LAMPIRAN

MIGRASI:

```
<?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('users', function (Blueprint $table)
{

            $table->id();

            $table->string('name');

            $table->string('email')->unique();

            $table->string('nim')->nullable();

            $table->string('no');

            $table->string('fakultas')->nullable();

            $table->string('jurusan')->nullable();

            $table->timestamp('email_verified_at')-
>nullable();

            $table->string('password');

            $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('users');

        Schema::dropIfExists('password_reset_tokens');

        Schema::dropIfExists('sessions');

    }

};


<?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('matkuls', function (Blueprint
$table) {
```

```
            $table->id();
            $table->string('kode');        // contoh: IF101
            $table->string('nama');                // contoh: Algoritma & Pemrograman
            $table->integer('sks');                // jumlah SKS
            $table->text('deskripsi')->nullable();  // penjelasan singkat
            $table->integer('semester');           // semester ditawarkan (1-8)
            $table->string('jurusan');             // jurusan/program studi (jika perlu)
            $table->string('tipe'); // teori/praktikum
            $table->string('status'); // status aktif/tidak
            $table->timestamps();
        });
    }

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

```
<?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('kegiatans', function (Blueprint $table) {
            $table->id();
            $table->string('nama');
            $table->timestamps();
        });
    }

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

```
<?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('agenda_kuliahs', function (Blueprint $table) {
            $table->id();
            $table->string('judul');
            $table->text('deskripsi')->nullable();
            $table->unsignedBigInteger('user_id')->nullable();
            $table->date('tanggal');
            $table->unsignedBigInteger('kegiatan_id');
```

```
            $table->timestamps();


            $table->foreign('kegiatan_id')->references('id')->on('kegiatans')->onDelete('cascade');
        });
    }

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


<?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('akademiks', function (Blueprint $table) {
            $table->id();
            $table->unsignedBigInteger('agenda_id');
            $table->unsignedBigInteger('user_id')->nullable();
            $table->unsignedBigInteger('matkul_id'); // relasi ke table matkul
            $table->integer('nilai')->nullable();     // diisi saat menu KHS
            $table->timestamps();

            $table->foreign('agenda_id')->references('id')->on('agenda_kuliahs')->onDelete('cascade');
        });
    }

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


<?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('jadwals', function (Blueprint $table) {
            $table->id();
            $table->foreignId('user_id');
            $table->string('nama')->nullable();
            $table->string('nim')->nullable();
            $table->string('fakultas')->nullable();
            $table->string('jurusan')->nullable();
            $table->date('waktu')->nullable();
            $table->text('masalah')->nullable();
```

```
            $table->string('antrian')->nullable();
            $table->string('status');
            $table->timestamps();
        });
    }

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

```
<?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('messages', function (Blueprint
$table) {
            $table->id();
            $table->foreignId('user_id')->constrained();
            $table->foreignId('recipient_id')-
>constrained('users');
            $table->text('text')->nullable();
            $table->timestamps();
        });
    }

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

MODEL:

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class AgendaKuliah extends Model
{

    protected $table = 'agenda_kuliahs';

protected $fillable = [
    'judul',
    'deskripsi',
    'tanggal',
    'kegiatan_id',
    'user_id'
];

    public function kegiatan()
    {
        return $this->belongsTo(Kegiatan::class);
```

```
    }

    public function akademiks()
    {
        return $this->hasMany(Akademik::class,
'agenda_id');
    }
    public function user()
{
    return $this->belongsTo(User::class);
}
}
```

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Akademik extends Model
{
    protected $table = 'akademiks';

protected $fillable = [
    'agenda_id',
    'matkul_id',
    'nilai',
    'user_id'
];

    public function agenda()
    {
        return $this->belongsTo(AgendaKuliah::class,
'agenda_id');
    }
    public function matkul()
{
    return $this->belongsTo(Matkul::class, 'matkul_id');
}
}
```

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Jadwal extends Model
{
        protected $fillable =
['user_id','nama','nim','fakultas','jurusan',
    'masalah','waktu','antrian','status'];
}
```

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Kegiatan extends Model
{
        protected $fillable = ['nama'];

    public function agendaKuliahs()
    {
        return $this->hasMany(AgendaKuliah::class);
    }
}
```

```
<?php
```

```
namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Matkul extends Model
{
        protected $fillable = [
        'kode',
        'nama',
        'sks',
        'deskripsi',
        'semester',
        'jurusan',
        'tipe',
        'status',
    ];
}
<?php

namespace App\Models;

use
Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;

class Message extends Model
{
    use HasFactory;

    public $table = 'messages';
    protected $fillable = ['user_id', 'text', 'recipient_id'];

    public function user(): BelongsTo {
        return $this->belongsTo(User::class, 'user_id');
    }
    public function recipient(): BelongsTo {
        return $this->belongsTo(User::class,
'recipient_id');
    }


    public function getTimeAttribute(): string {
        return $this->created_at ? $this->created_at-
>format("d M Y, H:i:s") : 'N/A';
    }


}

<?php

namespace App\Models;

// use Illuminate\Contracts\Auth\MustVerifyEmail;
use
Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as
Authenticatable;
use Illuminate\Notifications\Notifiable;
use Spatie\Permission\Traits\HasRoles;

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

    /**
     * The attributes that are mass assignable.
```

```
     *
     * @var list<string>
     */
    protected $fillable = [
        'name',
        'email',
        'password',
        'no',
        'nim',
        'fakultas',
        'jurusan'
    ];

    /**
     * The attributes that should be hidden for
serialization.
     *
     * @var list<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',
        ];
    }
}
```

