', 'luka', 'parasit']);
            $table->enum('mata_hidung', ['normal', 'berair', 'bengkak', 'bernanah']);
            $table->enum('feses', ['normal', 'encer', 'keras', 'berdarah']);
            $table->enum('aktivitas', ['aktif', 'lesu', 'gelisah', 'lemas']);
            $table->text('tindakan');
```

```
            $table-
>text('rekomendasi');
            $table-
>enum('status_kesehatan',
['sehat', 'perlu_perhatian',
'sakit'])-
>default('perlu_perhatian');
            $table->timestamps();

            // Foreign key
constraints
            $table-
>foreign('idPenyuluh')-
>references('idUser')-
>on('users')-
>onDelete('cascade');
            $table-
>foreign('idPeternak')-
>references('idUser')-
>on('users')-
>onDelete('cascade');
            $table-
>foreign('idTernak')-
>references('idTernak')-
>on('ternak')-
>onDelete('cascade');

            // Indexes for better
performance
            $table-
>index(['idPenyuluh',
'tanggal_pemeriksaan']);
            $table-
>index(['idPeternak',
'tanggal_pemeriksaan']);
            $table-
>index(['idTernak',
'tanggal_pemeriksaan']);
            $table-
>index('status_kesehatan');
            $table-
>index('tanggal_pemeriksaan');
        });
    }

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

2025_07_05_093325_create_activity_logs_table.php

```
<?php

use
Illuminate\Database\Migrations\Mi
gration;
use
Illuminate\Database\Schema\Bluepr
int;
use
Illuminate\Support\Facades\Schema
;

return new class extends
Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('activity_
logs', function (Blueprint
$table) {
            $table->id();
            $table-
>unsignedBigInteger('user_id');
            $table-
>string('user_type', 50); //
'penyuluh', 'peternak', 'admin'
            $table-
>string('action', 100);
            $table->json('data')-
>nullable();
            $table-
>ipAddress('ip_address')-
>nullable();
            $table-
>text('user_agent')->nullable();
            $table-
>timestamp('created_at');

            // Foreign key
            $table-
>foreign('user_id')-
>references('idUser')-
>on('users')-
>onDelete('cascade');

            // Indexes
            $table-
>index(['user_id',
'created_at']);
            $table-
>index(['action', 'created_at']);
            $table-
>index('user_type');
        });
    }

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

2025_07_11_140443_create_informasi_table.php

```
<?php

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

return new class extends
Migration
{
    public function up(): void
    {
        Schema::create('informasis', function (Blueprint $table) {
            $table->id('idInformasi');
            $table->string('judul');
            $table->text('deskripsi');
            $table->dateTime('tanggal_kegiatan')->nullable();
            $table->string('lokasi')->nullable();
            $table->string('kategori')->default('umum');
            $table->unsignedBigInteger('idPenyuluh')->nullable();
            $table->string('foto')->nullable();
            $table->foreign('idPenyuluh')->references('idUser')->on('users')->onDelete('cascade');
            $table->timestamps();
        });
    }

    public function down(): void
    {
        Schema::dropIfExists('informasi');
    }
};
```

2025_07_11_141811_create_pakan_table.php

```
<?php

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

return new class extends
Migration
{
    public function up(): void
    {
        Schema::create('pakan', function (Blueprint $table) {
            $table->id('idPakan');
            $table->string('judul');
            $table->text('deskripsi');
            $table->string('foto')->nullable();
            $table->string('sumber')->nullable();
            $table->enum('jenis_pakan', ['hijauan', 'konsentrat', 'silase', 'campuran', 'lainnya'])->default('lainnya');
            $table->boolean('is_published')->default(true);
            $table->timestamps();
        });
    }

    public function down(): void
    {
        Schema::dropIfExists('informasi_pakan');
    }
};
```

View

Admin

dashboard.blade.php

```
@section('content')
    <div class="space-y-6">
        <!-- Welcome Section -->
        <div class="bg-gradient-to-r from-primary to-secondary rounded-2xl p-6 text-white">
            <div class="flex items-center justify-between">
                <div>
                    <h2 class="text-2xl font-bold mb-2">Selamat Datang, {{
```

```
Auth::user()->nama ??
Auth::user()->name }}! 👋
                    </h2>
                    <p
class="text-green-100">Hari ini
adalah {{ \Carbon\Carbon::now()-
>isoFormat('dddd, D MMMM Y')
}}</p>
                    <p
class="text-sm text-green-200 mt-
1">Pakeng Ternak Desa dalam
kondisi baik dan siap
produktif!</p>
                </div>
                <div
class="hidden md:block">
                    <div
class="bg-white/20 backdrop-blur-
sm rounded-xl p-4 text-center">
                        <div
class="text-3xl mb-2">🐄</div>
                        <div
class="text-sm">Total Ternak Desa
Pakeng</div>
                        <div
class="text-2xl font-bold">{{
$totalTernak }}</div>
                    </div>
                </div>
            </div>
        </div>

        <!-- Quick Stats Cards --
>
        <div class="grid grid-
cols-1 md:grid-cols-2 lg:grid-
cols-4 gap-6">
            <!-- Total Ternak -->
            <div class="stats-
card bg-white rounded-xl p-8
shadow-sm border border-gray-
100">
                <div class="flex
items-center justify-between">
                    <div>
                        <p
class="text-gray-600 text-sm
font-medium">Total Ternak</p>
                        <p
class="text-3xl font-bold text-
gray-900">{{ $totalTernak ?? 25
}}</p>
                        <p
class="text-green-600 text-sm mt-
1">
                        </p>
                    </div>
                    <div
class="bg-blue-100 p-3 rounded-
full">
                        <svg
class="w-8 h-8 text-blue-600"
fill="none" stroke="currentColor"
viewBox="0 0 24 24">
                            <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
                                d
="M19 21V5a2 2 0 00-2-2H7a2 2 0
00-2 2v16m14 0h2m-2 0h-5m-9 0H3m2
0h5M9 7h1m-1 4h1m4-4h1m-1 4h1m-5
10v-5a1 1 0 011-1h2a1 1 0 011
1v5m-4 0h4">
                            </pat
h>
                        </svg>
                    </div>
                </div>
            </div>

            <!-- Ternak Sehat -->
            <div class="stats-
card bg-white rounded-xl p-8
shadow-sm border border-gray-
100">
                <div class="flex
items-center justify-between">
                    <div>
                        <p
class="text-gray-600 text-sm
font-medium">Ternak Sehat</p>
                        <p
class="text-3xl font-bold text-
gray-900">{{ $ternakSehat ?? 23
}}</p>
                    </div>
                    <div
class="bg-green-100 p-3 rounded-
full">
                        <svg
class="w-8 h-8 text-green-600"
fill="none" stroke="currentColor"
viewBox="0 0 24 24">
                            <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
                                d
="M4.318 6.318a4.5 4.5 0 000
6.364L12 20.364l7.682-7.682a4.5
4.5 0 00-6.364-6.364L12 7.636l-
1.318-1.318a4.5 4.5 0 00-6.364
0z">
                            </pat
h>
                        </svg>
                    </div>
                </div>
            </div>

            <!-- Perlu Perhatian
-->
            <div class="stats-
card bg-white rounded-xl p-8
shadow-sm border border-gray-100
@endpush
```

user.blade.php

```
@section('content')
    <div class="space-y-6">
        <div class="flex flex-col
sm:flex-row sm:items-center
sm:justify-between gap-4">
            <div class="flex
items-center space-x-4">
                <div
class="relative">
                    <div
class="absolute inset-y-0 left-0
pl-3 flex items-center pointer-
events-none">
                        <svg
class="h-5 w-5 text-gray-400"
fill="none" stroke="currentColor"
viewBox="0 0 24 24">
                            <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"

d="M21 21l-6-6m2-5a7 7 0 11-14 0
7 7 0 0114 0z"></path>
                        </svg>
                    </div>
                    <input
type="text" id="searchInput"

class="search-box block w-full
pl-10 pr-3 py-2 border border-
gray-300 rounded-lg
focus:outline-none focus:ring-2
focus:ring-primary focus:border-
transparent"

placeholder="Cari user..."
onkeyup="searchUser()">
                </div>
                <select
id="roleFilter"
                    class="block
px-3 py-2 border border-gray-300
rounded-lg focus:outline-none
focus:ring-2 focus:ring-primary
focus:border-transparent"

onchange="filterByRole()">
                    <option
value="">Semua Role</option>
                    <option
value="Admin">Admin</option>
                    <option
value="Penyuluh">Penyuluh</option
>
                    <option
value="Peternak">Peternak</option
>
                </select>
            </div>
            <div class="flex
items-center space-x-3">
                <button
onclick="openAddModal()"

class="inline-flex items-center
px-4 py-2 bg-primary text-white
rounded-lg text-sm font-medium
hover:bg-secondary focus:outline-
none focus:ring-2 focus:ring-
primary transform hover:scale-105
transition-all">
                    <svg
class="h-4 w-4 mr-2" fill="none"
stroke="currentColor" viewBox="0
0 24 24">
                        <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"

d="M12 6v6m0 0v6m0-6h6m-6
0H6"></path>
                    </svg>
                    Tambah User
                </button>
            </div>
        </div>

        <div class="grid grid-
cols-1 md:grid-cols-4 gap-4">
            <div class="bg-white
rounded-xl p-6 shadow-sm border
border-gray-100">
                <div class="flex
items-center justify-between">
                    <div>
                        <p
class="text-sm font-medium text-
gray-600">Total User</p>
                        <p
class="text-2xl font-bold text-
gray-900">{{ $totalUser ?? 150
}}</p>
                    </div>
                    <div
class="bg-blue-100 p-3 rounded-
full">
                        <svg
class="w-6 h-6 text-blue-600"
fill="none" stroke="currentColor"
viewBox="0 0 24 24">
                            <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"

d="M12 4.354a4 4 0 110 5.292M15
21H3v-1a6 6 0 0112 0v1zm0 0h6v-
1a6 6 0 00-9-5.197m13.5-9a2.5 2.5
0 11-5 0 2.5 2.5 0 015 0z">

</path>
                        </svg>
                    </div>
```

```
                </div>
            </div>

            <div class="bg-white
rounded-xl p-6 shadow-sm border
border-gray-100">
                <div class="flex
items-center justify-between">
                    <div>
                        <p
class="text-sm font-medium text-
gray-600">Aktif</p>
                        <p
class="text-2xl font-bold text-
green-600">{{ $userAktif ?? 135
}}</p>
                    </div>
                    <div
class="bg-green-100 p-3 rounded-
full">
                        <svg
class="w-6 h-6 text-green-600"
fill="none" stroke="currentColor"
viewBox="0 0 24 24">
                            <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"

d="M9 12l2 2 4-4m6 2a9 9 0 11-18
0 9 9 0 0118 0z"></path>
                        </svg>
                    </div>
                </div>
            </div>

            <div class="bg-white
rounded-xl p-6 shadow-sm border
border-gray-100">
                <div class="flex
items-center justify-between">
                    <div>
                        <p
class="text-sm font-medium text-
gray-600">Non-Aktif</p>
                        <p
class="text-2xl font-bold text-
red-600">{{ $userNonAktif ?? 12
}}</p>
                    </div>
                    <div
class="bg-red-100 p-3 rounded-
full">
                        <svg
class="w-6 h-6 text-red-600"
fill="none" stroke="currentColor"
viewBox="0 0 24 24">
                            <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"

d="M10 14l2-2m0 0l2-2m-2 2l-2-2m2
2l2 2m7-2a9 9 0 11-18 0 9 9 0
0118 0z"></path>
```

```
                @csrf
                <div class="grid
grid-cols-1 md:grid-cols-2 gap-
6">
                    <div
class="col-span-1 md:col-span-2">
                        <label
for="name" class="block text-sm
font-medium text-gray-700 mb-2">
                            Nama
Lengkap <span class="text-red-
500">*</span>
                        </label>
                        <input
type="text" id="name" name="nama"
required

class="w-full px-3 py-2 border
border-gray-300 rounded-lg
focus:outline-none focus:ring-2
focus:ring-primary focus:border-
transparent"

placeholder="Masukkan nama
lengkap">
                    </div>

                    <div>
                        <label
for="email" class="block text-sm
font-medium text-gray-700 mb-2">
                            Email
<span class="text-red-
500">*</span>
                        </label>
                        <input
type="email" id="email"
name="email" required

class="w-full px-3 py-2 border
border-gray-300 rounded-lg
focus:outline-none focus:ring-2
focus:ring-primary focus:border-
transparent"

placeholder="user@example.com">
                    </div>

                    <div>
                        <label
for="phone" class="block text-sm
font-medium text-gray-700 mb-2">
                            No.
Telepon
                        </label>
                        <input
type="tel" id="phone"
name="phone"

class="w-full px-3 py-2 border
border-gray-300 rounded-lg
focus:outline-none focus:ring-2
```

```
focus:ring-primary focus:border-
transparent"

placeholder="08xxxxxxxxx">
                    </div>

                    <div>
                        <label
for="password" class="block text-
sm font-medium text-gray-700 mb-
2">

Password <span class="text-red-
500">*</span>
                        </label>
                        <input
type="password" id="password"
name="password" required

class="w-full px-3 py-2 border
border-gray-300 rounded-lg
focus:outline-none focus:ring-2
focus:ring-primary focus:border-
transparent"

placeholder="Minimal 8 karakter">
                    </div>

                    <div>
                        <label
for="password_confirmation"
class="block text-sm font-medium
text-gray-700 mb-2">

Konfirmasi Password <span
class="text-red-500">*</span>
                        </label>
                        <input
type="password"
id="password_confirmation"
name="password_confirmation"
required

class="w-full px-3 py-2 border
border-gray-300 rounded-lg
focus:outline-none focus:ring-2
focus:ring-primary focus:border-
transparent"

placeholder="Ulangi password">
                    </div>

                    <div>
                        <label
for="role" class="block text-sm
font-medium text-gray-700 mb-2">
                            Role
<span class="text-red-
500">*</span>
                        </label>
                        <select
id="role" name="role" required

class="w-full px-3 py-2 border
border-gray-300 rounded-lg
focus:outline-none focus:ring-2
focus:ring-primary focus:border-
transparent">

<option value="">Pilih
Role</option>

<option
value="Admin">Admin</option>

<option
value="Penyuluh">Penyuluh</option
>

<option
value="Peternak">Peternak</option
>
                        </select>
                    </div>

                    <div
class="col-span-1 md:col-span-2">
                        <label
for="status" class="block text-sm
font-medium text-gray-700 mb-2">

Status <span class="text-red-
500">*</span>
                        </label>
                        <div
class="grid grid-cols-3 gap-4">

<label

class="flex items-center p-3
border border-gray-300 rounded-lg
cursor-pointer hover:bg-gray-50
transition-colors">

<input type="radio" name="status"
value="aktif" checked

class="h-4 w-4 text-primary
focus:ring-primary border-gray-
300">

<div class="ml-3">

<div class="flex items-center">

<span class="w-3 h-3 bg-green-500
rounded-full mr-2"></span>

<span class="text-sm font-medium
text-gray-900">Aktif</span>

</div>

<p class="text-xs text-gray-
500">User dapat login</p>
```

```
</div>

</label>


<label

class="flex items-center p-3
border border-gray-300 rounded-lg
cursor-pointer hover:bg-gray-50
transition-colors">

<input type="radio" name="status"
value="nonaktif"

class="h-4 w-4 text-primary
focus:ring-primary border-gray-
300">

<div class="ml-3">

<div class="flex items-center">

<span class="w-3 h-3 bg-red-500
rounded-full mr-2"></span>

<span class="text-sm font-medium
text-gray-900">Non-Aktif</span>

</div>

<p class="text-xs text-gray-
500">User tidak dapat login</p>

</div>

</label>


<label

class="flex items-center p-3
border border-gray-300 rounded-lg
cursor-pointer hover:bg-gray-50
transition-colors">

<input type="radio" name="status"
value="suspend"

class="h-4 w-4 text-primary
focus:ring-primary border-gray-
300">

<div class="ml-3">

<div class="flex items-center">

<span class="w-3 h-3 bg-yellow-
500 rounded-full mr-2"></span>

<span class="text-sm font-medium
text-gray-900">Suspend</span>

</div>

<p class="text-xs text-gray-
500">Akun di suspend</p>

</div>

</label>
                        </div>
                    </div>
                </div>

                <div class="flex
items-center justify-end space-x-
3 mt-8 pt-6 border-t border-gray-
200">
                    <button
type="button"
onclick="closeAddModal()"

class="px-4 py-2 bg-gray-300
text-gray-700 text-sm font-medium
rounded-lg hover:bg-gray-400
focus:outline-none focus:ring-2
focus:ring-gray-300 transition-
colors">
                        Batal
                    </button>
                    <button
type="submit" id="submitBtn"

class="px-6 py-2 bg-primary text-
white text-sm font-medium
rounded-lg hover:bg-secondary
focus:outline-none focus:ring-2
focus:ring-primary transition-
colors">
                        <span
id="submitText">Simpan
User</span>
                        <svg
id="submitLoading" class="hidden
animate-spin -mr-1 ml-3 h-4 w-4
text-white inline"

fill="none" viewBox="0 0 24 24">

<circle class="opacity-25"
cx="12" cy="12" r="10"
stroke="currentColor"

stroke-width="4"></circle>
                            <path
class="opacity-75"
fill="currentColor"

d="M4 12a8 8 0 018-8V0C5.373 0 0
5.373 0 12h4zm2 5.291A7.962 7.962
0 014 12H0c0 3.042 1.135 5.824 3
7.938l3-2.647z">

</path>
```

```
                    </svg>
                </button>
            </div>
        </form>
    </div>
</div>

<!-- Detail User Modal -->
<div id="detailModal"
    class="fixed inset-0 bg-
gray-600 bg-opacity-50 overflow-
y-auto h-full w-full hidden z-50
modal-backdrop">
    <div class="relative top-
10 mx-auto p-5 border w-full max-
w-3xl shadow-lg rounded-lg bg-
white modal-content">
        <div class="flex
items-center justify-between mb-
6">
            <h3 class="text-
xl font-semibold text-gray-
900">Detail User</h3>
            <button
onclick="closeDetailModal()"
class="text-gray-400 hover:text-
gray-600">
                <svg
class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0
0 24 24">
                    <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
d="M6 18L18 6M6 6l12 12">
                    </path>
                </svg>
            </button>
        </div>

        <div class="grid
grid-cols-1 md:grid-cols-2 gap-
6">
            <div
class="space-y-4">
                <div
class="flex items-center space-x-
4 mb-6">
                    <div
class="user-avatar" style="width:
80px; height: 80px; font-size:
2rem;" id="detailAvatar">
                        U
                    </div>
                    <div>
                        <div
class="detail-label">Nama
Lengkap</div>
                        <div
class="detail-value text-lg font-
semibold" id="detailNama">-</div>
                    </div>
                </div>

                <div>
                    <div
class="detail-label">ID
User</div>
                    <div
class="detail-value"
id="detailId">-</div>
                </div>

                <div>
                    <div
class="detail-label">Email</div>
                    <div
class="detail-value"
id="detailEmail">-</div>
                </div>

                <div>
                    <div
class="detail-label">No.
Telepon</div>
                    <div
class="detail-value"
id="detailPhone">-</div>
                </div>

                <div>
                    <div
class="detail-label">Role</div>
                    <div
class="detail-value">
                        <span
id="detailRole" class="role-
badge">-</span>
                    </div>
                </div>

                <div>
                    <div
class="detail-label">Status</div>
                    <div
class="detail-value">
                        <span
id="detailStatus" class="status-
badge">-</span>
                    </div>
                </div>
            </div>