CONTROLLER:

```
<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use App\Models\User;
use Illuminate\Auth\Events\Registered;
use Illuminate\Http\RedirectResponse;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Hash;
use Illuminate\Validation\Rules;
use Inertia\Inertia;
use Inertia\Response;
use Spatie\Permission\Models\Role;

class RegisteredUserController extends Controller
{
    /**
     * Show the registration page.
     */
    public function create(): Response
    {
        return Inertia::render('auth/register');
    }

    /**
     * Handle an incoming registration request.
     *
     * @throws
\Illuminate\Validation\ValidationException
```

```
     */
    public function store(Request $request): RedirectResponse
    {
        $request->validate([
            'name' => 'required|string|max:255',
            'email' => 'required|string|lowercase|email|max:255|unique:'.User::class,
            'password' => ['required', 'confirmed', Rules\Password::defaults()],
            'no' => 'required',
            'nim' => 'required',
            'fakultas' => 'nullable',
            'jurusan' => 'nullable'
        ]);

        $user = User::create([
            'name' => $request->name,
            'email' => $request->email,
            'no' => $request->no,
            'password' => Hash::make($request->password),
            'nim' =>$request->nim,
            'fakultas' => $request->fakultas,
            'jurusan' => $request->jurusan
        ]);

        $Role = Role::findByName('mahasiswa');
        $user->assignRole($Role);
    event(new Registered($user));

        Auth::login($user);

        return to_route('dashboard');
    }
}
```

```
<?php

namespace App\Http\Controllers;

use App\Models\AgendaKuliah;
use App\Models\Akademik;
use App\Models\Kegiatan;
use App\Models\Matkul;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class AgendaKuliahController extends Controller
{
    /**
     * Display a listing of the resource.
     */
public function index()
{
    $user = auth()->user();

    // base query
    $query = AgendaKuliah::with(['kegiatan', 'akademiks', 'user'])->latest();

    // filter berdasarkan role
    if ($user->hasRole('mahasiswa')) {
        // hanya agenda miliknya
        $query->where('user_id', $user->id);
    }
    // kalau admin, dosen, dosenPA → tidak ada filter (lihat semua data)

    $agendas   = $query->get();
    $kegiatans = Kegiatan::all();
    $matkuls   = Matkul::all();
```

```
    return inertia('AgendaKuliah/Index', [
        'agendas'   => $agendas,
        'kegiatans' => $kegiatans,
        'matkuls'   => $matkuls,
    ]);
}


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

    $agendaData = $request->validate([
        'judul'       => 'required|string|max:255',
        'deskripsi'   => 'required|string',
        'tanggal'     => 'required|date',
        'kegiatan_id' => 'required|exists:kegiatans,id',
    ]);

    // tambahkan user_id
    $agendaData['user_id'] = $user->id;

    $agenda = AgendaKuliah::create($agendaData);

    if ($request->has('akademiks') && is_array($request->akademiks)) {
        foreach ($request->akademiks as $data) {
            $validatedAkademik = validator($data, [
                'matkul_id' => 'required|exists:matkuls,id',
                'nilai'     => 'nullable|string'
            ])->validate();
            $validatedAkademik['agenda_id'] = $agenda->id;
            $validatedAkademik['user_id']   = $user->id; // <— tambah ini

            Akademik::create($validatedAkademik);
        }
    }

    return redirect()->route('agenda-kuliah.index')->with('success', 'Agenda kuliah berhasil dibuat.');
}

    /**
     * Display the specified resource.
     */
public function show(AgendaKuliah $agendaKuliah)
{
    $agendaKuliah->load('kegiatan', 'akademiks.matkul', 'user');

    return inertia('AgendaKuliah/Show', [
        'agenda' => $agendaKuliah
    ]);
}

public function show2(AgendaKuliah $agendaKuliah)
{
    $agendaKuliah->load('kegiatan', 'akademiks.matkul', 'user');

    return inertia('AgendaKuliah/Hasil', [
        'agenda' => $agendaKuliah
    ]);
}
```

```
/**
 * Update nilai akademik (menu KHS)
 */
public function updateNilai(Request $request, Akademik $akademik)
{

    $validated = $request->validate([
        'nilai' => 'required'
    ]);

    $akademik->update([
        'nilai'   => $validated['nilai'],
    ]);

    return redirect()->back()->with('success', 'Nilai berhasil di perbarui.');
}

    /**
     * Remove the specified resource from storage.
     */
    public function destroy(AgendaKuliah $agendaKuliah)