            <div
class="space-y-4">
                <div>
                    <div
class="detail-label">Tanggal
Bergabung</div>
                    <div
class="detail-value"
id="detailCreated">-</div>
                </div>

                <div>
```

```
                        <div
class="detail-label">Login
Terakhir</div>
                        <div
class="detail-value"
id="detailLastLogin">-</div>
                    </div>

                </div>
            </div>

            <div class="flex
items-center justify-end space-x-
3 mt-8 pt-6 border-t border-gray-
200">
                <button
onclick="closeDetailModal()"
                    class="px-4
py-2 bg-gray-300 text-gray-700
text-sm font-medium rounded-lg
hover:bg-gray-400 focus:outline-
none focus:ring-2 focus:ring-
gray-300 transition-colors">
                    Tutup
                </button>
            </div>
        </div>
    </div>

    <!-- Edit User Modal -->
    <div id="editModal"
        class="fixed inset-0 bg-
gray-600 bg-opacity-50 overflow-
y-auto h-full w-full hidden z-50
modal-backdrop">
        <div class="relative top-
10 mx-auto p-5 border w-full max-
w-2xl shadow-lg rounded-lg bg-
white modal-content">
            <div class="flex
items-center justify-between mb-
6">
                <h3 class="text-
xl font-semibold text-gray-
900">Edit User</h3>
                <button
onclick="closeEditModal()"
class="text-gray-400 hover:text-
gray-600">
                    <svg
class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0
0 24 24">
                        <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
d="M6 18L18 6M6 6l12 12">
                        </path>
                    </svg>
                </button>
            </div>

            <form
id="editUserForm" method="POST">
                @csrf
                @method('PUT')
                <input
type="hidden" id="editUserId"
name="id">

                <div class="grid
grid-cols-1 md:grid-cols-2 gap-
6">
                    <div
class="col-span-1 md:col-span-2">
                        <label
for="editName" class="block text-
sm font-medium text-gray-700 mb-
2">
                            Nama
Lengkap <span class="text-red-
500">*</span>
                        </label>
                        <input
type="text" id="editName"
name="nama" required

class="w-full px-3 py-2 border
border-gray-300 rounded-lg
focus:outline-none focus:ring-2
focus:ring-primary focus:border-
transparent"

placeholder="Masukkan nama
lengkap">
                    </div>

                    <div>
                        <label
for="editEmail" class="block
text-sm font-medium text-gray-700
mb-2">
                            Email
<span class="text-red-
500">*</span>
                        </label>
                        <input
type="email" id="editEmail"
name="email" required

class="w-full px-3 py-2 border
border-gray-300 rounded-lg
focus:outline-none focus:ring-2
focus:ring-primary focus:border-
transparent"

placeholder="user@example.com">
                    </div>

                    <div>
                        <label
for="editPhone" class="block
text-sm font-medium text-gray-700
mb-2">
```

```
                            No.
Telepon
                        </label>
                        <input
type="tel" id="editPhone"
name="phone"

class="w-full px-3 py-2 border
border-gray-300 rounded-lg
focus:outline-none focus:ring-2
focus:ring-primary focus:border-
transparent"

placeholder="08xxxxxxxxx">
                    </div>

                    <div>
                        <label
for="editRole" class="block text-
sm font-medium text-gray-700 mb-
2">
                            Role
<span class="text-red-
500">*</span>
                        </label>
                        <select
id="editRole" name="role"
required

class="w-full px-3 py-2 border
border-gray-300 rounded-lg
focus:outline-none focus:ring-2
focus:ring-primary focus:border-
transparent">

<option value="">Pilih
Role</option>

<option
value="Admin">Admin</option>

<option
value="Penyuluh">Penyuluh</option
>

<option
value="Peternak">Peternak</option
>
                        </select>
                    </div>


                    <div
class="col-span-1 md:col-span-2">
                        <label
for="editStatus" class="block
text-sm font-medium text-gray-700
mb-2">

Status <span class="text-red-
500">*</span>
                        </label>
                            <div
class="grid grid-cols-3 gap-4">

<label

class="flex items-center p-3
border border-gray-300 rounded-lg
cursor-pointer hover:bg-gray-50
transition-colors">

<input type="radio" name="status"
value="aktif"
id="editStatusAktif"

class="h-4 w-4 text-primary
focus:ring-primary border-gray-
300">

<div class="ml-3">

<div class="flex items-center">

<span class="w-3 h-3 bg-green-500
rounded-full mr-2"></span>

<span class="text-sm font-medium
text-gray-900">Aktif</span>

</div>

<p class="text-xs text-gray-
500">User dapat login</p>

</div>

</label>


<label

class="flex items-center p-3
border border-gray-300 rounded-lg
cursor-pointer hover:bg-gray-50
transition-colors">

<input type="radio" name="status"
value="nonaktif"
id="editStatusNonaktif"

class="h-4 w-4 text-primary
focus:ring-primary border-gray-
300">

<div class="ml-3">

<div class="flex items-center">

<span class="w-3 h-3 bg-red-500
rounded-full mr-2"></span>

<span class="text-sm font-medium
text-gray-900">Non-Aktif</span>
```

```
</div>

<p class="text-xs text-gray-
500">User tidak dapat login</p>

</div>

</label>


<label

class="flex items-center p-3
border border-gray-300 rounded-lg
cursor-pointer hover:bg-gray-50
transition-colors">

<input type="radio" name="status"
value="suspend"
id="editStatusSuspend"

class="h-4 w-4 text-primary
focus:ring-primary border-gray-
300">

<div class="ml-3">

<div class="flex items-center">

<span class="w-3 h-3 bg-yellow-
500 rounded-full mr-2"></span>

<span class="text-sm font-medium
text-gray-900">Suspend</span>

</div>

<p class="text-xs text-gray-
500">Akun di suspend</p>

</div>

</label>
                        </div>
                    </div>
                </div>

                <div class="flex
items-center justify-end space-x-
3 mt-8 pt-6 border-t border-gray-
200">
                    <button
type="button"
onclick="closeEditModal()"

class="px-4 py-2 bg-gray-300
text-gray-700 text-sm font-medium
rounded-lg hover:bg-gray-400
focus:outline-none focus:ring-2
focus:ring-gray-300 transition-
colors">
                        Batal
                    </button>
                    <button
type="submit" id="editSubmitBtn"

class="px-6 py-2 bg-primary text-
white text-sm font-medium
rounded-lg hover:bg-secondary
focus:outline-none focus:ring-2
focus:ring-primary transition-
colors">
                        <span
id="editSubmitText">Update
User</span>
                        <svg
id="editSubmitLoading"
class="hidden animate-spin -mr-1
ml-3 h-4 w-4 text-white inline"

fill="none" viewBox="0 0 24 24">

<circle class="opacity-25"
cx="12" cy="12" r="10"
stroke="currentColor"

stroke-width="4"></circle>
                            <path
class="opacity-75"
fill="currentColor"

d="M4 12a8 8 0 018-8V0C5.373 0 0
5.373 0 12h4zm2 5.291A7.962 7.962
0 014 12H0c0 3.042 1.135 5.824 3
7.938l3-2.647z">

</path>
                        </svg>
                    </button>
                </div>
            </form>
        </div>
    </div>

    <!-- Delete Confirmation
Modal -->
    <div id="deleteModal"
        class="fixed inset-0 bg-
gray-600 bg-opacity-50 overflow-
y-auto h-full w-full hidden z-50
modal-backdrop">
        <div class="relative top-
20 mx-auto p-5 border w-96
shadow-lg rounded-lg bg-white
modal-content">
            <div class="mt-3
text-center">
                <div class="mx-
auto flex items-center justify-
center h-12 w-12 rounded-full bg-
red-100">
                    <svg
class="h-6 w-6 text-red-600"
fill="none" stroke="currentColor"
viewBox="0 0 24 24">
```

```
                        <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"

d="M12 9v2m0 4h.01m-6.938
4h13.856c1.54 0 2.502-1.667
1.732-2.5L13.732 4c-.77-.833-
1.664-.833-2.464 0L4.35 16.5c-
.77.833.192 2.5 1.732 2.5z">
                        </path>
                    </svg>
                </div>
                <h3 class="text-
lg font-medium text-gray-900 mt-
4">Hapus User</h3>
                <p class="text-sm
text-gray-500 mt-2">
                    Apakah Anda
yakin ingin menghapus user <span
id="deleteNamaUser" class="font-
semibold"></span>?
                    Tindakan ini
tidak dapat dibatalkan.
                </p>
                <div
class="items-center px-4 py-3 mt-
4 flex justify-center space-x-2">
                    <button
id="confirmDeleteBtn"
type="button"

class="px-4 py-2 bg-red-600 text-
white text-base font-medium
rounded-md hover:bg-red-700
focus:outline-none focus:ring-2
focus:ring-red-500 min-w-[100px]
flex items-center justify-
center">
                        <span
id="deleteText">Hapus</span>
                        <svg
id="deleteLoading" class="hidden
animate-spin ml-2 h-4 w-4 text-
white" fill="none"

viewBox="0 0 24 24">

<circle class="opacity-25"
cx="12" cy="12" r="10"
stroke="currentColor"

stroke-width="4"></circle>
                            <path
class="opacity-75"
fill="currentColor"

d="M4 12a8 8 0 018-8V0C5.373 0 0
5.373 0 12h4zm2 5.291A7.962 7.962
0 014 12H0c0 3.042 1.135 5.824 3
7.938l3-2.647z">

</path>
                        </svg>
                    </button>
                    <button
onclick="closeDeleteModal()"
type="button"

class="px-4 py-2 bg-gray-300
text-gray-900 text-base font-
medium rounded-md hover:bg-gray-
400 focus:outline-none
focus:ring-2 focus:ring-gray-300
min-w-[100px]">
                        Batal
                    </button>
                </div>
            </div>
        </div>
    </div>
@endsection

@push('scripts')
    <script>
        // Global variables
        let currentView = "grid";
        let deleteId = null;
        let currentUserData =
null;

        // View switching
function
        function switchView(view)
{
            currentView = view;
            const gridView =
document.getElementById("gridView
");
            const tableView =
document.getElementById("tableVie
w");
            const gridBtn =
document.getElementById("gridView
Btn");
            const tableBtn =
document.getElementById("tableVie
wBtn");

            if (view === "grid")
{

gridView.classList.remove("hidden
");

tableView.classList.add("hidden")
;

gridBtn.classList.add("view-btn-
active", "bg-white", "text-gray-
900", "shadow-sm");

gridBtn.classList.remove("text-
gray-600");

tableBtn.classList.remove("view-
```

```
btn-active", "bg-white", "text-
gray-900", "shadow-sm");

tableBtn.classList.add("text-
gray-600");
            } else {

gridView.classList.add("hidden");

tableView.classList.remove("hidde
n");

tableBtn.classList.add("view-btn-
active", "bg-white", "text-gray-
900", "shadow-sm");

tableBtn.classList.remove("text-
gray-600");

gridBtn.classList.remove("view-
btn-active", "bg-white", "text-
gray-900", "shadow-sm");

gridBtn.classList.add("text-gray-
600");
            }
        }

        // Search function
        function searchUser() {
            const searchTerm =
document.getElementById("searchIn
put").value.toLowerCase();
            const cards =
document.querySelectorAll(".user-
card");
            const rows =
document.querySelectorAll(".user-
row");

            cards.forEach((card)
=> {
                const name =
card.getAttribute("data-
name").toLowerCase();
                const email =
card.getAttribute("data-
email").toLowerCase();

card.style.display =
(name.includes(searchTerm) ||
email.includes(searchTerm)) ?
"block" : "none";
            });

            rows.forEach((row) =>
{
                const name =
row.getAttribute("data-
name").toLowerCase();
                const email =
row.getAttribute("data-
email").toLowerCase();
                row.style.display
= (name.includes(searchTerm) ||
email.includes(searchTerm)) ?
"table-row" :
                    "none";
            });
        }

        // Filter by status
function
        function filterByStatus()
{
            const statusFilter =
document.getElementById("statusFi
lter").value;
            const cards =
document.querySelectorAll(".user-
card");
            const rows =
document.querySelectorAll(".user-
row");

            cards.forEach((card)
=> {
                const status =
card.getAttribute("data-status");

card.style.display =
!statusFilter || status ===
statusFilter ? "block" : "none";
            });

            rows.forEach((row) =>
{
                const status =
row.getAttribute("data-status");
                row.style.display
= !statusFilter || status ===
statusFilter ? "table-row" :
"none";
            });
        }

        // Filter by role
function
        function filterByRole() {
            const roleFilter =
document.getElementById("roleFilt
er").value;
            const cards =
document.querySelectorAll(".user-
card");
            const rows =
document.querySelectorAll(".user-
row");

            cards.forEach((card)
=> {
                const role =
card.getAttribute("data-role");

card.style.display = !roleFilter
```

```
|| role === roleFilter ? "block"
: "none";
            });

            rows.forEach((row) =>
{
                const role =
row.getAttribute("data-role");
                row.style.display
= !roleFilter || role ===
roleFilter ? "table-row" :
"none";
            });
        }

        // Add modal functions
        function openAddModal() {

document.getElementById("addModal
").classList.remove("hidden");

document.body.style.overflow =
"hidden";
            setTimeout(() =>
document.getElementById("name").f
ocus(), 100);
        }

        function closeAddModal()
{
            const modal =
document.getElementById("addModal
");

modal.classList.add("hidden");

document.body.style.overflow =
"auto";

document.getElementById("addUserF
orm").reset();

            // Reset status radio
button ke "aktif"
            const aktifRadio =
document.querySelector('input[nam
e="status"][value="aktif"]');
            if (aktifRadio)
aktifRadio.checked = true;
        }

        // Get user data from
element
        function
getUserDataFromElement(element) {
            let parent =
element.closest('[data-id]');
            if (!parent) {
                parent =
element.closest('.user-card,
.user-row');
            }
            if (!parent) {

console.error('Parent element
dengan data tidak ditemukan');
                return null;
            }

            return {
                id:
parent.getAttribute("data-id"),
                name:
parent.getAttribute("data-name"),
                email:
parent.getAttribute("data-
email"),
                phone:
parent.getAttribute("data-
phone"),
                role:
parent.getAttribute("data-role"),
                status:
parent.getAttribute("data-
status"),
                created:
parent.getAttribute("data-
created"),
                lastLogin:
parent.getAttribute("data-last-
login")
            };
        }

        // Detail modal functions
        function
openDetailModal(element) {
            const data =
getUserDataFromElement(element);
            if (!data) {
                alert('Data user
tidak ditemukan');
                return;
            }

            currentUserData =
data;

document.getElementById("detailAv
atar").textContent = data.name ?
data.name.substring(0,
2).toUpperCase() : 'U';

document.getElementById("detailNa
ma").textContent = data.name ||
'-';

document.getElementById("detailId
").textContent = "#" + (data.id
|| '-');

document.getElementById("detailEm
ail").textContent = data.email ||
'-';
```

```
document.getElementById("detailPh
one").textContent = data.phone ||
'-';

            // Format tanggal
            if (data.created &&
data.created !== '-') {
                try {
                    const date =
new Date(data.created);

document.getElementById("detailCr
eated").textContent =
date.toLocaleDateString("id-ID",
{
                        year:
'numeric',
                        month:
'long',
                        day:
'numeric'
                    });
                } catch (e) {

document.getElementById("detailCr
eated").textContent =
data.created;
                }
            } else {

document.getElementById("detailCr
eated").textContent = '-';
            }

            if (data.lastLogin &&
data.lastLogin !== '-') {
                try {
                    const date =
new Date(data.lastLogin);

document.getElementById("detailLa
stLogin").textContent =
date.toLocaleDateString("id-ID",
{
                        year:
'numeric',
                        month:
'long',
                        day:
'numeric',
                        hour: '2-
digit',
                        minute:
'2-digit'
                    });
                } catch (e) {

document.getElementById("detailLa
stLogin").textContent =
data.lastLogin;
                }
            } else {

document.getElementById("detailLa
stLogin").textContent = '-';
            }

            const roleBadge =
document.getElementById("detailRo
le");
            if (data.role) {

roleBadge.textContent =
data.role.charAt(0).toUpperCase()
+ data.role.slice(1);

roleBadge.className = `role-badge
role-${data.role}`;
            } else {

roleBadge.textContent = '-';

roleBadge.className = 'role-
badge';
            }

            const statusBadge =
document.getElementById("detailSt
atus");
            if (data.status) {

statusBadge.textContent =
data.status.charAt(0).toUpperCase
() + data.status.slice(1);

statusBadge.className = `status-
badge status-${data.status}`;
            } else {

statusBadge.textContent = '-';

statusBadge.className = 'status-
badge';
            }


document.getElementById("detailMo
dal").classList.remove("hidden");

document.body.style.overflow =
"hidden";
        }

        function
closeDetailModal() {

document.getElementById("detailMo
dal").classList.add("hidden");

document.body.style.overflow =
"auto";
            currentUserData =
null;
        }
```

```
        // Edit modal functions
        function
openEditModal(element) {
            const data =
getUserDataFromElement(element);
            if (!data) {
                alert('Data user
tidak ditemukan');
                return;
            }


populateEditForm(data);

document.getElementById("editModa
l").classList.remove("hidden");

document.body.style.overflow =
"hidden";
        }

        function
populateEditForm(data) {

document.getElementById("editUser
Id").value = data.id || '';

document.getElementById("editName
").value = data.name || '';

document.getElementById("editEmai
l").value = data.email || '';

document.getElementById("editPhon
e").value = data.phone || '';

document.getElementById("editRole
").value = data.role || '';

            // Set status radio
button
            if (data.status) {
                const statusRadio
=
document.getElementById("editStat
us" +
data.status.charAt(0).toUpperCase
() + data.status
                    .slice(1));
                if (statusRadio)
{

statusRadio.checked = true;
                }
            }

            const form =
document.getElementById("editUser
Form");
            form.action =
`/admin/users/${data.id}/update`;
        }

        function closeEditModal()
{

document.getElementById("editModa
l").classList.add("hidden");

document.body.style.overflow =
"auto";

document.getElementById("editUser
Form").reset();
        }

        // Delete modal functions
        function
openDeleteModal(id) {
            deleteId = id;
            const element =
document.querySelector(`[data-
id="${id}"]`);
            const nama = element
? element.getAttribute("data-
name") : "user ini";

document.getElementById("deleteNa
maUser").textContent = nama;

document.getElementById("deleteMo
dal").classList.remove("hidden");

document.body.style.overflow =
"hidden";
        }

        function
closeDeleteModal() {

document.getElementById("deleteMo
dal").classList.add("hidden");

document.body.style.overflow =
"auto";
            deleteId = null;
        }

        async function
confirmDelete() {
            if (!deleteId) {
                alert('ID user
tidak ditemukan');
                return;
            }

            const deleteBtn =
document.getElementById("confirmD
eleteBtn");
            const deleteText =
document.getElementById("deleteTe
xt");
            const deleteLoading =
document.getElementById("deleteLo
ading");
```

```
            deleteBtn.disabled =
true;

deleteText.textContent =
"Menghapus...";

deleteLoading.classList.remove("h
idden");

            try {
                const csrf =
document.querySelector('meta[name
="csrf-
token"]')?.getAttribute("content"
);

                if (!csrf) {
                    throw new
Error('CSRF token tidak
ditemukan');
                }

                const response =
await
fetch(`/admin/users/${deleteId}`,
{
                    method:
"DELETE",
                    headers: {
                        "Content-
Type": "application/json",
                        "X-CSRF-
TOKEN": csrf,
                        "X-
Requested-With":
"XMLHttpRequest",
                        "Accept":
"application/json"
                    }
                });

                const
responseData = await
response.json();

                if (response.ok
&& responseData.success) {

closeDeleteModal();

alert(responseData.message ||
"Data berhasil dihapus!");
                    setTimeout(()
=> location.reload(), 500);
                } else {
                    throw new
Error(responseData.message ||
"Gagal menghapus data");
                }

            } catch (error) {

console.error("Error:", error);
                alert("Terjadi
kesalahan: " + error.message);
            } finally {

deleteBtn.disabled = false;

deleteText.textContent = "Hapus";

deleteLoading.classList.add("hidd
en");
            }
        }

        // Toggle user status
function
        async function
toggleUserStatus(id) {
            try {
                const csrf =
document.querySelector('meta[name
="csrf-
token"]')?.getAttribute("content"
);

                const response =
await
fetch(`/admin/users/${id}/toggle-
status`, {
                    method:
"POST",
                    headers: {
                        "Content-
Type": "application/json",
                        "X-CSRF-
TOKEN": csrf,
                        "X-
Requested-With":
"XMLHttpRequest",
                        "Accept":
"application/json"
                    }
                });

                const
responseData = await
response.json();

                if (response.ok
&& responseData.success) {

alert(responseData.message ||
"Status user berhasil diubah!");
                    setTimeout(()
=> location.reload(), 500);
                } else {
                    throw new
Error(responseData.message ||
"Gagal mengubah status");
                }

            } catch (error) {

console.error("Error:", error);
```

```
                alert("Terjadi
kesalahan: " + error.message);
            }
        }

        // Form submission
handler
        function
handleFormSubmit(form, type) {
            const required = type
=== "add" ? ["name", "email",
"role"] : ["editName",
"editEmail", "editRole"];

            let isValid = true;


required.forEach((fieldId) => {
                const field =
document.getElementById(fieldId);
                if (!field ||
!field.value.trim()) {

field?.classList.add("border-red-
500");
                    isValid =
false;
                } else {

field.classList.remove("border-
red-500");
                }
            });

            // Validate password
for add form
            if (type === "add") {
                const password =
document.getElementById("password
").value;
                const
passwordConfirmation =
document.getElementById("password
_confirmation").value;

                if (!password ||
password.length < 8) {

document.getElementById("password
").classList.add("border-red-
500");
                    isValid =
false;

alert("Password minimal 8
karakter");
                }

                if (password !==
passwordConfirmation) {

document.getElementById("password
_confirmation").classList.add("bo
rder-red-500");
                    isValid =
false;

alert("Konfirmasi password tidak
cocok");
                }
            }

            const status =
form.querySelector('input[name="s
tatus"]:checked');
            if (!status) {
                alert("Mohon
pilih status user.");
                isValid = false;
            }

            if (!isValid) {
                alert("Mohon
lengkapi semua field yang wajib
diisi (bertanda *)");
                return;
            }

            const submitBtn =
document.getElementById(type ===
"add" ? "submitBtn" :
"editSubmitBtn");
            const submitText =
document.getElementById(type ===
"add" ? "submitText" :
"editSubmitText");
            const submitLoading =
document.getElementById(type ===
"add" ? "submitLoading" :
"editSubmitLoading");

            submitBtn.disabled =
true;

submitText.textContent = type ===
"add" ? "Menyimpan..." :
"Mengupdate...";

submitLoading.classList.remove("h
idden");

            const formData = new
FormData(form);

            fetch(form.action, {
                    method:
"POST",
                    body:
formData,
                    headers: {
                        "X-
Requested-With":
"XMLHttpRequest",
                        "X-CSRF-
TOKEN":
```

Penyuluh
dashboard.blade.php

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

@section('title', 'Dashboard
Penyuluh')
@section('page-title', 'Dashboard
Penyuluh')
@section('page-description',
    'Selamat datang kembali, ' .
    (Auth::user()->nama ??
Auth::user()->name) .
    '! Berikut
    ringkasan penyuluhan Anda
hari ini.')

    @push('styles')
        <style>
            .stats-card {
                transition: all
0.3s ease;
            }

            .stats-card:hover {
                transform:
translateY(-2px);
                box-shadow: 0
10px 25px -5px rgba(0, 0, 0,
0.1);
            }

            .weather-gradient {
                background:
linear-gradient(135deg, #74b9ff
0%, #0984e3 100%);
            }

            .health-gradient {
                background:
linear-gradient(135deg, #00b894
0%, #00a085 100%);
            }

            .alert-gradient {
                background:
linear-gradient(135deg, #fd79a8
0%, #e84393 100%);
            }

            .chart-container {
                position:
relative;
                height: 300px;
            }

            .pulse-animation {
                animation: pulse
2s infinite;
            }

            @keyframes pulse {
                0%,
                100% {
                    opacity: 1;
                }

                50% {
                    opacity: 0.7;
                }
            }

            .ternak-card {
                transition: all
0.3s ease;
                border: 2px solid
transparent;
            }

            .ternak-card:hover {
                transform:
translateY(-2px);
                box-shadow: 0 8px
20px rgba(0, 0, 0, 0.1);
                border-color:
#16a34a;
            }

            .weather-info {
                backdrop-filter:
blur(10px);
                background:
rgba(255, 255, 255, 0.1);
            }
        </style>
    @endpush

@section('content')
    <div class="space-y-6">
        <!-- Welcome Section -->
        <div class="bg-gradient-
to-r from-primary to-secondary
rounded-2xl p-6 text-white">
            <div class="flex
items-center justify-between">
                <div>
                    <h2
class="text-2xl font-bold mb-
2">Selamat Datang, {{
Auth::user()->nama ??
Auth::user()->name }}! 👋
                    </h2>
                    <p
class="text-green-100">Hari ini
adalah {{ \Carbon\Carbon::now()-
>isoFormat('dddd, D MMMM Y')
}}</p>
                    <p
class="text-sm text-green-200 mt-
1">Penyuluhan Anda dalam kondisi
baik dan siap produktif!</p>
                </div>
```

```
                <div
class="hidden md:block">
                    <div
class="bg-white/20 backdrop-blur-
sm rounded-xl p-4 text-center">
                        <div
class="text-3xl mb-2">🐄</div>
                        <div
class="text-sm">Total Ternak Desa
Pakeng</div>
                        <div
class="text-2xl font-bold">{{
$totalTernak }}</div>
                    </div>
                </div>
            </div>

            <!-- Quick Stats Cards --
>
            <div class="grid grid-
cols-1 md:grid-cols-2 lg:grid-
cols-4 gap-6">
                <!-- Total Ternak -->
                <div class="stats-
card bg-white rounded-xl p-8
shadow-sm border border-gray-
100">
                    <div class="flex
items-center justify-between">
                        <div>
                            <p
class="text-gray-600 text-sm
font-medium">Total Ternak</p>
                            <p
class="text-3xl font-bold text-
gray-900">{{ $totalTernak ?? 25
}}</p>
                            <p
class="text-green-600 text-sm mt-
1">
                            </p>
                        </div>
                        <div
class="bg-blue-100 p-3 rounded-
full">
                            <svg
class="w-8 h-8 text-blue-600"
fill="none" stroke="currentColor"
viewBox="0 0 24 24">
                                <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
                                    d
="M19 21V5a2 2 0 00-2-2H7a2 2 0
00-2 2v16m14 0h2m-2 0h-5m-9 0H3m2
0h5M9 7h1m-1 4h1m4-4h1m-1 4h1m-5
10v-5a1 1 0 011-1h2a1 1 0 011
1v5m-4 0h4">
                                </pat
h>
                            </svg>
                        </div>
                    </div>
                </div>

                <!-- Ternak Sehat -->
                <div class="stats-
card bg-white rounded-xl p-8
shadow-sm border border-gray-
100">
                    <div class="flex
items-center justify-between">
                        <div>
                            <p
class="text-gray-600 text-sm
font-medium">Ternak Sehat</p>
                            <p
class="text-3xl font-bold text-
gray-900">{{ $ternakSehat ?? 23
}}</p>
                        </div>
                        <div
class="bg-green-100 p-3 rounded-
full">
                            <svg
class="w-8 h-8 text-green-600"
fill="none" stroke="currentColor"
viewBox="0 0 24 24">
                                <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
                                    d
="M4.318 6.318a4.5 4.5 0 000
6.364L12 20.364l7.682-7.682a4.5
4.5 0 00-6.364-6.364L12 7.636l-
1.318-1.318a4.5 4.5 0 00-6.364
0z">
                                </pat
h>
                            </svg>
                        </div>
                    </div>
                </div>

                <!-- Perlu Perhatian
-->
                <div class="stats-
card bg-white rounded-xl p-8
shadow-sm border border-gray-
100">
                    <div class="flex
items-center justify-between">
                        <div>
                            <p
class="text-gray-600 text-sm
font-medium">Perlu Perhatian</p>
                            <p
class="text-3xl font-bold text-
gray-900">{{ $ternakSakit ?? 2
}}</p>
                        </div>
                        <div
class="bg-red-100 p-3 rounded-
full pulse-animation">
```

```
                        <svg
class="w-8 h-8 text-red-600"
fill="none" stroke="currentColor"
viewBox="0 0 24 24">
                            <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
                                d
="M12 9v2m0 4h.01m-6.938
4h13.856c1.54 0 2.502-1.667
1.732-2.5L13.732 4c-.77-.833-
1.664-.833-2.464 0L4.35 16.5c-
.77.833.192 2.5 1.732 2.5z">
                            </pat
h>
                        </svg>
                    </div>
                </div>
            </div>

            <!-- Konsultasi Aktif
-->
            <div class="stats-
card bg-white rounded-xl p-8
shadow-sm border border-gray-
100">
                <div class="flex
items-center justify-between">
                    <div>
                        <p
class="text-gray-600 text-sm
font-medium">Konsultasi Aktif</p>
                        <p
class="text-3xl font-bold text-
gray-900">{{ $konsultasiAktif ??
3 }}</p>
                    </div>
                    <div
class="bg-purple-100 p-3 rounded-
full">
                        <svg
class="w-8 h-8 text-purple-600"
fill="none" stroke="currentColor"
viewBox="0 0 24 24">
                            <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
                                d
="M8 12h.01M12 12h.01M16
12h.01M21 12c0 4.418-4.03 8-9
8a9.863 9.863 0 01-4.255-.949L3
20l1.395-3.72C3.512 15.042 3
13.574 3 12c0-4.418 4.03-8 9-8s9
3.582 9 8z">
                            </pat
h>
                        </svg>
                    </div>
                </div>
            </div>
        </div>
```

```
        <!-- Main Content Grid:
Ternak Terbaru (Kiri) dan Cuaca
(Kanan) -->
        <div class="grid lg:grid-
cols-3 gap-6">
            <!-- Ternak Terbaru -
Kiri (2 kolom) -->
            <div class="lg:col-
span-2">
                <div class="bg-
white rounded-xl p-6 shadow-sm
border border-gray-100 h-full">
                    <div
class="flex items-center justify-
between mb-6">
                        <h3
class="text-lg font-semibold
text-gray-900 flex items-center">
                            <span
class="mr-2">🐄</span>
                            Terna
k Terbaru
                        </h3>
                        <a
href="{{ route('penyuluh.ternak')
}}"
                            class
="text-sm text-primary
hover:text-secondary font-medium
transition-colors">
                            Lihat
Semua →
                        </a>
                    </div>

                    <div
class="grid grid-cols-1 sm:grid-
cols-2 lg:grid-cols-3 gap-4">
                        @foreach
($ternaks as $ternak)
                            <div
class="ternak-card border border-
gray-200 rounded-lg p-4">
                                <
div class="flex items-center
justify-between mb-3">

  <span class="text-xl">🐄</span>

  <s

      class="inline-flex items-
center px-2.5 py-0.5 rounded-full
text-xs font-medium

      {{ $ternak->status ===
'sehat' ? 'bg-green-100 text-
green-800' : 'bg-red-100 text-
red-800' }}">

      {{ ucfirst($ternak->status
?? 'Sehat') }}
```

```
 linecap="round" stroke-
linejoin="round" stroke-width="2"

              d="M2.458 12C3.732
7.943 7.523 5 12 5c4.478 0 8.268
2.943 9.542 7-1.274 4.057-5.064
7-9.542 7-4.477 0-8.268-2.943-
9.542-7z">

          </path>

      </svg>

      Detail

  </button>



  <button

      onclick="confirmDeleteRepor
t({{ $laporan->id }}, '{{
$laporan->ternak->namaTernak ??
'Ternak' }}')"

      class="px-3 py-2 bg-red-600
text-white rounded text-sm
hover:bg-red-700 transition-
colors"

      title="Hapus Laporan">

      <svg class="w-4 h-4"
fill="none" stroke="currentColor"
viewBox="0 0 24 24">

          <path stroke-
linecap="round" stroke-
linejoin="round" stroke-width="2"

              d="M19 7l-.867
12.142A2 2 0 0116.138 21H7.862a2
2 0 01-1.995-1.858L5 7m5 4v6m4-
6v6m1-10V4a1 1 0 00-1-1h-4a1 1 0
00-1 1v3M4 7h16">

          </path>

      </svg>

  </button>
                                  <
/div>
                              </div
>
                          </div>
                      @empty
                          <!--
Empty state tetap sama -->
                          <div
class="col-span-full text-center
py-12">
                              <svg
class="mx-auto h-12 w-12 text-
gray-400" fill="none"
stroke="currentColor"
                                  v
iewBox="0 0 24 24">
                                  <
path stroke-linecap="round"
stroke-linejoin="round" stroke-
width="2"

  d="M9 12h6m-6 4h6m2 5H7a2 2 0
01-2-2V5a2 2 0 012-2h5.586a1 1 0
01.707.293l5.414 5.414a1 1 0
01.293.707V19a2 2 0 01-2 2z">
                                  <
/path>
                              </svg
>
                              <h3
class="mt-2 text-lg font-medium
text-gray-900">Belum Ada
Laporan</h3>
                              <p
class="mt-1 text-gray-500">Belum
ada laporan kesehatan yang
tersedia.</p>
                              <butt
on onclick="switchTab('form')"
                                  c
lass="mt-4 px-4 py-2 bg-green-600
text-white rounded-lg hover:bg-
green-700">
                                  B
uat Laporan Pertama
                              </but
ton>
                          </div>
                      @endforelse
                  </div>
              </div>
          </div>
      </div>
```

Peternak
konsultasi.blade.php

```
@section('content')
    <div class="space-y-6">
        <div class="flex flex-col
sm:flex-row sm:items-center
sm:justify-between gap-4">
            <div class="flex
items-center space-x-4">
                <div
class="relative">
                    <div
class="absolute inset-y-0 left-0
```

```
pl-3 flex items-center pointer-
events-none">
                        <svg
class="h-5 w-5 text-gray-400"
fill="none" stroke="currentColor"
viewBox="0 0 24 24">
                            <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
                                d
="M21 21l-6-6m2-5a7 7 0 11-14 0 7
7 0 0114 0z"></path>
                        </svg>
                    </div>
                    <input
type="text" id="searchInput"
                        class="bl
ock w-full pl-10 pr-3 py-2 border
border-gray-300 rounded-lg
focus:outline-none focus:ring-2
focus:ring-primary focus:border-
transparent"
                        placehold
er="Cari konsultasi..."
onkeyup="searchKonsultasi()">
                </div>
            </div>

            <div class="flex
items-center space-x-3">
                <select
id="statusFilter"
onchange="filterByStatus()"
                    class="block
px-3 py-2 border border-gray-300
rounded-lg focus:outline-none
focus:ring-2 focus:ring-primary
focus:border-transparent">
                    <option
value="">Semua Status</option>
                    <option
value="pending">Pending</option>
                    <option
value="berlangsung">Berlangsung</
option>
                    <option
value="selesai">Selesai</option>
                    <option
value="dibatalkan">Dibatalkan</op
tion>
                </select>

                <select
id="categoryFilter"
onchange="filterByCategory()"
                    class="block
px-3 py-2 border border-gray-300
rounded-lg focus:outline-none
focus:ring-2 focus:ring-primary
focus:border-transparent">
                    <option
value="">Semua Kategori</option>
                    <option
value="kesehatan">Kesehatan</opti
on>
                    <option
value="nutrisi">Nutrisi</option>
                    <option
value="breeding">Breeding</option
>
                    <option
value="perawatan">Perawatan</opti
on>
                    <option
value="umum">Umum</option>
                </select>

                <button
onclick="openKonsultasiModal()"
                    class="inline
-flex items-center px-4 py-2 bg-
primary text-white rounded-lg
text-sm font-medium hover:bg-
secondary focus:outline-none
focus:ring-2 focus:ring-primary
transform hover:scale-105
transition-all">
                    <svg
class="h-4 w-4 mr-2" fill="none"
stroke="currentColor" viewBox="0
0 24 24">
                        <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
                            d="M8
12h.01M12 12h.01M16 12h.01M21
12c0 4.418-4.03 8-9 8a9.863 9.863
0 01-4.255-.949L3 20l1.395-
3.72C3.512 15.042 3 13.574 3
12c0-4.418 4.03-8 9-8s9 3.582 9
8z">
                        </path>
                    </svg>
                    Konsultasi
Baru
                </button>
            </div>
        </div>

        <div class="grid grid-
cols-1 md:grid-cols-4 gap-4">
            <div class="bg-white
rounded-xl p-6 shadow-sm border
border-gray-100">
                <div class="flex
items-center justify-between">
                    <div>
                        <p
class="text-sm font-medium text-
gray-600">Total Konsultasi</p>
                        <p
class="text-2xl font-bold text-
gray-900">{{ $totalKonsultasi ??
24 }}</p>
                    </div>
```

```
                    <div
class="bg-blue-100 p-3 rounded-
full">
                        <svg
class="w-6 h-6 text-blue-600"
fill="none" stroke="currentColor"
viewBox="0 0 24 24">
                            <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
                                d
="M8 12h.01M12 12h.01M16
12h.01M21 12c0 4.418-4.03 8-9
8a9.863 9.863 0 01-4.255-.949L3
20l1.395-3.72C3.512 15.042 3
13.574 3 12c0-4.418 4.03-8 9-8s9
3.582 9 8z">
                            </pat
h>
                        </svg>
                    </div>
                </div>
            </div>

            <div class="bg-white
rounded-xl p-6 shadow-sm border
border-gray-100">
                <div class="flex
items-center justify-between">
                    <div>
                        <p
class="text-sm font-medium text-
gray-600">Berlangsung</p>
                        <p
class="text-2xl font-bold text-
green-600">{{
$konsultasiBerlangsung ?? 3
}}</p>
                    </div>
                    <div
class="bg-green-100 p-3 rounded-
full">
                        <svg
class="w-6 h-6 text-green-600"
fill="none" stroke="currentColor"
viewBox="0 0 24 24">
                            <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
                                d
="M9 12l2 2 4-4m6 2a9 9 0 11-18 0
9 9 0 0118 0z">
                            </pat
h>
                        </svg>
                    </div>
                </div>
            </div>

            <div class="bg-white
rounded-xl p-6 shadow-sm border
border-gray-100">
                <div class="flex
items-center justify-between">
                    <div>
                        <p
class="text-sm font-medium text-
gray-600">Selesai</p>
                        <p
class="text-2xl font-bold text-
purple-600">{{ $konsultasiSelesai
?? 18 }}</p>
                    </div>
                    <div
class="bg-purple-100 p-3 rounded-
full">
                        <svg
class="w-6 h-6 text-purple-600"
fill="none" stroke="currentColor"
viewBox="0 0 24 24">
                            <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
                                d
="M9 5H7a2 2 0 00-2 2v10a2 2 0
002 2h8a2 2 0 002-2V7a2 2 0 00-2-
2h-2M9 5a2 2 0 002 2h2a2 2 0 002-
2M9 5a2 2 0 012-2h2a2 2 0 012 2m-
6 9l2 2 4-4">
                            </pat
h>
                        </svg>
                    </div>
                </div>
            </div>

            <div class="bg-white
rounded-xl p-6 shadow-sm border
border-gray-100">
                <div class="flex
items-center justify-between">
                    <div>
                        <p
class="text-sm font-medium text-
gray-600">Pending</p>
                        <p
class="text-2xl font-bold text-
yellow-600">{{ $konsultasiPending
?? 3 }}</p>
                    </div>
                    <div
class="bg-yellow-100 p-3 rounded-
full">
                        <svg
class="w-6 h-6 text-yellow-600"
fill="none" stroke="currentColor"
viewBox="0 0 24 24">
                            <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
                                d
="M12 8v4l3 3m6-3a9 9 0 11-18 0 9
9 0 0118 0z">
                            </pat
h>
```

```
                    </svg>
                </div>
            </div>
        </div>
    </div>

    <div class="bg-white
rounded-xl shadow-sm border
border-gray-100 overflow-hidden">
        <div class="px-6 py-4
border-b border-gray-200">
            <h3 class="text-
lg font-semibold text-gray-
900">Riwayat Konsultasi</h3>
        </div>

        <div class="divide-y
divide-gray-200">
            @forelse
($konsultasis ?? [] as $item)
                <div
class="p-6 hover:bg-gray-50
transition-colors konsultasi-
item"
                    data-
judul="{{ $item->judul_konsultasi
}}" data-kategori="{{ $item-
>kategori }}"
                    data-
jenis="{{ $item->ternak->jenis
}}" data-status="{{ $item-
>ternak->status }}"
                    data-
status="{{ $item->status }}"
data-id="{{ $item->idKonsultasi
}}"
                    data-
penyuluh="{{ $item->penyuluh ?
$item->penyuluh->nama : 'Belum
ditentukan' }}"
                    data-
ternak="{{ $item->ternak ? $item-
>ternak->namaTernak : 'Ternak
tidak ditemukan' }}"
                    data-
ternak-id="{{ $item->ternak ?
$item->ternak->idTernak : 'N/A'
}}"
                    data-
deskripsi="{{ $item->deskripsi
}}" data-foto="{{ $item-
>foto_ternak }}"
                    data-
created="{{ $item->created_at-
>format('d M Y H:i') }}">
                    <div
class="flex items-center justify-
between">
                        <div
class="flex-1">
                            <
div class="flex items-center
space-x-3 mb-2">
  <h4 class="text-lg font-
semibold text-gray-900">{{ $item-
>judul_konsultasi }}</h4>

  <span class="category-tag">{{
ucfirst($item->kategori)
}}</span>

  <span class="status-badge
status-{{ $item->status }}">

      @switch($item->status)

          @case('pending')

              Pending

          @break


          @case('berlangsung')

              Berlangsung

          @break


          @case('selesai')

              Selesai

          @break


          @case('dibatalkan')

              Dibatalkan

          @break

      @endswitch

  </span>
                            <
/div>

                            <
div class="flex items-center
space-x-6 text-sm text-gray-600
mb-3">

  <div class="flex items-center
space-x-2">

      @if ($item->penyuluh)

      @else

          <div class="flex items-
center space-x-2">

              <div
```

```
      class="max-w-full max-h-96
rounded-lg shadow-md object-
cover">
                                  <
/div>
                              </div
>

                          </div>

                          <!--
Right Column - Sidebar Info -->
                          <div
class="space-y-6">
                              <!--
Info Penyuluh -->
                              <div
class="bg-white border border-
gray-200 rounded-lg p-6">
                                  <
h5 class="text-lg font-semibold
text-gray-900 mb-4 flex items-
center">

  <svg class="w-5 h-5 mr-2 text-
indigo-600" fill="none"
stroke="currentColor"

      viewBox="0 0 24 24">

      <path stroke-
linecap="round" stroke-
linejoin="round" stroke-width="2"

          d="M16 7a4 4 0 11-8 0 4
4 0 018 0zM12 14a7 7 0 00-7
7h14a7 7 0 00-7-7z"></path>

  </svg>

  Penyuluh/Ahli
                                  <
/h5>
                                  <
div id="detailPenyuluh"
class="flex items-center space-x-
3 mb-4">


                                  <
/div>
                              </div
>

                              <!--
Info Ternak -->
                              <div
class="bg-white border border-
gray-200 rounded-lg p-6">
                                  <
h5 class="text-lg font-semibold
text-gray-900 mb-4 flex items-
center">

  <span id="detailTernakEmoji"
class="text-2xl mr-2">🐄</span>

  Info Ternak
                                  <
/h5>
                                  <
div id="detailTernak"
class="space-y-3">

  <!-- Ternak info will be
populated here -->
                                  <
/div>
                              </div
>
                          </div>
                      </div>
                  </div>
              </div>
          </div>
      @endsection
```

ternak.blade.php

```
@section('content')
    <div class="space-y-6">
        <div class="flex flex-col
sm:flex-row sm:items-center
sm:justify-between gap-4">
            <div class="flex
items-center space-x-4">
                <div
class="relative">
                    <div
class="absolute inset-y-0 left-0
pl-3 flex items-center pointer-
events-none">
                        <svg
class="h-5 w-5 text-gray-400"
fill="none" stroke="currentColor"
viewBox="0 0 24 24">
                            <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
                              d
="M21 21l-6-6m2-5a7 7 0 11-14 0 7
7 0 0114 0z"></path>
                        </svg>
                    </div>
                    <input
type="text" id="searchInput"
                        class="se
arch-box block w-full pl-10 pr-3
py-2 border border-gray-300
rounded-lg focus:outline-none
focus:ring-2 focus:ring-primary
focus:border-transparent"
                        placehold
er="Cari ternak..."
onkeyup="searchTernak()">
```

```
                </div>

                <select
id="statusFilter"
                    class="block
px-3 py-2 border border-gray-300
rounded-lg focus:outline-none
focus:ring-2 focus:ring-primary
focus:border-transparent"
                    onchange="fil
terByStatus()">
                    <option
value="">Semua Status</option>
                    <option
value="sehat">Sehat</option>
                    <option
value="sakit">Sakit</option>
                    <option
value="perawatan">Perawatan</opti
on>
                </select>
            </div>

            <div class="flex
items-center space-x-3">
                <button
onclick="openAddModal()"
                    class="inline
-flex items-center px-4 py-2 bg-
primary text-white rounded-lg
text-sm font-medium hover:bg-
secondary focus:outline-none
focus:ring-2 focus:ring-primary
transform hover:scale-105
transition-all">
                    <svg
class="h-4 w-4 mr-2" fill="none"
stroke="currentColor" viewBox="0
0 24 24">
                        <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
                            d="M1
2 6v6m0 0v6m0-6h6m-6 0H6"></path>
                    </svg>
                    Tambah Ternak
                </button>
            </div>
        </div>

        <div class="grid grid-
cols-1 md:grid-cols-4 gap-4">
            <div class="bg-white
rounded-xl p-6 shadow-sm border
border-gray-100">
                <div class="flex
items-center justify-between">
                    <div>
                        <p
class="text-sm font-medium text-
gray-600">Total Ternak</p>
                        <p
class="text-2xl font-bold text-
gray-900">{{ $totalTernak ?? 25
}}</p>
                    </div>
                    <div
class="bg-blue-100 p-3 rounded-
full">
                        <svg
class="w-6 h-6 text-blue-600"
fill="none" stroke="currentColor"
viewBox="0 0 24 24">
                            <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
                                d
="M19 21V5a2 2 0 00-2-2H7a2 2 0
00-2 2v16m14 0h2m-2 0h-5m-9 0H3m2
0h5M9 7h1m-1 4h1m4-4h1m-1 4h1m-5
10v-5a1 1 0 011-1h2a1 1 0 011
1v5m-4 0h4">
                            </pat
h>
                        </svg>
                    </div>
                </div>
            </div>

            <div class="bg-white
rounded-xl p-6 shadow-sm border
border-gray-100">
                <div class="flex
items-center justify-between">
                    <div>
                        <p
class="text-sm font-medium text-
gray-600">Sehat</p>
                        <p
class="text-2xl font-bold text-
green-600">{{ $ternakSehat ?? 21
}}</p>
                    </div>
                    <div
class="bg-green-100 p-3 rounded-
full">
                        <svg
class="w-6 h-6 text-green-600"
fill="none" stroke="currentColor"
viewBox="0 0 24 24">
                            <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
                                d
="M4.318 6.318a4.5 4.5 0 000
6.364L12 20.364l7.682-7.682a4.5
4.5 0 00-6.364-6.364L12 7.636l-
1.318-1.318a4.5 4.5 0 00-6.364
0z">
                            </pat
h>
                        </svg>
                    </div>
                </div>
            </div>
```

```
            <div class="bg-white
rounded-xl p-6 shadow-sm border
border-gray-100">
                <div class="flex
items-center justify-between">
                    <div>
                        <p
class="text-sm font-medium text-
gray-600">Sakit</p>
                        <p
class="text-2xl font-bold text-
red-600">{{ $ternakSakit ?? 2
}}</p>
                    </div>
                    <div
class="bg-red-100 p-3 rounded-
full">
                        <svg
class="w-6 h-6 text-red-600"
fill="none" stroke="currentColor"
viewBox="0 0 24 24">
                            <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
                                d
="M12 9v2m0 4h.01m-6.938
4h13.856c1.54 0 2.502-1.667
1.732-2.5L13.732 4c-.77-.833-
1.664-.833-2.464 0L4.35 16.5c-
.77.833.192 2.5 1.732 2.5z">
                            </pat
h>
                        </svg>
                    </div>
                </div>
            </div>

            <div class="bg-white
rounded-xl p-6 shadow-sm border
border-gray-100">
                <div class="flex
items-center justify-between">
                    <div>
                        <p
class="text-sm font-medium text-
gray-600">Perawatan</p>
                        <p
class="text-2xl font-bold text-
yellow-600">{{ $ternakPerawatan
?? 2 }}</p>
                    </div>
                    <div
class="bg-yellow-100 p-3 rounded-
full">
                        <svg
class="w-6 h-6 text-yellow-600"
fill="none" stroke="currentColor"
viewBox="0 0 24 24">
                            <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
                                d
="M12 6V4m0 2a2 2 0 100 4m0-4a2 2
0 110 4m-6 8a2 2 0 100-4m0 4a2 2
0 100 4m0-4v2m0-6V4m6 6v10m6-2a2
2 0 100-4m0 4a2 2 0 100 4m0-
4v2m0-6V4">
                            </pat
h>
                        </svg>
                    </div>
                </div>
            </div>
        </div>

        <div class="flex items-
center justify-between">
            <div class="flex
items-center space-x-2">
                <span
class="text-sm text-gray-
600">Tampilan:</span>
                <div class="flex
bg-gray-100 rounded-lg p-1">
                    <button
onclick="switchView('grid')"
id="gridViewBtn"
                        class="px
-3 py-1 text-sm font-medium
rounded-md transition-colors
view-btn-active">
                        <svg
class="w-4 h-4 mr-1 inline"
fill="none" stroke="currentColor"
viewBox="0 0 24 24">
                            <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
                                d
="M4 6a2 2 0 012-2h2a2 2 0 012
2v2a2 2 0 01-2 2H6a2 2 0 01-2-
2V6zM14 6a2 2 0 012-2h2a2 2 0 012
2v2a2 2 0 01-2 2h-2a2 2 0 01-2-
2V6zM4 16a2 2 0 012-2h2a2 2 0 012
2v2a2 2 0 01-2 2H6a2 2 0 01-2-2v-
2zM14 16a2 2 0 012-2h2a2 2 0 012
2v2a2 2 0 01-2 2h-2a2 2 0 01-2-
2v-2z">
                            </pat
h>
                        </svg>
                        Grid
                    </button>
                    <button
onclick="switchView('table')"
id="tableViewBtn"
                        class="px
-3 py-1 text-sm font-medium
rounded-md transition-colors
text-gray-600 hover:text-gray-
900">
                        <svg
class="w-4 h-4 mr-1 inline"
fill="none" stroke="currentColor"
viewBox="0 0 24 24">
```

```
                            <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
                                d
="M3 10h18M3 14h18m-9-4v8m-7
0V4a1 1 0 011-1h3M8 3v2m0 0V3m0
2h8m0-2v2m0-2V3m0 2h3a1 1 0 011
1v16">
                            </pat
h>
                        </svg>
                        Tabel
                    </button>
                </div>
            </div>

            <div class="text-sm
text-gray-600">
                Menampilkan {{
$ternakList->count() ?? 10 }}
dari {{ $totalTernak ?? 25 }}
ternak
            </div>
        </div>

        <!-- Grid View -->
        <div id="gridView"
class="grid grid-cols-1 md:grid-
cols-2 lg:grid-cols-3 xl:grid-
cols-4 gap-6">
            @forelse ($ternakList
as $index => $ternak)
                <div class="bg-
white rounded-xl shadow-sm border
border-gray-100 card-hover
ternak-card"
                    data-name="{{
$ternak->namaTernak ?? 'Sapi #' .
sprintf('%03d', $index + 1) }}"
                    data-
status="{{ $ternak->status ??
'sehat' }}" data-id="{{ $ternak-
>idTernak ?? $index + 1 }}"
                    data-
jenis="{{ $ternak->jenis ??
['Sapi Limosin', 'Sapi Brahman',
'Sapi Angus', 'Sapi
Simental'][$index % 4] }}"
                    data-umur="{{
$ternak->tanggalLahir ?
\Carbon\Carbon::parse($ternak-
>tanggalLahir)->age : rand(1, 5)
}}"
                    data-
berat="{{ $ternak->berat ??
rand(200, 500) }}"
                    data-tanggal-
lahir="{{ $ternak->tanggalLahir
?? date('Y-m-d', strtotime('-' .
rand(1, 5) . ' years')) }}"
                    data-
kelamin="{{ $ternak-
>jenis_kelamin ?? ['Jantan',
'Betina'][rand(0, 1)] }}"
                    data-asal="{{
$ternak->asal ?? 'Pembelian' }}"
                    data-
keterangan="{{ $ternak-
>keterangan ?? 'Tidak ada
keterangan khusus' }}"
                    data-foto="{{
$ternak->fotoTernak }}">

                    <div
class="p-6">
                        <div
class="flex items-center justify-
between mb-4">
                            <div
class="text-3xl">🐄</div>
                            <span
class="status-badge status-{{
$ternak->status ?? 'sehat' }}">
                                {
{ ucfirst($ternak->status ??
'sehat') }}
                            </spa
n>
                        </div>

                        <div
class="space-y-2">
                            <h3
class="text-lg font-semibold
text-gray-900">
                                {
{ $ternak->namaTernak ?? 'Sapi #'
. sprintf('%03d', $index + 1) }}
                            </h3>
                            <p
class="text-sm text-gray-600">
                                {
{ $ternak->jenis ?? ['Sapi
Limosin', 'Sapi Brahman', 'Sapi
Angus', 'Sapi Simental'][$index %
4] }}
                            </p>
                            <div
class="flex items-center text-sm
text-gray-500">
                                <
svg class="w-4 h-4 mr-1"
fill="none" stroke="currentColor"
viewBox="0 0 24 24">

  <path stroke-linecap="round"
stroke-linejoin="round" stroke-
width="2"

      d="M8 7V3m8 4V3m-9 8h10M5
21h14a2 2 0 002-2V7a2 2 0 00-2-
2H5a2 2 0 00-2 2v12a2 2 0 002
2z">
```

```
                                d
ata-umur="{{ $ternak-
>tanggalLahir ?
\Carbon\Carbon::parse($ternak-
>tanggalLahir)->age : rand(1, 5)
}}"
                                d
ata-berat="{{ $ternak->berat ??
rand(200, 500) }}"
                                d
ata-tanggal-lahir="{{ $ternak-
>tanggalLahir ?? date('Y-m-d',
strtotime('-' . rand(1, 5) . '
years')) }}"
                                d
ata-kelamin="{{ $ternak-
>jenis_kelamin ?? ['Jantan',
'Betina'][rand(0, 1)] }}"
                                d
ata-asal="{{ $ternak->asal ??
'Pembelian' }}"
                                d
ata-keterangan="{{ $ternak-
>keterangan ?? 'Tidak ada
keterangan khusus' }}">

                                <
td class="px-6 py-4 whitespace-
nowrap">

  <div class="flex items-center">

      <div class="text-2xl mr-
3">🐄</div>

      <div>

          <div class="text-sm
font-medium text-gray-900">{{
$ternak->namaTernak }}</div>

          <div class="text-sm
text-gray-500">ID: #{{ $ternak-
>idTernak ?? $index + 1 }}

          </div>

      </div>

  </div>
                                <
/td>
                                <
td class="px-6 py-4 whitespace-
nowrap text-sm text-gray-900">

  @if ($ternak->tanggalLahir)

      {{
\Carbon\Carbon::parse($ternak-
>tanggalLahir)->age }} tahun

  @else

      {{ rand(1, 5) }} tahun

  @endif
                                <
/td>
                                <
td class="px-6 py-4 whitespace-
nowrap text-sm text-gray-900">

  {{ $ternak->berat ?? rand(200,
500) }} kg
                                <
/td>
                                <
td class="px-6 py-4 whitespace-
nowrap">

  <span class="status-badge
status-{{ $ternak->status }}">

      {{ ucfirst($ternak->status)
}}

  </span>
                                <
/td>
                                <
td class="px-6 py-4 whitespace-
nowrap text-sm font-medium">

  <div class="flex space-x-2">

      <button
onclick="openDetailModal(this)"

          class="action-btn bg-
blue-100 text-blue-600 hover:bg-
blue-200"

          title="Lihat Detail">

          <svg class="w-4 h-4"
fill="none" stroke="currentColor"

              viewBox="0 0 24
24">

              <path stroke-
linecap="round" stroke-
linejoin="round" stroke-width="2"

                  d="M15 12a3 3 0
11-6 0 3 3 0 016 0z" />

              <path stroke-
linecap="round" stroke-
linejoin="round" stroke-width="2"

                  d="M2.458
12C3.732 7.943 7.523 5 12 5c4.478
```

```
0 8.268 2.943 9.542 7-1.274
4.057-5.064 7-9.542 7-4.477 0-
8.268-2.943-9.542-7z" />

          </svg>

      </button>

      <button
onclick="openEditModal(this)"

          class="action-btn bg-
green-100 text-green-600
hover:bg-green-200"

          title="Edit">

          <svg class="w-4 h-4"
fill="none" stroke="currentColor"

              viewBox="0 0 24
24">

              <path stroke-
linecap="round" stroke-
linejoin="round" stroke-width="2"

                  d="M11 5H6a2 2
0 00-2 2v11a2 2 0 002 2h11a2 2 0
002-2v-5m-1.414-9.414a2 2 0
112.828 2.828L11.828 15H9v-
2.828l8.586-8.586z" />

          </svg>

      </button>

      <button
onclick="openDeleteModal('{{
$ternak->idTernak ?? $index + 1
}}')"

          class="action-btn bg-
red-100 text-red-600 hover:bg-
red-200" title="Hapus"

          type="button">

          <svg class="w-4 h-4"
fill="none" stroke="currentColor"

              viewBox="0 0 24
24">

              <path stroke-
linecap="round" stroke-
linejoin="round" stroke-width="2"

                  d="M19 7l-.867
12.142A2 2 0 0116.138 21H7.862a2
2 0 01-1.995-1.858L5 7m5 4v6m4-
6v6m1-10V4a1 1 0 00-1-1h-4a1 1 0
00-1 1v3M4 7h16" />

          </svg>

      </button>

  </div>
                                  <
/td>
                              </tr>
                          @empty
                              <tr>
                                  <
td colspan="5" class="text-center
px-6 py-4 text-gray-500">Tidak
ada data ternak

  tersedia.</td>
                              </tr>
                          @endforel
se
                      </tbody>
                  </table>
              </div>
          </div>

          <!-- Pagination -->
          @if (isset($ternakList)
&& method_exists($ternakList,
'links'))
              <div class="flex
justify-center">
                  {{ $ternakList-
>links() }}
              </div>
          @else
              <div class="flex
justify-center">
                  <nav
class="relative z-0 inline-flex
rounded-md shadow-sm -space-x-
px">
                      <a href="#"
                          class="re
lative inline-flex items-center
px-2 py-2 rounded-l-md border
border-gray-300 bg-white text-sm
font-medium text-gray-500
hover:bg-gray-50">
                          <span
class="sr-only">Previous</span>
                          <svg
class="h-5 w-5"
fill="currentColor" viewBox="0 0
20 20">
                              <path
fill-rule="evenodd"
                                  d
="M12.707 5.293a1 1 0 010
1.414L9.414 10l3.293 3.293a1 1 0
01-1.414 1.414l-4-4a1 1 0 010-
1.414l4-4a1 1 0 011.414 0z"
                                  c
lip-rule="evenodd" />
```

```
                        </svg>
                    </a>
                    <a href="#"
                        class="re
lative inline-flex items-center
px-4 py-2 border border-gray-300
bg-primary text-sm font-medium
text-white">1</a>
                    <a href="#"
                        class="re
lative inline-flex items-center
px-4 py-2 border border-gray-300
bg-white text-sm font-medium
text-gray-700 hover:bg-gray-
50">2</a>
                    <a href="#"
                        class="re
lative inline-flex items-center
px-4 py-2 border border-gray-300
bg-white text-sm font-medium
text-gray-700 hover:bg-gray-
50">3</a>
                    <a href="#"
                        class="re
lative inline-flex items-center
px-2 py-2 rounded-r-md border
border-gray-300 bg-white text-sm
font-medium text-gray-500
hover:bg-gray-50">
                        <span
class="sr-only">Next</span>
                        <svg
class="h-5 w-5"
fill="currentColor" viewBox="0 0
20 20">
                            <path
fill-rule="evenodd"
                                d
="M7.293 14.707a1 1 0 010-
1.414L10.586 10 7.293 6.707a1 1 0
011.414-1.414l4 4a1 1 0 010
1.414l-4 4a1 1 0 01-1.414 0z"
                                c
lip-rule="evenodd" />
                        </svg>
                    </a>
                </nav>
            </div>
        @endif
    </div>

    <!-- Add Ternak Modal -->
    <div id="addModal"
        class="fixed inset-0 bg-
gray-600 bg-opacity-50 overflow-
y-auto h-full w-full hidden z-50
modal-backdrop">
        <div class="relative top-
10 mx-auto p-5 border w-full max-
w-2xl shadow-lg rounded-lg bg-
white modal-content">
            <div class="flex
items-center justify-between mb-
6">
                <h3 class="text-
xl font-semibold text-gray-
900">Tambah Ternak Baru</h3>
                <button
onclick="closeAddModal()"
class="text-gray-400 hover:text-
gray-600">
                    <svg
class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0
0 24 24">
                        <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
d="M6 18L18 6M6 6l12 12">
                        </path>
                    </svg>
                </button>
            </div>

            <form
id="addTernakForm" action="{{
route('ternak.store') }}"
method="POST"
enctype="multipart/form-data">
                @csrf
                <div class="grid
grid-cols-1 md:grid-cols-2 gap-
6">
                    <div
class="col-span-1 md:col-span-2">
                        <label
for="namaTernak" class="block
text-sm font-medium text-gray-700
mb-2">
                            Nama
Ternak <span class="text-red-
500">*</span>
                        </label>
                        <input
type="text" id="namaTernak"
name="namaTernak" required
                            class
="w-full px-3 py-2 border border-
gray-300 rounded-lg
focus:outline-none focus:ring-2
focus:ring-primary focus:border-
transparent"
                            place
holder="Contoh: Sapi Makmur 001">
                        <p
class="text-xs text-gray-500 mt-
1">Berikan nama unik untuk
memudah identifikasi</p>
                    </div>

                    <div>
                        <label
for="jenis" class="block text-sm
font-medium text-gray-700 mb-2">
```

```
                            Jenis
Ternak <span class="text-red-
500">*</span>
                        </label>
                        <select
id="jenis" name="jenis" required
                            class
="w-full px-3 py-2 border border-
gray-300 rounded-lg
focus:outline-none focus:ring-2
focus:ring-primary focus:border-
transparent">
                            <opti
on value="">Pilih Jenis
Ternak</option>
                            <opti
on value="Sapi Limosin">Sapi
Limosin</option>
                            <opti
on value="Sapi Brahman">Sapi
Brahman</option>
                            <opti
on value="Sapi Angus">Sapi
Angus</option>
                            <opti
on value="Sapi Simental">Sapi
Simental</option>
                            <opti
on value="Sapi Bali">Sapi
Bali</option>
                            <opti
on value="Sapi Ongole">Sapi
Ongole</option>
                            <opti
on value="Kambing Boer">Kambing
Boer</option>
                            <opti
on value="Kambing Etawa">Kambing
Etawa</option>
                            <opti
on
value="Kambing
Jawarandu">Kambing
Jawarandu</option>
                            <opti
on value="Domba Garut">Domba
Garut</option>
                            <opti
on value="Domba Merino">Domba
Merino</option>
                            <opti
on
value="Lainnya">Lainnya</option>
                        </select>
                    </div>

                    <div>
                        <label
for="jenis_kelamin" class="block
text-sm font-medium text-gray-700
mb-2">
                            Jenis
Kelamin <span class="text-red-
500">*</span>
                        </label>
                        <select
id="jenis_kelamin"
name="jenis_kelamin" required
                            class
="w-full px-3 py-2 border border-
gray-300 rounded-lg
focus:outline-none focus:ring-2
focus:ring-primary focus:border-
transparent">
                            <opti
on value="">Pilih Jenis
Kelamin</option>
                            <opti
on value="Jantan">Jantan</option>
                            <opti
on value="Betina">Betina</option>
                        </select>
                    </div>

                    <div>
                        <label
for="berat" class="block text-sm
font-medium text-gray-700 mb-2">
                            Berat
(Kg)
                        </label>
                        <input
type="number" id="berat"
name="berat" min="0" step="0.1"
                            class
="w-full px-3 py-2 border border-
gray-300 rounded-lg
focus:outline-none focus:ring-2
focus:ring-primary focus:border-
transparent"
                            place
holder="0.0">
                    </div>

                    <div>
                        <label
for="tanggalLahir" class="block
text-sm font-medium text-gray-700
mb-2">
                            Tangg
al Lahir
                        </label>
                        <input
type="date" id="tanggalLahir"
name="tanggalLahir" max="{{
date('Y-m-d') }}"
                            class
="w-full px-3 py-2 border border-
gray-300 rounded-lg
focus:outline-none focus:ring-2
focus:ring-primary focus:border-
transparent">
                    </div>

                    <div>
```

```
                        <label
for="asal" class="block text-sm
font-medium text-gray-700 mb-2">
                            Asal/
Sumber Ternak
                        </label>
                        <select
id="asal" name="asal"
                            class
="w-full px-3 py-2 border border-
gray-300 rounded-lg
focus:outline-none focus:ring-2
focus:ring-primary focus:border-
transparent">
                            <opti
on value="">Pilih Asal
Ternak</option>
                            <opti
on
value="Pembelian">Pembelian</opti
on>
                            <opti
on value="Kelahiran">Kelahiran
(Breeding)</option>
                            <opti
on
value="Hibah">Hibah/Bantuan</opti
on>
                            <opti
on
value="Warisan">Warisan</option>
                            <opti
on
value="Lainnya">Lainnya</option>
                        </select>
                    </div>

                    <div
class="col-span-1 md:col-span-2">
                        <label
for="status" class="block text-sm
font-medium text-gray-700 mb-2">
                            Statu
s Kesehatan <span class="text-
red-500">*</span>
                        </label>
                        <div
class="grid grid-cols-3 gap-4">
                            <labe
l
                                c
lass="flex items-center p-3
border border-gray-300 rounded-lg
cursor-pointer hover:bg-gray-50
transition-colors">
                                <
input type="radio" name="status"
value="sehat" checked

  class="h-4 w-4 text-primary
focus:ring-primary border-gray-
300">
                                <
div class="ml-3">

  <div class="flex items-center">

      <span class="w-3 h-3 bg-
green-500 rounded-full mr-
2"></span>

      <span class="text-sm font-
medium text-gray-
900">Sehat</span>

  </div>

  <p class="text-xs text-gray-
500">Kondisi normal</p>
                                <
/div>
                            </lab
el>

                            <labe
l
                                c
lass="flex items-center p-3
border border-gray-300 rounded-lg
cursor-pointer hover:bg-gray-50
transition-colors">
                                <
input type="radio" name="status"
value="sakit"

  class="h-4 w-4 text-primary
focus:ring-primary border-gray-
300">
                                <
div class="ml-3">

  <div class="flex items-center">

      <span class="w-3 h-3 bg-
red-500 rounded-full mr-
2"></span>

      <span class="text-sm font-
medium text-gray-
900">Sakit</span>

  </div>

  <p class="text-xs text-gray-
500">Memerlukan perawatan</p>
                                <
/div>
                            </lab
el>

                            <labe
l
                                c
lass="flex items-center p-3
border border-gray-300 rounded-lg
```

```
cursor-pointer hover:bg-gray-50
transition-colors">
                                <
input type="radio" name="status"
value="perawatan"

  class="h-4 w-4 text-primary
focus:ring-primary border-gray-
300">
                                <
div class="ml-3">

  <div class="flex items-center">

      <span class="w-3 h-3 bg-
yellow-500 rounded-full mr-
2"></span>

      <span class="text-sm font-
medium text-gray-
900">Perawatan</span>

  </div>

  <p class="text-xs text-gray-
500">Dalam pengobatan</p>
                                <
/div>
                            </lab
el>
                        </div>
                    </div>

                    <div
class="col-span-1 md:col-span-2">
                        <label
for="fotoTernak" class="block
text-sm font-medium text-gray-700
mb-2">
                            Foto
Ternak
                        </label>
                        <div
class="flex items-center justify-
center w-full">
                            <labe
l for="fotoTernak"
                                c
lass="flex flex-col items-center
justify-center w-full h-32
border-2 border-gray-300 border-
dashed rounded-lg cursor-pointer
bg-gray-50 hover:bg-gray-100">
                                <
div class="flex flex-col items-
center justify-center pt-5 pb-6">

  <svg class="w-8 h-8 mb-4 text-
gray-500" aria-hidden="true"

      xmlns="http://www.w3.org/20
00/svg" fill="none" viewBox="0 0
20 16">

      <path stroke="currentColor"
stroke-linecap="round" stroke-
linejoin="round"

          stroke-width="2"

          d="M13 13h3a3 3 0 0 0
0-6h-.025A5.56 5.56 0 0 0 16 6.5
5.5 5.5 0 0 0 5.207 5.021C5.137
5.017 5.071 5 5 5a4 4 0 0 0 0
8h2.167M10 15V6m0 0L8 8m2-2 2 2"
/>

  </svg>

  <p class="mb-2 text-sm text-
gray-500"><span class="font-
semibold">Klik untuk

          upload</span> atau drag
& drop</p>

  <p class="text-xs text-gray-
500">PNG, JPG atau JPEG (MAX.
2MB)</p>
                                <
/div>
                                <
input id="fotoTernak"
name="fotoTernak" type="file"
class="hidden" accept="image/*"

  onchange="previewImage(this)">
                            </lab
el>
                        </div>
                        <div
id="imagePreview" class="mt-3
hidden">
                            <img
id="preview" src="" alt="Preview"
                                c
lass="w-full h-32 object-cover
rounded-lg">
                        </div>
                    </div>

                    <div
class="col-span-1 md:col-span-2">
                        <label
for="keterangan" class="block
text-sm font-medium text-gray-700
mb-2">
                            Keter
angan Tambahan
                        </label>
                        <textarea
id="keterangan" name="keterangan"
rows="3"
                            class
="w-full px-3 py-2 border border-
gray-300 rounded-lg
```

```
focus:outline-none focus:ring-2
focus:ring-primary focus:border-
transparent resize-none"
                                place
holder="Catatan khusus tentang
ternak ini..."></textarea>
                            </div>
                        </div>

                        <div class="flex
items-center justify-end space-x-
3 mt-8 pt-6 border-t border-gray-
200">
                            <button
type="button"
onclick="closeAddModal()"
                                class="px
-4 py-2 bg-gray-300 text-gray-700
text-sm font-medium rounded-lg
hover:bg-gray-400 focus:outline-
none focus:ring-2 focus:ring-
gray-300 transition-colors">
                                Batal
                            </button>
                            <button
type="submit" id="submitBtn"
                                class="px
-6 py-2 bg-primary text-white
text-sm font-medium rounded-lg
hover:bg-secondary focus:outline-
none focus:ring-2 focus:ring-
primary transition-colors">
                                <span
id="submitText">Simpan
Ternak</span>
                                <svg
id="submitLoading" class="hidden
animate-spin -mr-1 ml-3 h-4 w-4
text-white inline"
                                    fill=
"none" viewBox="0 0 24 24">
                                    <circ
le class="opacity-25" cx="12"
cy="12" r="10"
stroke="currentColor"
                                        s
troke-width="4"></circle>
                                    <path
class="opacity-75"
fill="currentColor"
                                        d
="M4 12a8 8 0 018-8V0C5.373 0 0
5.373 0 12h4zm2 5.291A7.962 7.962
0 014 12H0c0 3.042 1.135 5.824 3
7.938l3-2.647z">
                                    </pat
h>
                                </svg>
                            </button>
                        </div>
                    </form>
                </div>
            </div>

    <!-- Detail Ternak Modal -->
    <div id="detailModal"
        class="fixed inset-0 bg-
gray-600 bg-opacity-50 overflow-
y-auto h-full w-full hidden z-50
modal-backdrop">
        <div class="relative top-
10 mx-auto p-5 border w-full max-
w-3xl shadow-lg rounded-lg bg-
white modal-content">
            <div class="flex
items-center justify-between mb-
6">
                <h3 class="text-
xl font-semibold text-gray-
900">Detail Ternak</h3>
                <button
onclick="closeDetailModal()"
class="text-gray-400 hover:text-
gray-600">
                    <svg
class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0
0 24 24">
                        <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
d="M6 18L18 6M6 6l12 12">
                        </path>
                    </svg>
                </button>
            </div>

            <div class="grid
grid-cols-1 md:grid-cols-2 gap-
6">
                <div
class="space-y-4">
                    <div>
                        <div
class="detail-label">Nama
Ternak</div>
                        <div
class="detail-value text-lg font-
semibold" id="detailNama">-</div>
                    </div>

                    <div>
                        <div
class="detail-label">ID
Ternak</div>
                        <div
class="detail-value"
id="detailId">-</div>
                    </div>

                    <div>
                        <div
class="detail-label">Jenis
Ternak</div>
```

```
                        <div
class="detail-value"
id="detailJenis">-</div>
                    </div>

                    <div>
                        <div
class="detail-label">Jenis
Kelamin</div>
                        <div
class="detail-value"
id="detailKelamin">-</div>
                    </div>

                    <div>
                        <div
class="detail-label">Umur</div>
                        <div
class="detail-value"
id="detailUmur">-</div>
                    </div>

                    <div>
                        <div
class="detail-label">Berat</div>
                        <div
class="detail-value"
id="detailBerat">-</div>
                    </div>

                    <div>
                        <div
class="detail-label">Tanggal
Lahir</div>
                        <div
class="detail-value"
id="detailTanggalLahir">-</div>
                    </div>

                    <div>
                        <div
class="detail-
label">Asal/Sumber</div>
                        <div
class="detail-value"
id="detailAsal">-</div>
                    </div>

                    <div>
                        <div
class="detail-label">Status
Kesehatan</div>
                        <div
class="detail-value">
                            <span
id="detailStatus" class="status-
badge">-</span>
                        </div>
                    </div>
                </div>

                <div
class="space-y-4">
                    <div>
                        <div
class="detail-label">Foto
Ternak</div>
                        <div
class="detail-value">
                            @if
(isset($ternak) && isset($ternak-
>fotoTernak))
                                <
img id="detailFoto" src="{{
asset('storage/foto_ternak/' .
$ternak->fotoTernak) }}"

  alt="Foto Ternak" class="w-full
h-48 object-cover rounded-lg" />
                            @else
                                <
p class="text-sm text-gray-
500">Tidak ada foto ternak</p>
                            @endi
f
                        </div>
                    </div>
                    <div>
                        <div
class="detail-
label">Keterangan</div>
                        <div
class="detail-value"
id="detailKeterangan">-</div>
                    </div>
                </div>
            </div>

            <div class="flex
items-center justify-end space-x-
3 mt-8 pt-6 border-t border-gray-
200">
                <button
onclick="closeDetailModal()"
                    class="px-4
py-2 bg-gray-300 text-gray-700
text-sm font-medium rounded-lg
hover:bg-gray-400 focus:outline-
none focus:ring-2 focus:ring-
gray-300 transition-colors">
                    Tutup
                </button>

            </div>
        </div>
    </div>

    <!-- Edit Ternak Modal -->
    <div id="editModal"
        class="fixed inset-0 bg-
gray-600 bg-opacity-50 overflow-
y-auto h-full w-full hidden z-50
modal-backdrop">
        <div class="relative top-
10 mx-auto p-5 border w-full max-
```

```
w-2xl shadow-lg rounded-lg bg-
white modal-content">
            <div class="flex
items-center justify-between mb-
6">
                <h3 class="text-
xl font-semibold text-gray-
900">Edit Ternak</h3>
                <button
onclick="closeEditModal()"
class="text-gray-400 hover:text-
gray-600">
                    <svg
class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0
0 24 24">
                        <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
d="M6 18L18 6M6 6l12 12">
                        </path>
                    </svg>
                </button>
            </div>

            <form
id="editTernakForm" method="POST"
enctype="multipart/form-data">
                @csrf
                @method('PUT')
                <input
type="hidden" id="editTernakId"
name="idTernak">

                <div class="grid
grid-cols-1 md:grid-cols-2 gap-
6">
                    <div
class="col-span-1 md:col-span-2">
                        <label
for="editNamaTernak" class="block
text-sm font-medium text-gray-700
mb-2">
                            Nama
Ternak <span class="text-red-
500">*</span>
                        </label>
                        <input
type="text" id="editNamaTernak"
name="namaTernak" required
                            class
="w-full px-3 py-2 border border-
gray-300 rounded-lg
focus:outline-none focus:ring-2
focus:ring-primary focus:border-
transparent"
                            place
holder="Contoh: Sapi Makmur 001">
                    </div>

                    <div>
                        <label
for="editJenis" class="block
text-sm font-medium text-gray-700
mb-2">
                            Jenis
Ternak <span class="text-red-
500">*</span>
                        </label>
                        <select
id="editJenis" name="jenis"
required
                            class
="w-full px-3 py-2 border border-
gray-300 rounded-lg
focus:outline-none focus:ring-2
focus:ring-primary focus:border-
transparent">
                            <opti
on value="">Pilih Jenis
Ternak</option>
                            <opti
on value="Sapi Limosin">Sapi
Limosin</option>
                            <opti
on value="Sapi Brahman">Sapi
Brahman</option>
                            <opti
on value="Sapi Angus">Sapi
Angus</option>
                            <opti
on value="Sapi Simental">Sapi
Simental</option>
                            <opti
on value="Sapi Bali">Sapi
Bali</option>
                            <opti
on value="Sapi Ongole">Sapi
Ongole</option>
                            <opti
on value="Kambing Boer">Kambing
Boer</option>
                            <opti
on value="Kambing Etawa">Kambing
Etawa</option>
                            <opti
on value="Kambing
Jawarandu">Kambing
Jawarandu</option>
                            <opti
on value="Domba Garut">Domba
Garut</option>
                            <opti
on value="Domba Merino">Domba
Merino</option>
                            <opti
on
value="Lainnya">Lainnya</option>
                        </select>
                    </div>

                    <div>
                        <label
for="editJenisKelamin"
class="block text-sm font-medium
text-gray-700 mb-2">
```

```
                            Jenis
Kelamin <span class="text-red-
500">*</span>
                        </label>
                        <select
id="editJenisKelamin"
name="jenis_kelamin" required
                            class
="w-full px-3 py-2 border border-
gray-300 rounded-lg
focus:outline-none focus:ring-2
focus:ring-primary focus:border-
transparent">
                            <opti
on value="">Pilih Jenis
Kelamin</option>
                            <opti
on value="Jantan">Jantan</option>
                            <opti
on value="Betina">Betina</option>
                        </select>
                    </div>

                    <div>
                        <label
for="editBerat" class="block
text-sm font-medium text-gray-700
mb-2">
                            Berat
(Kg)
                        </label>
                        <input
type="number" id="editBerat"
name="berat" min="0" step="0.1"
                            class
="w-full px-3 py-2 border border-
gray-300 rounded-lg
focus:outline-none focus:ring-2
focus:ring-primary focus:border-
transparent"
                            place
holder="0.0">
                    </div>

                    <div>
                        <label
for="editTanggalLahir"
class="block text-sm font-medium
text-gray-700 mb-2">
                            Tangg
al Lahir
                        </label>
                        <input
type="date" id="editTanggalLahir"
name="tanggalLahir" max="{{
date('Y-m-d') }}"
                            class
="w-full px-3 py-2 border border-
gray-300 rounded-lg
focus:outline-none focus:ring-2
focus:ring-primary focus:border-
transparent">
                    </div>

                    <div>
                        <label
for="editAsal" class="block text-
sm font-medium text-gray-700 mb-
2">
                            Asal/
Sumber Ternak
                        </label>
                        <select
id="editAsal" name="asal"
                            class
="w-full px-3 py-2 border border-
gray-300 rounded-lg
focus:outline-none focus:ring-2
focus:ring-primary focus:border-
transparent">
                            <opti
on value="">Pilih Asal
Ternak</option>
                            <opti
on
value="Pembelian">Pembelian</opti
on>
                            <opti
on value="Kelahiran">Kelahiran
(Breeding)</option>
                            <opti
on
value="Hibah">Hibah/Bantuan</opti
on>
                            <opti
on
value="Warisan">Warisan</option>
                            <opti
on
value="Lainnya">Lainnya</option>
                        </select>
                    </div>

                    <div
class="col-span-1 md:col-span-2">
                        <label
for="editStatus" class="block
text-sm font-medium text-gray-700
mb-2">
                            Statu
s Kesehatan <span class="text-
red-500">*</span>
                        </label>
                        <div
class="grid grid-cols-3 gap-4">
                            <labe
l
                                c
lass="flex items-center p-3
border border-gray-300 rounded-lg
cursor-pointer hover:bg-gray-50
transition-colors">
                                <
input type="radio" name="status"
value="sehat"
id="editStatusSehat"
```

```
  class="h-4 w-4 text-primary
focus:ring-primary border-gray-
300">
                                <
div class="ml-3">

  <div class="flex items-center">

      <span class="w-3 h-3 bg-
green-500 rounded-full mr-
2"></span>

      <span class="text-sm font-
medium text-gray-
900">Sehat</span>

  </div>

  <p class="text-xs text-gray-
500">Kondisi normal</p>
                                <
/div>
                            </lab
el>

                            <labe
l
                                c
lass="flex items-center p-3
border border-gray-300 rounded-lg
cursor-pointer hover:bg-gray-50
transition-colors">
                                <
input type="radio" name="status"
value="sakit"
id="editStatusSakit"

  class="h-4 w-4 text-primary
focus:ring-primary border-gray-
300">
                                <
div class="ml-3">

  <div class="flex items-center">

      <span class="w-3 h-3 bg-
red-500 rounded-full mr-
2"></span>

      <span class="text-sm font-
medium text-gray-
900">Sakit</span>

  </div>

  <p class="text-xs text-gray-
500">Memerlukan perawatan</p>
                                <
/div>
                            </lab
el>
                                <labe
l
                                c
lass="flex items-center p-3
border border-gray-300 rounded-lg
cursor-pointer hover:bg-gray-50
transition-colors">
                                <
input type="radio" name="status"
value="perawatan"
id="editStatusPerawatan"

  class="h-4 w-4 text-primary
focus:ring-primary border-gray-
300">
                                <
div class="ml-3">

  <div class="flex items-center">

      <span class="w-3 h-3 bg-
yellow-500 rounded-full mr-
2"></span>

      <span class="text-sm font-
medium text-gray-
900">Perawatan</span>

  </div>

  <p class="text-xs text-gray-
500">Dalam pengobatan</p>
                                <
/div>
                            </lab
el>
                        </div>
                    </div>

                    <div
class="col-span-1 md:col-span-2">
                        <label
for="editFoto" class="block text-
sm font-medium text-gray-700 mb-
2">
                            Foto
Ternak
                        </label>
                        <div
class="flex items-center justify-
center w-full">
                            <labe
l for="editFoto"
                                c
lass="flex flex-col items-center
justify-center w-full h-32
border-2 border-gray-300 border-
dashed rounded-lg cursor-pointer
bg-gray-50 hover:bg-gray-100">
                                <
div class="flex flex-col items-
center justify-center pt-5 pb-6">
```

```
  <svg class="w-8 h-8 mb-4 text-
gray-500" aria-hidden="true"

      xmlns="http://www.w3.org/20
00/svg" fill="none" viewBox="0 0
20 16">

      <path stroke="currentColor"
stroke-linecap="round" stroke-
linejoin="round"

          stroke-width="2"

          d="M13 13h3a3 3 0 0 0
0-6h-.025A5.56 5.56 0 0 0 16 6.5
5.5 5.5 0 0 0 5.207 5.021C5.137
5.017 5.071 5 5 5a4 4 0 0 0 0
8h2.167M10 15V6m0 0L8 8m2-2 2 2"
/>

  </svg>

  <p class="mb-2 text-sm text-
gray-500"><span class="font-
semibold">Klik untuk

          upload</span> atau drag
& drop</p>

  <p class="text-xs text-gray-
500">PNG, JPG atau JPEG (MAX.
2MB)</p>
                                <
/div>
                                <
input id="editFoto"
name="fotoTernak" type="file"
class="hidden" accept="image/*"

  onchange="previewEditImage(this
)">
                                </lab
el>
                            </div>
                            <div
id="editImagePreview" class="mt-3
hidden">
                                <img
id="editPreview" src=""
alt="Preview"
                                c
lass="w-full h-32 object-cover
rounded-lg">
                            </div>
                        </div>

                        <div
class="col-span-1 md:col-span-2">
                            <label
for="editKeterangan" class="block
text-sm font-medium text-gray-700
mb-2">
                                Keter
angan Tambahan
                            </label>
                            <textarea
id="editKeterangan"
name="keterangan" rows="3"
                                class
="w-full px-3 py-2 border border-
gray-300 rounded-lg
focus:outline-none focus:ring-2
focus:ring-primary focus:border-
transparent resize-none"
                                place
holder="Catatan khusus tentang
ternak ini..."></textarea>
                        </div>
                    </div>

                    <div class="flex
items-center justify-end space-x-
3 mt-8 pt-6 border-t border-gray-
200">
                        <button
type="button"
onclick="closeEditModal()"
                            class="px
-4 py-2 bg-gray-300 text-gray-700
text-sm font-medium rounded-lg
hover:bg-gray-400 focus:outline-
none focus:ring-2 focus:ring-
gray-300 transition-colors">
                            Batal
                        </button>
                        <button
type="submit" id="editSubmitBtn"
                            class="px
-6 py-2 bg-primary text-white
text-sm font-medium rounded-lg
hover:bg-secondary focus:outline-
none focus:ring-2 focus:ring-
primary transition-colors">
                            <span
id="editSubmitText">Update
Ternak</span>
                            <svg
id="editSubmitLoading"
class="hidden animate-spin -mr-1
ml-3 h-4 w-4 text-white inline"
                                fill=
"none" viewBox="0 0 24 24">
                                <circ
le class="opacity-25" cx="12"
cy="12" r="10"
stroke="currentColor"
                                    s
troke-width="4"></circle>
                                <path
class="opacity-75"
fill="currentColor"
                                    d
="M4 12a8 8 0 018-8V0C5.373 0 0
5.373 0 12h4zm2 5.291A7.962 7.962
```

```
0 014 12H0c0 3.042 1.135 5.824 3
7.938l3-2.647z">
                            </pat
h>
                        </svg>
                    </button>
                </div>
            </form>
        </div>
    </div>

    <!-- Delete Confirmation
Modal -->
    <div id="deleteModal"
        class="fixed inset-0 bg-
gray-600 bg-opacity-50 overflow-
y-auto h-full w-full hidden z-50
modal-backdrop">
        <div class="relative top-
20 mx-auto p-5 border w-96
shadow-lg rounded-lg bg-white
modal-content">
            <div class="mt-3
text-center">
                <div class="mx-
auto flex items-center justify-
center h-12 w-12 rounded-full bg-
red-100">
                    <svg
class="h-6 w-6 text-red-600"
fill="none" stroke="currentColor"
viewBox="0 0 24 24">
                        <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
                            d="M1
2 9v2m0 4h.01m-6.938
4h13.856c1.54 0 2.502-1.667
1.732-2.5L13.732 4c-.77-.833-
1.664-.833-2.464 0L4.35 16.5c-
.77.833.192 2.5 1.732 2.5z">
                        </path>
                    </svg>
                </div>
                <h3 class="text-
lg font-medium text-gray-900 mt-
4">Hapus Ternak</h3>
                <p class="text-sm
text-gray-500 mt-2">
                    Apakah Anda
yakin ingin menghapus data ternak
<span id="deleteNamaTernak"
                        class="fo
nt-semibold"></span>?
                    Tindakan ini
tidak dapat dibatalkan.
                </p>
                <div
class="items-center px-4 py-3 mt-
4 flex justify-center space-x-2">
                    <button
id="confirmDeleteBtn"
type="button"
                        class="px
-4 py-2 bg-red-600 text-white
text-base font-medium rounded-md
hover:bg-red-700 focus:outline-
none focus:ring-2 focus:ring-red-
500 min-w-[100px] flex items-
center justify-center">
                        <span
id="deleteText">Hapus</span>
                        <svg
id="deleteLoading" class="hidden
animate-spin ml-2 h-4 w-4 text-
white" fill="none"
                            viewB
ox="0 0 24 24">
                            <circ
le class="opacity-25" cx="12"
cy="12" r="10"
stroke="currentColor"
                                s
troke-width="4"></circle>
                            <path
class="opacity-75"
fill="currentColor"
                                d
="M4 12a8 8 0 018-8V0C5.373 0 0
5.373 0 12h4zm2 5.291A7.962 7.962
0 014 12H0c0 3.042 1.135 5.824 3
7.938l3-2.647z">
                            </pat
h>
                        </svg>
                    </button>
                    <button
onclick="closeDeleteModal()"
type="button"
                        class="px
-4 py-2 bg-gray-300 text-gray-900
text-base font-medium rounded-md
hover:bg-gray-400 focus:outline-
none focus:ring-2 focus:ring-
gray-300 min-w-[100px]">
                        Batal
                    </button>
                </div>
            </div>
        </div>
    </div>
@endsection
```

home.blade.php

```
<!-- Hero Section -->
    <section id="home"
class="hero-bg min-h-screen flex
items-center pt-16">
        <div class="max-w-7xl mx-
auto px-4 sm:px-6 lg:px-8">
            <div class="grid
grid-cols-1 lg:grid-cols-2 gap-12
items-center">
                <div class="text-
white">
```

```
                    <h1
class="text-4xl sm:text-5xl
lg:text-6xl font-bold mb-6
leading-tight">
                        Sistem
Manajemen <span class="text-
yellow-300">Peternakan</span>
Modern
                    </h1>
                    <p
class="text-lg sm:text-xl mb-8
text-green-100">
                        Kelola
ternak Anda dengan mudah, pantau
kesehatan, dapatkan konsultasi
ahli, dan tingkatkan
                        produktiv
itas peternakan dengan teknologi
terdepan.
                    </p>
                    <div
class="flex flex-col sm:flex-row
gap-4">
                        <button
                            class
="bg-yellow-400 text-gray-900 px-
6 sm:px-8 py-3 sm:py-4 rounded-lg
font-semibold hover:bg-yellow-300
transition-colors">
                            <a
href="{{ route('admin.register')
}}">
                                M
ulai Sekarang
                            </a>
                        </button>
                        <button
                            class
="border-2 border-white text-
white px-6 sm:px-8 py-3 sm:py-4
rounded-lg font-semibold
hover:bg-white hover:text-primary
transition-colors">
                            <a
href="{{ route('admin.login')
}}">
                                K
elola Ternak Anda
                            </a>
                        </button>
                    </div>
                </div>
                <div
class="hidden lg:block">
                    <div
class="floating-animation">
                        <!--<img
src="images/sapi.jpg" alt="Gambar
Ternak" class="w-full h-96
object-contain">-->
                        <svg
class="w-full h-96 text-green-
200" fill="currentColor"
viewBox="0 0 24 24">
                            <path
                                d
="M12 2C6.48 2 2 6.48 2 12s4.48
10 10 10 10-4.48 10-10S17.52 2 12
2zM7.07 18.28c.43-.9 3.05-1.78
4.93-1.78s4.51.88 4.93 1.78C15.57
19.36 13.86 20 12 20s-3.57-.64-
4.93-1.72zm11.29-1.45c-1.43-1.74-
4.9-2.33-6.36-2.33s-4.93.59-6.36
2.33C4.62 15.49 4 13.82 4 12c0-
4.41 3.59-8 8-8s8 3.59 8 8c0
1.82-.62 3.49-1.64 4.83zM12 6c-
1.94 0-3.5 1.56-3.5 3.5S10.06 13
12 13s3.5-1.56 3.5-3.5S13.94 6 12
6zm0 5c-.83 0-1.5-.67-1.5-
1.5S11.17 8 12 8s1.5.67 1.5
1.5S12.83 11 12 11z" />
                        </svg>
                    </div>
                </div>
            </div>
        </div>
    </section>

    <!-- About Section -->
    <section id="about"
class="py-20 bg-gray-50">
        <div class="max-w-7xl mx-
auto px-4 sm:px-6 lg:px-8">
            <div class="grid
grid-cols-1 lg:grid-cols-2 gap-12
items-center">
                <div>
                    <h2
class="text-3xl sm:text-4xl font-
bold text-gray-900 mb-6">Mengapa
Memilih Pakeng-Ternak?</h2>
                    <p
class="text-base sm:text-lg text-
gray-600 mb-8">
                        Kami
memahami tantangan yang dihadapi
peternak modern. Pakeng-Ternak
hadir sebagai solusi
                        teknologi
yang mengintegrasikan semua aspek
manajemen peternakan dalam satu
platform yang mudah
                        digunakan
.
                    </p>
                    <div
class="space-y-4">
                        <div
class="flex items-start">
                            <div
                                c
lass="bg-primary w-6 h-6 rounded-
full flex items-center justify-
center mr-4 mt-1 flex-shrink-0">
```

```
                                    <
svg class="w-4 h-4 text-white"
fill="none" stroke="currentColor"

  viewBox="0 0 24 24">

  <path stroke-linecap="round"
stroke-linejoin="round" stroke-
width="2"

      d="M5 13l4 4L19 7"></path>
                                    <
/svg>
                                </div
>
                                <div>
                                    <
h3 class="text-lg font-semibold
text-gray-900">Multi-Role
Dashboard</h3>
                                    <
p class="text-gray-600">Dashboard
khusus untuk peternak, admin, dan
penyuluh dengan

  fitur sesuai kebutuhan</p>
                                </div
>
                            </div>
                            <div
class="flex items-start">
                                <div
                                    c
lass="bg-primary w-6 h-6 rounded-
full flex items-center justify-
center mr-4 mt-1 flex-shrink-0">
                                    <
svg class="w-4 h-4 text-white"
fill="none" stroke="currentColor"

  viewBox="0 0 24 24">

  <path stroke-linecap="round"
stroke-linejoin="round" stroke-
width="2"

      d="M5 13l4 4L19 7"></path>
                                    <
/svg>
                                </div
>
                                <div>
                                    <
h3 class="text-lg font-semibold
text-gray-900">Monitoring
Kesehatan Real-time</h3>
                                    <
p class="text-gray-600">Pantau
kondisi kesehatan ternak dan
dapatkan rekomendasi

  perawatan tepat waktu</p>
                                </div
>
                            </div>
                            <div
class="flex items-start">
                                <div
                                    c
lass="bg-primary w-6 h-6 rounded-
full flex items-center justify-
center mr-4 mt-1 flex-shrink-0">
                                    <
svg class="w-4 h-4 text-white"
fill="none" stroke="currentColor"

  viewBox="0 0 24 24">

  <path stroke-linecap="round"
stroke-linejoin="round" stroke-
width="2"

      d="M5 13l4 4L19 7"></path>
                                    <
/svg>
                                </div
>
                                <div>
                                    <
h3 class="text-lg font-semibold
text-gray-900">Laporan
Komprehensif</h3>
                                    <
p class="text-gray-600">Generate
laporan detail untuk analisis
bisnis dan pengambilan

  keputusan</p>
                                </div
>
                            </div>
                        </div>
                    </div>
                    <div class="bg-
white rounded-2xl p-6 sm:p-8
shadow-xl">
                        <div
class="text-center">
                            <div
                                class
="bg-primary/10 w-20 h-20
rounded-full flex items-center
justify-center mx-auto mb-6">
                                <svg
class="w-10 h-10 text-primary"
fill="none" stroke="currentColor"
                                    v
iewBox="0 0 24 24">
                                    <
path stroke-linecap="round"
stroke-linejoin="round" stroke-
width="2"

  d="M13 10V3L4 14h7v7l9-11h-
7z"></path>
```

```
                            </svg
>
                        </div>
                        <h3
class="text-2xl font-bold text-
gray-900 mb-4">Siap Memulai?</h3>
                        <p
class="text-gray-600 mb-
6">Bergabunglah dengan ribuan
peternak yang telah merasakan
                            kemud
ahan Pakeng-Ternak</p>
                        <button
                            class
="bg-primary text-white px-8 py-3
rounded-lg font-semibold
hover:bg-primary/90 transition-
colors w-full">
                            Dafta
r Gratis Sekarang
                        </button>
                    </div>
                </div>
            </div>
        </div>
    </section>
```

Aunth
login.blade.php

```
    <div class="w-full max-w-6xl
mx-auto grid grid-cols-1 lg:grid-
cols-2 gap-8 items-center">
        <!-- Left Side - Branding
& Info -->
        <div class="text-white
text-center lg:text-left slide-
in">
            <!-- Logo -->
            <div class="flex
items-center justify-center
lg:justify-start mb-8">
                <div class="bg-
white/20 backdrop-filter
backdrop-blur-lg rounded-xl p-3">
                    <svg
class="h-12 w-12 text-white"
fill="none" stroke="currentColor"
viewBox="0 0 24 24">
                        <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
                            d="M1
9 21V5a2 2 0 00-2-2H7a2 2 0 00-2
2v16m14 0h2m-2 0h-5m-9 0H3m2
0h5M9 7h1m-1 4h1m4-4h1m-1 4h1m-5
10v-5a1 1 0 011-1h2a1 1 0 011
1v5m-4 0h4">
                        </path>
                    </svg>
                </div>
                <span class="ml-4
text-3xl font-bold">Pakeng-
Ternak</span>
            </div>

            <h1 class="text-4xl
lg:text-5xl font-bold mb-6
leading-tight">
                Selamat Datang
Kembali!
            </h1>
            <p class="text-xl
text-green-100 mb-8 max-w-md mx-
auto lg:mx-0">
                Masuk ke
dashboard Anda dan lanjutkan
mengelola peternakan dengan
teknologi terdepan.
            </p>

            <!-- Features
highlight -->
        </div>

        <!-- Right Side - Login
Form -->
        <div class="w-full max-w-
md mx-auto">
            <div class="glass-
effect rounded-2xl shadow-2xl p-8
slide-in" style="animation-delay:
0.2s;">
                <div class="text-
center mb-8">
                    <h2
class="text-3xl font-bold text-
gray-900 mb-
2">Masuk</h2>
                </div>



                <!-- Error
Messages (Laravel Style) -->
                <div id="error-
message"
                    class="hidden
mb-4 bg-red-50 border border-red-
200 text-red-800 px-4 py-3
rounded-lg">
                    <div
class="flex items-center">
                        <svg
class="w-5 h-5 text-red-500 mr-3"
fill="none" stroke="currentColor"
                            viewB
ox="0 0 24 24">
                            <path
stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
                                d
="M12 8v4m0 4h.01M21 12a9 9 0 11-
18 0 9 9 0 0118 0z"></path>
                        </svg>
```

```
                        <span
id="error-text">Email atau
password salah</span>
                    </div>
                </div>

                <!-- Login Form -
->
                <form
id="loginForm" method="POST"
action="{{
route('admin.login.submit') }}">
                    @csrf
                    <input
type="hidden" id="selected-role"
name="role" value="personal">

                    <div
class="mb-4">
                        <label
for="email" class="block text-sm
font-medium text-gray-700 mb-
2">Email</label>
                        <div
class="relative">
                            <div
class="absolute inset-y-0 left-0
pl-3 flex items-center pointer-
events-none">
                                <
svg class="h-5 w-5 text-gray-400"
fill="none" stroke="currentColor"

  viewBox="0 0 24 24">

  <path stroke-linecap="round"
stroke-linejoin="round" stroke-
width="2"

      d="M16 12a4 4 0 10-8 0 4 4
0 008 0zm0 0v1.5a2.5 2.5 0 005
0V12a9 9 0 10-9 9m4.5-1.206a8.959
8.959 0 01-4.5 1.207">

  </path>
                                <
/svg>
                            </div
>
                            <inpu
t type="email" id="email"
name="email" required
                                c
lass="block w-full pl-10 pr-3 py-
3 border border-gray-300 rounded-
lg focus:outline-none focus:ring-
2 focus:ring-primary
focus:border-transparent
transition-colors"
                                p
laceholder="nama@email.com">
                        </div>
                    </div>

                    <div
class="mb-6">
                        <label
for="password" class="block text-
sm font-medium text-gray-700 mb-
2">Password</label>
                        <div
class="relative">
                            <div
class="absolute inset-y-0 left-0
pl-3 flex items-center pointer-
events-none">
                                <
svg class="h-5 w-5 text-gray-400"
fill="none" stroke="currentColor"

  viewBox="0 0 24 24">

  <path stroke-linecap="round"
stroke-linejoin="round" stroke-
width="2"

      d="M12 15v2m-6 4h12a2 2 0
002-2v-6a2 2 0 00-2-2H6a2 2 0 00-
2 2v6a2 2 0 002 2zm10-10V7a4 4 0
00-8 0v4h8z">

  </path>
                                <
/svg>
                            </div
>
                            <inpu
t type="password" id="password"
name="password" required
                                c
lass="block w-full pl-10 pr-10
py-3 border border-gray-300
rounded-lg focus:outline-none
focus:ring-2 focus:ring-primary
focus:border-transparent
transition-colors"
                                p
laceholder="••••••••">
                            <butt
on type="button" class="absolute
inset-y-0 right-0 pr-3 flex
items-center"
                                o
nclick="togglePassword()">
                                <
svg id="eye-icon" class="h-5 w-5
text-gray-400 hover:text-gray-
600" fill="none"

  stroke="currentColor"
viewBox="0 0 24 24">

  <path stroke-linecap="round"
stroke-linejoin="round" stroke-
width="2"
```

```
      d="M15 12a3 3 0 11-6 0 3 3 0 016 0z"></path>

  <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2"

      d="M2.458 12C3.732 7.943 7.523 5 12 5c4.478 0 8.268 2.943 9.542 7-1.274 4.057-5.064 7-9.542 7-4.477 0-8.268-2.943-9.542-7z">

  </path>
                                </svg>
                            </button>
                        </div>
                    </div>

                    <div class="flex items-center justify-between mb-6">
                    </div>

                    <button type="submit"
                        class="w-full bg-primary text-white py-3 px-4 rounded-lg font-semibold hover:bg-secondary transition-colors focus:outline-none focus:ring-2 focus:ring-primary focus:ring-offset-2 transform hover:scale-105 transition-transform">
                        <span id="login-text">Masuk</span>
                        <svg id="loading-spinner" class="hidden animate-spin -mr-1 ml-3 h-5 w-5 text-white inline" fill="none" viewBox="0 0 24 24">
                            <circle class="opacity-25" cx="12" cy="12" r="10" stroke="currentColor" stroke-width="4"></circle>
                            <path class="opacity-75" fill="currentColor" d="M4 12a8 8 0 018-8V0C5.373 0 0 5.373 0 12h4zm2 5.291A7.962 7.962 0 014 12H0c0 3.042 1.135 5.824 3 7.938l3-2.647z">
                            </path>
                        </svg>
                    </button>
                </form>

                <!-- Register Link -->
                <div class="mt-6 text-center">
                    <p class="text-gray-600">
                        Belum punya akun?
                        <a href="{{ route('admin.register') }}" class="text-primary hover:text-secondary font-semibold transition-colors">
                            Daftar di sini
                        </a>
                    </p>
                </div>

                <!-- Back to Home -->
                <div class="mt-4 text-center">
                    <a href="/" class="text-gray-500 hover:text-gray-700 text-sm transition-colors inline-flex items-center">
                        <svg class="w-4 h-4 mr-1" fill="none" stroke="currentColor" viewBox="0 0 24 24">
                            <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M10 19l-7-7m0 0l7-7m-7 7h18"></path>
                        </svg>
                        Kembali ke beranda
                    </a>
                </div>
            </div>
        </div>
    </div>
```

Routes
web.php

```
<?php

use App\Http\Controllers\AdminController;
use App\Http\Controllers\AdminSesiController;
```

```
use
App\Http\Controllers\InformasiCon
troller;
use
App\Http\Controllers\InformasiPak
anController;
use
App\Http\Controllers\KonsultasiCo
ntroller;
use
App\Http\Controllers\LaporanContr
oller;
use
App\Http\Controllers\PakanControl
ler;
use
App\Http\Controllers\PenyuluhCont
roller;
use
App\Http\Controllers\PerawatanCon
troller;
use
App\Http\Controllers\PeternakCont
roller;
use
App\Http\Controllers\TernakContro
ller;
use
Illuminate\Support\Facades\Route;

Route::get('/', function () {
    return view('home');
});

Route::prefix('admin')-
>middleware('admin:Admin')-
>group(function () {
    Route::get('/dashboard',
[AdminController::class,
'index'])
        -
>name('admin.dashboard');
    Route::get('/kelola-user',
[AdminController::class,
'userView'])
        ->name('admin.userView');
    Route::post('/store-user',
[AdminController::class,
'storeUser'])
        ->name('user.store');
    Route::delete('/users/{idUser
}', [AdminController::class,
'deleteUser'])
        ->name('user.delete');
    Route::put('/users/{idUser}/u
pdate', [AdminController::class,
'updateUser'])
        ->name('user.update');
    Route::get('/ternak',
[AdminController::class,
'ternak'])
        ->name('admin.ternak');
    Route::get('/informasi',
[AdminController::class,
'informasi'])
        -
>name('admin.informasi');
    Route::post('/store-
informasi',
[InformasiController::class,
'store'])
        -
>name('informasi.store');
    Route::get('/pakan',
[AdminController::class,
'pakan'])
        ->name('admin.pakan');
    Route::post('/store-pakan',
[InformasiPakanController::class,
'store'])
        ->name('pakan.store');
    Route::delete('pakan/{idPakan
}',
[InformasiPakanController::class,
'destroy'])-
>name('pakan.destroy');
    Route::put('pakan/{idPakan}/u
pdate',
[InformasiPakanController::class,
'update'])->name('pakan.update');
    Route::delete('informasi/{idI
nformasi}',
[InformasiController::class,
'destroy'])-
>name('informasi.destroy');
    Route::put('informasi/{idInfo
rmasi}/update',
[InformasiController::class,
'update'])-
>name('informasi.update');
});

Route::prefix('admin')-
>middleware('authenticated')-
>group(function () {
    Route::get('login',
[AdminSesiController::class,
'adminLoginView'])-
>name('admin.login');
    Route::post('login',
[AdminSesiController::class,
'login'])-
>name('admin.login.submit');
    Route::get('register',
[AdminSesiController::class,
'registerView'])-
>name('admin.register');
    Route::post('register',
[AdminSesiController::class,
'register'])-
>name('admin.register.submit');
});

// Perbaikan Routes untuk
Peternak
```

```
Route::prefix('peternak')-
>middleware('admin:Peternak')-
>group(function () {

    // Dashboard Route
    Route::get('/dashboard',
[PeternakController::class,
'index'])
        -
>name('peternak.dashboard');

    // Ternak Routes - RESTful
Resource Routes
    Route::get('/ternak',
[TernakController::class,
'index'])
        -
>name('peternak.ternak');

    Route::get('/ternak/create',
[TernakController::class,
'create'])
        ->name('ternak.create');

    Route::post('/ternak',
[TernakController::class,
'store'])
        ->name('ternak.store');

    Route::get('/ternak/{id}',
[TernakController::class,
'show'])
        ->name('ternak.show');

    Route::get('/ternak/{id}/edit
', [TernakController::class,
'edit'])
        ->name('ternak.edit');

    Route::put('/ternak/{id}',
[TernakController::class,
'update'])
        ->name('ternak.update');

    Route::delete('/ternak/{id}',
[TernakController::class,
'destroy'])
        ->name('ternak.destroy');

    // Additional Ternak Routes
    Route::get('/ternak/{id}/data
', [TernakController::class,
'getTernakData'])
        ->name('ternak.getData');

    Route::get('/ternak/export',
[TernakController::class,
'export'])
        ->name('ternak.export');

    // Konsultasi Routes -
RESTful Resource Routes
    Route::get('/konsultasi',
[KonsultasiController::class,
'index'])
        -
>name('peternak.konsultasi');;

    Route::post('/konsultasi',
[KonsultasiController::class,
'store'])
        -
>name('konsultasi.store');

    Route::get('/konsultasi/{id}'
, [KonsultasiController::class,
'show'])
        -
>name('konsultasi.show');

    Route::delete('/konsultasi/{i
d}',
[KonsultasiController::class,
'destroy'])
        -
>name('konsultasi.destroy');

    // Kesehatan Route
    Route::get('/kesehatan',
[PeternakController::class,
'kesehatan'])
        -
>name('peternak.kesehatan');

    // Laporan Route
    Route::get('/laporan',
[PeternakController::class,
'laporan'])
        -
>name('peternak.laporan');

    // Profile Route
    Route::get('/profile',
[PeternakController::class,
'profile'])
        -
>name('peternak.profile');

    Route::put('/profile',
[PeternakController::class,
'updateProfile'])
        -
>name('peternak.profile.update');
});

// Alternative: Menggunakan
Resource Route (lebih clean)
Route::prefix('peternak')-
>middleware('admin:Peternak')-
>group(function () {

    // Dashboard Route
    Route::get('/dashboard',
[PeternakController::class,
'index'])
```

```
        -
>name('peternak.dashboard');

    // Pakan Route
    Route::get('/pakan',
[PeternakController::class,
'pakan'])
        ->name('peternak.pakan');

    // Ternak Resource Routes
    Route::resource('ternak',
TernakController::class)-
>except(['create', 'edit']);

    // Override route names untuk
konsistensi dengan blade template
    Route::get('/ternak',
[TernakController::class,
'index'])
        -
>name('peternak.ternak');

    // Additional Ternak Routes
    Route::get('/ternak/{id}/data
', [TernakController::class,
'getTernakData'])
        ->name('ternak.getData');

    Route::post('/ternak/export',
[TernakController::class,
'export'])
        ->name('ternak.export');

    // Konsultasi Resource Routes
    Route::resource('konsultasi',
KonsultasiController::class)-
>only(['index', 'store', 'show',
'destroy']);
    // Override route names untuk
konsistensi
    Route::get('/konsultasi',
[KonsultasiController::class,
'index'])
        -
>name('peternak.konsultasi');

    // Other Routes
    Route::get('/kesehatan',
[PeternakController::class,
'kesehatan'])
        -
>name('peternak.kesehatan');

    Route::get('/laporan',
[PeternakController::class,
'laporan'])
        -
>name('peternak.laporan');

    Route::get('/profile',
[PeternakController::class,
'profile'])
        -
>name('peternak.profile');

    Route::put('/profile',
[PeternakController::class,
'updateProfile'])
        -
>name('peternak.profile.update');
});

Route::prefix('penyuluh')-
>middleware('admin:Penyuluh')-
>group(function () {
    // Dashboard Penyuluh
    Route::get('/dashboard',
[PenyuluhController::class,
'index'])
        -
>name('penyuluh.dashboard');

    // Daftar ternak route
    Route::get('/ternak',
[PenyuluhController::class,
'ternak'])
        -
>name('penyuluh.ternak');

    // Profil Route
    Route::get('/profile',
[PenyuluhController::class,
'profile'])
        -
>name('penyuluh.profile');

    // Ternak Route
    Route::get('/ternak',
[PenyuluhController::class,
'ternak'])
        -
>name('penyuluh.ternak');

    Route::get('/laporan',
[LaporanController::class,
'index'])
        -
>name('penyuluh.laporan');
    Route::post('/laporan/store',
[LaporanController::class,
'store'])
        -
>name('penyuluh.laporan.store');
    Route::get('/laporan/{id}/det
ail', [LaporanController::class,
'getDetail'])
        ->name('show.laporan');
    Route::get('laporan/{id}/edit
', [LaporanController::class,
'edit'])->name('edit');
    Route::get('laporan/{id}/prin
t', [LaporanController::class,
'printReport'])->name('print');
```

```
    Route::get('laporan/{id}/pdf'
, [LaporanController::class,
'exportPdf'])->name('pdf');
    Route::delete('laporan/{id}',
[LaporanController::class,
'destroy'])->name('destroy');
    Route::post('laporan/bulk-
delete',
[LaporanController::class,
'bulkDestroy'])->name('bulk-
destroy');


    // Rekomendasi Pakan Penyuluh
    Route::get('/pakan',
[PakanController::class,
'index'])
        ->name('penyuluh.pakan');
    Route::post('/pakan/add-
pakan', [PakanController::class,
'store'])->name('pakan.store');

    Route::get('/pakan/detail/{id
}', [PakanController::class,
'pakanDetailView'])
        -
>name('pakan.detailView');
    Route::get('/pakan/update/{id
}', [PakanController::class,
'pakanEditView'])
        ->name('pakan.editView');
    Route::put('/pakan/edit/{id}'
, [PakanController::class,
'edit'])
        ->name('pakan.edit');
    Route::delete('/pakan/delete/
{id}', [PakanController::class,
'destroy'])
        ->name('pakan.destroy');

    // Perawatan Penyuluh
    Route::get('/perawatan',
[PerawatanController::class,
'index'])
        -
>name('penyuluh.perawatan');
    Route::get('/perawatan/tambah
', [PerawatanController::class,
'perawatanStoreView'])
        -
>name('perawatan.addView');
    Route::post('/perawatan/add-
perawatan',
[PerawatanController::class,
'store'])-
>name('perawatan.store');

    Route::get('/perawatan/detail
/{id}',
[PerawatanController::class,
'perawatanDetailView'])
        -
>name('perawatan.detailView');

    Route::get('/perawatan/update
/{id}',
[PerawatanController::class,
'perawatanEditView'])
        -
>name('perawatan.editView');

    Route::put('/perawatan/edit{i
d}', [PerawatanController::class,
'edit'])
        ->name('perawatan.edit');

    Route::delete('/perawatan/del
ete/{id}',
[PerawatanController::class,
'destroy'])
        -
>name('perawatan.destroy');

    // Daftar Konsultasi
    Route::get('/konsultasi',
[KonsultasiController::class,
'penyuluhView'])
        -
>name('penyuluh.konsultasi');
    Route::get('/konsultasi/updat
e/{id}',
[KonsultasiController::class,
'konsultasiEditView'])
        -
>name('konsultasi.editView');
    Route::get('/konsultasi/{id}/
detail',
[KonsultasiController::class,
'getDetail'])
        -
>name('penyuluh.ambilDetail');
    Route::put('/konsultasi/updat
e-respon/{id}',
[KonsultasiController::class,
'storeResponPenyuluh'])
        -
>name('konsultasi.updateRespon');
    Route::put('/konsultasi/updat
e-status/{id}',
[KonsultasiController::class,
'updateStatus'])
        -
>name('konsultasi.updateStatus');
});

Route::post('/logout',
[AdminSesiController::class,
'logout'])->name('logout');

if (app()->environment('local'))
{
    Route::get('/test-
errors/{code}', function ($code)
{
        switch ($code) {
            case '401':
```

```
                abort(401);
            case '403':
                abort(403);
            case '404':
                abort(404);
            case '500':
                abort(500);
            default:
                abort(404);
        }
    });
}

Route::get('/unauthorized',
function () {
    abort(403);
})->name('unauthorized');

Route::get('/informasi',
[InformasiController::class,
'index'])-
>name('informasi.index');
Route::get('/informasi/{idInforma
si}',
[InformasiController::class,
'show'])->name('informasi.show');
Route::get('/pakan',
[InformasiPakanController::class,
'index'])->name('pakan.index');
Route::get('/pakan/{idPakan}',
[InformasiPakanController::class,
'show'])-
>name('public.pakan.show');
Route::get('/informasi/{idInforma
si}',
[InformasiController::class,
'show'])-
>name('public.informasi.show')
```

PEMERINTAH PROVINSI SULAWESI SELATAN

DINAS PENANAMAN MODAL DAN PELAYANAN TERPADU SATU PINTU

Jl.Bougenville No.5 Telp. (0411) 441077 Fax. (0411) 448936
Website : http://simap-new.sulselprov.go.id Email : ptsp@sulselprov.go.id
Makassar 90231

Nomor	:	**14619/S.01/PTSP/2025**	**Kepada Yth.**
Lampiran	:	-	Bupati Pinrang
Perihal	:	**Izin penelitian**	c.q Kepala DPMPTSP Kab. Pinrang
			di-
			Tempat

Berdasarkan surat Ketua LPPM Univ. Muhamamdiyah Parepare Nomor : 255/LPPM/II.3.AU/IP/2025 tanggal 19 Juni 2025 perihal tersebut diatas, mahasiswa/peneliti dibawah ini:

N a m a	:	**WAHYUNITA. T**
Nomor Pokok	:	7315074302030001
Program Studi	:	Teknik Informatika
Pekerjaan/Lembaga	:	Mahasiswa (S1)
Alamat	:	Jl. Jend. Ahmad Yani Km. 6 Kota Parepare

dpmptsp
PROVINSI SULAWESI SELATAN

Bermaksud untuk melakukan penelitian di daerah/kantor saudara dalam rangka menyusun SKRIPSI, dengan judul :

" Sistem Informasi Peningkatan Kualitas Ternak Sapi di Desa Pakeng Kecamatan Lembang "

Yang akan dilaksanakan dari : Tgl. ***02 Juli s/d 02 Agustus 2025***

Sehubungan dengan hal tersebut diatas, pada prinsipnya kami ***menyetujui*** kegiatan dimaksud dengan ketentuan yang tertera di belakang surat izin penelitian.

Demikian Surat Keterangan ini diberikan agar dipergunakan sebagaimana mestinya.

Diterbitkan di Makassar
Pada Tanggal 02 Juli 2025

KEPALA DINAS PENANAMAN MODAL DAN PELAYANAN TERPADU SATU PINTU PROVINSI SULAWESI SELATAN

ASRUL SANI, S.H., M.Si.
Pangkat : PEMBINA UTAMA MUDA (IV/c)
Nip : 19750321 200312 1 008

Tembusan Yth
1. Ketua LPPM Univ. Muhamamdiyah Parepare;
2. *Pertinggal.*

Nomor: 14619/S.01/PTSP/2025

KETENTUAN PEMEGANG IZIN PENELITIAN :

1. Sebelum dan sesudah melaksanakan kegiatan, kepada yang bersangkutan melapor kepada Bupati/Walikota C q. Kepala Bappelitbangda Prov. Sulsel, apabila kegiatan dilaksanakan di Kab/Kota
2. Penelitian tidak menyimpang dari izin yang diberikan
3. Mentaati semua peraturan perundang-undangan yang berlaku dan mengindahkan adat istiadat setempat
4. Menyerahkan 1 (satu) eksamplar hardcopy dan softcopy kepada Gubernur Sulsel. Cq. Kepala Badan Perencanaan Pembangunan Penelitian dan Pengembangan Daerah Prov. Sulsel
5. Surat izin akan dicabut kembali dan dinyatakan tidak berlaku apabila ternyata pemegang surat izin ini tidak mentaati ketentuan tersebut diatas.

REGISTRASI ONLINE IZIN PENELITIAN DI WEBSITE :
https://izin-penelitian.sulselprov.go.id

OMOR REGISTRASI 20250702933824

Catatan :
• UU ITE No. 11 Tahun 2008 Pasal 5 ayat 1 "*Informasi Elektronik dan/atau hasil cetaknya merupakan alat bukti yang sah.*"
• Dokumen ini telah ditandatangani secara elektronik menggunakan **sertifikat elektronik** yang diterbitkan **BSrE**
• Surat ini dapat dibuktikan keasliannya dengan melakukan *scan* pada *QR Code*

ANGKET PENILAIAN SISTEM INFORMASI TERNAK SAPI

DESA PAKENG, KECAMATAN LEMBANG

Petunjuk:

Beri tanda silang (✓) pada kolom angka yang sesuai dengan pendapat Anda.

Skala Nilai:

1= Sangat Tidak Setuju

2 = Tidak Setuju

3 = Netral

4 = Setuju

5= Sangat Setuju

Identitas Responden *(opsional)*

- **Nama** : .....
- **Usia** : .....
- **Jenis Kelamin** ☐ Laki-laki ☐ Perempuan
- **Status** ☐ Peternak ☐ Lainnya: .....
- **Apakah Anda pernah menggunakan sistem ini?**
  ☐ Ya ☐ Belum

Bagian I – Penilaian Penggunaan Sistem

No	Pernyataan	1	2	3	4	5
1	Sistem mudah digunakan oleh saya sebagai peternak					
2	Tampilan sistem jelas dan tidak membingungkan.					
3	Saya memahami cara menggunakan sistem tanpa bantuan orang lain.					

4	Sistem membantu saya memahami cara pemberian pakan yang benar.					
5	Sistem memberi informasi kesehatan ternak secara mudah dipahami.					
6	Saya bisa berkonsultasi dengan penyuluh melalui sistem dengan mudah.					
7	Saya mendapatkan balasan konsultasi dengan cepat dan jelas.					
8	Informasi dari sistem bermanfaat untuk ternak saya.					
9	Sistem ini membantu saya mengambil keputusan dalam merawat ternak.					
10	Saya merasa sistem ini lebih baik daripada hanya bertanya secara langsung.					
11	Sistem ini membantu saya menghemat waktu.					
12	Fitur-fitur sistem sesuai dengan kebutuhan saya sebagai peternak.					
13	Sistem ini membuat saya lebih percaya diri dalam merawat ternak.					
14	Sistem sangat bermanfaat bagi peternak di desa.					
15	Saya puas menggunakan sistem ini secara keseluruhan.					

Bagian II – Pertanyaan Terbuka

1. Apa kelebihan sistem ini menurut Anda?
   .....
   .....
   .....
   .....
2. Apa kekurangan atau kendala yang Anda alami saat menggunakan sistem ini?
   .....

.....
.....
.....

3. Fitur apa yang paling Anda sukai?
   .....
   .....

4. Saran Anda agar sistem ini bisa lebih baik:
   .....
   .....
   .....
   .....

DATA TERNAK SAPI
DESA PAKENG KECAMATAN LEMBANG
KABUPATEN PINRANG

No	Nama	Alamat	Dusun	Kepala Rumah Tangga	No. Urut Bangunan	Jumlah Ekor Ternak
1	Tahir/Irma	Lajoro	Pakeng	Tahir	2	5
2	Wandi/Bunga Sakka	Lajoro	Pakeng	Wandi	5	6
3	Nusu/Hania	Lajoro	Pakeng	Nusu	7	4
4	Luter/Mia	Lajoro	Pakeng	Luter	9	7
5	Basri/Nurafni	Lajoro	Pakeng	Basi	10	6
6	Uji/Sunusi	Lajoro	Pakeng	Uji	11	4
7	Haris Lasen/Saida	Lajoro	Pakeng	Haris Lasen	12	5
8	Acong/Hamsiah	Lajoro	Pakeng	Acong	13	3
9	Tahang/Ani	Lajoro	Pakeng	Tahang	18	5
10	Yateng/Juhani	Lajoro	Pakeng	Yateng	19	4
11	Iskandar Tajangi/Sujirah	Lajoro	Pakeng	Iskandar Tajangi	20	8
12	Jamal Bin Ibrahim/Risa	Lajoro	Pakeng	Jamal Bin Ibrahim	22	6
13	Dami Bin Lahaba/Mina	Lajoro	Pakeng	Dami Bin Lahaba	23	7
14	Lacci/Hanisa	Lajoro	Pakeng	Lacci	24	3
15	Padda/Jami	Lajoro	Pakeng	Padda	25	5
16	Ramalli/Darna	Lajoro	Pakeng	Ramalli	29	7
17	Kangi/	Lajoro	Pakeng	Kangi	30	4
18	Pasi/Nasa	Lajoro	Pakeng	Pasi	35	3
19	Abd. Kadir/Heni	Lajoro	Pakeng	Abd. Kadir	36	6
20	Laupa/Ruma	Lajoro	Pakeng	Laupa	37	8
21	Tadi/Munira	Lajoro	Pakeng	Tadi	40	5
22	Abd. Rahim M/Darmi	Lajoro	Pakeng	Abd Rahim	51	6
23	Tuo/Wahi	Lajoro	Pakeng	Tuo	56	4
24	Saleo/Ponno	Lajoro	Pakeng	Saleo	59	3
25	Sapar/Usni	Lajoro	Pakeng	Sapar	83	5
26	Sawina/Onding	Lajoro	Pakeng	Sawina	84	6
27	Mappe/Panti	Lajoro	Pakeng	Mappe	89	4
28	Baco/Sani	Lajoro	Pakeng	Baco	73	6
29	Yunus/Ecce	Lajoro	Pakeng	Yunus	75	3
30	Hamsa/Mara	Lajoro	Pakeng	Hamsa	77	8
31	Samsudding/Suma	Lajoro	Pakeng	Samsudding	79	5
32	Salama/Bunga	Lajoro	Pakeng	Salama	67	3
33	Kadang/Isa	Lajoro	Pakeng	Kadang	72	7
34	Nurdin/Herlina	Lajoro	Pakeng	Nurdin	98	3
35	Haris/Sarah	Lajoro	Pakeng	Haris	100	7
36	Nurdin/Maraisa	Lajoro	Pakeng	Nurdin	101	4
37	Ali/Indo Tuo	Lajoro	Pakeng	Ali	103	6
38	Sakka/Hjh. Sunari	Lajoro	Pakeng	Sakka	104	5
39	Maseng/Darawia	Lajoro	Pakeng	Maseng	105	3
40	Ronta/Mari	Lajoro	Pakeng	Ronta	93	7
41	Tahang/Hali	Lajoro	Pakeng	Tahang	96	4
42	Muhida/Pisah	Lajoro	Pakeng	Muhida	110	5
43	Jumarang/Nasra	Lajoro	Pakeng	Jumarang	111	3
44	Sodding/Sara	Lajoro	Pakeng	Sodding	112	6
45	Lahi/Suri	Lajoro	Pakeng	Lahi	113	4
46	Jahida/Hasmawati	Lajoro	Pakeng	Jahida	116	7
47	Amir/Desi Junianti	Lajoro	Pakeng	Amir	120	8
48	Nirwana/Mirna Afsari	Lajoro	Pakeng	Nirwana	121	5
49	Arifuddin/Tuo	Lajoro	Pakeng	Arifuddin	123	3
50	Sauda/Daha	Lajoro	Pakeng	Sauda	129	5
51	Kaba/Bunga	Lajoro	Pakeng	Kaba	130	7
52	Sisi/Bua	Lajoro	Pakeng	Sisi	133	8
53	Nurdin/Diana	Lajoro	Pakeng	Nurdin	136	7
54	Parasi/Conong	Lajoro	Pakeng	Parasi	141	3
55	Muh Basri/Ita Puspita Sari	Lajoro	Pakeng	Muh Basri	144	6
56	Rippa/Salama	Lajoro	Pakeng	Rippa	145	3
57	Muhammad Nasir/Parasima	Lajoro	Pakeng	Muhammad Nasir	149	4
58	Asby Usman/Diana	Lajoro	Pakeng	Asby Usman	150	5
59	Kappia/Becce	Buttu Angin	Pakeng	Kappia	4	6

60	Rasa/Yallu	Buttu Angin	Pakeng	Rasa	8	5
61	Budu/Tappe	Buttu Angin	Pakeng	Budu	17	3
62	Sattu/Becce	Buttu Angin	Pakeng	Sattu	18	5
63	Kata/Numa	Buttu Angin	Pakeng	Kata	19	5
64	Tuo/Juneda	Buttu Angin	Pakeng	Tuo	21	4
65	Canding/Yappe	Buttu Angin	Pakeng	Canding	22	6
66	Asang/Dari	Buttu Angin	Pakeng	Asang	12	4
67	Rais/Kendeng	Buttu Angin	Pakeng	Rais	6	7
68	Linta/Bua	Buttu Angin	Pakeng	Linta	7	5
69	Lahadi/Sule	Sepang	Pakeng	Lahadi	2	4
70	Rudi/Masnia	Sepang	Pakeng	Rudi	6	7
71	Jupri/Ani	Sepang	Pakeng	Jupri	12	6
72	Bahtiar/Habiba	Sepang	Pakeng	Bahtiar	13	4
73	Alimuddin S/Maryam	Sepang	Pakeng	Alimuddin S	17	7
74	Saleo/Tina	Sepang	Pakeng	Saleo	18	6
75	Sukir/Tura	Sepang	Pakeng	Sukir	23	7
76	Sumaila/Herlina	Sepang	Pakeng	Sumaila	24	4
77	Lakka/Bunga	Sepang	Pakeng	Lakka	26	6
78	Rabaisa/Hasmita	Sepang	Pakeng	Rabaisa	30	7
79	Taju/Sureda	Sepang	Pakeng	Taju	39	3
80	Jappi/Umi	Sepang	Pakeng	Jappi	42	7
81	Hasri/Murni	Sepang	Pakeng	Hasri	47	14
82	Syamsina/Wahyu Putra Saenata	Sepang	Pakeng	Syamsina	61	5
83	Hamzah/Rahmatan	Sepang	Pakeng	Hamzah	62	5
84	Ali/Munni	Sepang	Pakeng	Ali	67	3
85	Ahmad Pata/Baharia	Sepang	Pakeng	Ahmad Pata	80	9
86	Budiman/Muliana	Sepang	Pakeng	Budiman	82	5
87	Seri/Bungaisa	Sepang	Pakeng	Seri	83	6
88	Tappe/Maringala	Sepang	Pakeng	Tappe	85	6
89	Sattu/Sinar	Sepang	Pakeng	Sattu	88	8
90	Biddi/	Sepang	Pakeng	Biddi	90	7
91	Amir/Hanaria	Sepang	Pakeng	Amir	91	5
92	Ponong Kadang Paga / Kartini Tapi	Sepang	Pakeng	Ponong Kadang Paga	101	4
93	Lamasi/Sami	Kampung Bari	Pakeng	Lamasi	26	5
94	Suranto/Kadang	Kampung Bari	Pakeng	Suranto	11	7
95	Rajja/Sitti	Kampung Bari	Pakeng	Rajja	107	7
96	Amir/Mai	Kampung Bari	Pakeng	Amir	120	6
97	Kadi Lahaiya/Lia Sampo	Kampung Bari	Pakeng	Kadi	122	6
98	Baco/Salamiyah	Kampung Bari	Pakeng	Baco	127	4
99	Syahril/Hatimang	Kampung Bari	Pakeng	Syahril	135	7
100	Tara/Tina	Kampung Bari	Pakeng	Tara	139	2
101	Daru/Manisa Laupa	Kampung Bari	Pakeng	Daru	144	6
102	Benu/Ramli	Kampung Bari	Pakeng	Benu	147	4
103	Usman/Mini	Kampung Bari	Pakeng	Usman	149	11
104	Mansu/Juhati	Kampung Bari	Pakeng	Mansu	153	5
105	Selle/Mariani	Kampung Bari	Pakeng	Selle	121	7
106	Tappe Baco/Sale Cahaiya	Kampung Bari	Pakeng	Tappe	151	5
107	Aliyas/ Sia Pallao	Kampung Bari	Pakeng	Aliyas	23	7
108	Rahim/Dabi	Kampung Bari	Pakeng	Rahim	24	4
109	Parasi/Sumiati	Kampung Bari	Pakeng	Parasi	25	7
110	Selle/Sanni	Kampung Bari	Pakeng	Selle	28	7
111	Sunre/Sukur	Kampung Bari	Pakeng	Suku	29	6
112	Modding/Sappe	Kampung Bari	Pakeng	Modding	30	5
113	Patolai/Barra	Kampung Bari	Pakeng	Patolai	31	6
114	Hasanuddin/Maddi	Kampung Bari	Pakeng	Hasanuddin	35	7
115	Bakri M/Anni	Kampung Bari	Pakeng	Bakri M	38	3
116	Lahamma Badawi/Marah Lasakka	Kampung Bari	Pakeng	Lahamma	39	4
117	Nurdin/Jani	Kampung Bari	Pakeng	Nurdin	44	5
118	Tappe/Pepe	Kampung Bari	Pakeng	Tappe	51	3
119	Hermang/Nurheti	Kampung Bari	Pakeng	Hermang	50	7
120	Tabu/Mare	Kampung Bari	Pakeng	Tabu	52	9
121	Supu Timbang/Masia	Kampung Bari	Pakeng	Supu Timbang	63	4
122	Amang/Sai	Kampung Bari	Pakeng	Amang	68	8
123	Raba/Maddong	Kampung Bari	Pakeng	Rabaisa	69	2

124	Hakim/Bunga	Kampung Baru	Pakeng	Hakim	70	7
125	Yudi Tungka/Muni	Kampung Baru	Pakeng	Yudi Tungka	74	7
126	Baco/Bunga	Kampung Baru	Pakeng	Baco	77	11
127	Indo Sakka/Ansya	Kampung Baru	Pakeng	Indo Sakka	80	9
128	Biding/Mansuria	Kampung Baru	Pakeng	Biding	82	6
129	Sami/Devi	Kampung Baru	Pakeng	Sami	91	7
TOTAL						**717**

Pakeng, 10 Juli 2025
An. Kepala Desa Pakeng
Kasi pemerintahan

SUHERIA

PEMERINTAH KABUPATEN PINRANG
DESA PAKENG
KECAMATAN LEMBANG

KARTU MONITORING BIMBINGAN

MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

PROPOSAL

Mahasiswa : Wahyunita.T	Pembimbing I : Andi Wafiah, S.Kom., M.Kom
NIM : 221280057	Pembimbing II : Ahmad Selao, S.T.P., M.Sc
Judul Skripsi : Sistem Informasi Peningkatan Kualitas Ternak Sapi Di Desa Pakeng Kecamatan Lembang	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 Format Penulisan disesuaikan dgn Pedoman	✓	Konsultasi 1 Satukan : Desain Database DBMS Relasi	17/02/2025
Konsultasi 2 Lengkapi data hasil survey	✓	Konsultasi 2 Typografi Tulisan Judul	29/02/2025
Konsultasi 3 Jabarkan Sti'ap Diagram	✓	Konsultasi 3 OK	
Konsultasi 4 Tambahkan rancangan Interface	✓	Konsultasi 4	
Konsultasi 5 ACC Proposal	✓	Konsultasi 5	

Lanjut ke halaman sebelah...

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton A4 berwarna Hijau muda dan dicetak timbal balik

5 15.19

KARTU MONITORING BIMBINGAN

MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

SKRIPSI

a : Wahyunita.T	Pembimbing I : Andi Wafiah, S.Kom., M.Kom
: 221280057	Pembimbing II : Ahmad Selao, S.T.P., M.Sc
Skripsi : Sistem Informasi Peningkatan Kualitas Ternak Sapi di Desa Pakeng Kecamatan Lembang	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
si 1 ronkan Judul - Tujuan n Latar Belakang	4.	Konsultasi 1 Pembahasan : Coding PHP & Mysql - tentang proses yang terlihat di Screen Shot.	
si 2 Penelitian yg disusun engkapi dilampiran	4.	Konsultasi 2 Tabel relasi dengan Bagan Struktur, Ok (perbaiki!)	
si 3 ronkan Dokumen n Aplikasi	4.	Konsultasi 3 Fitur Laporan (kombinasi PHP + ~~Ajax~~ Sql)	
si 4 cc ujian hasil	4.	Konsultasi 4 Acc hasil	
si 5 kapi revisi c ujian Tutup	4.	Konsultasi 5 Perbaiki telah di tandatangani Acc tutup	

Lanjut ke halaman sebelah...