    {
        $agendaKuliah->delete();

        return redirect()->back()->with('success', 'Agenda kuliah berhasil dihapus.');
    }
}
<?php

namespace App\Http\Controllers;

use App\Models\Message;
use App\Models\User;
use Illuminate\Http\JsonResponse;
use Illuminate\Http\Request;
use Inertia\Inertia;

class ChatController extends Controller
{

    // public function index()
    // {
    //     $user = auth()->user();

    //     return Inertia::render('ChatBox', [
    //         'user' => [
    //             'id' => $user->id,
    //             'name' => $user->name,
    //             'email' => $user->email,
    //         ],
    //         'rootUrl' => url('/'),
    //     ]);

    // }
    public function index2(): JsonResponse
    {
        // Mengembalikan semua pengguna dengan ID dan nama
        $users = User::select('id', 'name')->get();

        return response()->json($users);
    }
    public function chatUsers()
    {
        return Inertia::render('ChatUsers', [
            'users' => User::with('roles')->get()->map(function ($user) {
```

```
            return [
                'id' => $user->id,
                'name' => $user->name,
                'roles' => $user->roles->pluck('name')->toArray(), // Ambil nama role sebagai array
            ];
        }),
    ]);
}


public function chatBox($userId)
{
    $recipient = User::findOrFail($userId);
    return Inertia::render('ChatBox', [
        'recipient' => [
            'id' => $recipient->id,
            'name' => $recipient->name,
        ],
        'rootUrl' => url('/'),
    ]);
}


public function messages(Request $request): JsonResponse
{
    $validated = $request->validate([
        'recipient_id' => 'required|exists:users,id',
    ]);

    $messages = Message::with('user:id,name')
        ->where(function ($query) use ($validated) {
            $query->where('user_id', auth()->id())
                ->where('recipient_id', $validated['recipient_id']);
        })
        ->orWhere(function ($query) use ($validated) {
            $query->where('user_id', $validated['recipient_id'])
                ->where('recipient_id', auth()->id());
        })
        ->latest()
        ->limit(50)
        ->get()
        ->append('time');

    return response()->json($messages);
}


public function message(Request $request): JsonResponse
{
    $validated = $request->validate([
        'text' => 'required|string|max:255',
        'recipient_id' => 'required|exists:users,id', // Tambahkan validasi untuk recipient_id
    ]);

    $message = Message::create([
        'user_id' => auth()->id(),
        'text' => $validated['text'],
        'recipient_id' => $validated['recipient_id'], // Pastikan field ini ditambahkan
    ]);


return response()->json([
    'success' => true,
    'refresh' => true,
```

```
    'message' => "Message created and job dispatched.",
]);

    }

public function deleteChat(Request $request): JsonResponse
{
    $validated = $request->validate([
        'recipient_id' => 'required|exists:users,id',
        'keep_history' => 'required|boolean',
    ]);

    // Kalau tidak ingin menyimpan riwayat chat → hapus
    if (!$validated['keep_history']) {
        Message::where(function ($query) use ($validated) {
            $query->where('user_id', auth()->id())
                  ->where('recipient_id', $validated['recipient_id']);
        })->orWhere(function ($query) use ($validated) {
            $query->where('user_id', $validated['recipient_id'])
                  ->where('recipient_id', auth()->id());
        })->delete();
    }

    // Proses update status user seperti sebelumnya
    $user = auth()->user();
    $currentStatus = $user->status;


    return response()->json([
        'success' => true,
        'message' => $validated['keep_history']
            ? 'Chat ended, riwayat disimpan.'
            : 'Chat ended, riwayat dihapus.',
    ]);
}


    public function updatePassword(Request $request)
    {
        $request->validate([
            'user_id' => 'required|exists:users,id',
            'password' => 'required|confirmed|min:8',
        ]);

        $user = User::find($request->input('user_id'));

        if ($user) {
            $user->update([
                'password' => bcrypt($request->input('password')),
            ]);

            return redirect()->route('login')->with('success', 'Password berhasil diubah');
        }

        return redirect()->back()->withErrors(['user_id' => 'User tidak valid']);
    }


}

<?php

namespace App\Http\Controllers;
```

```
use App\Models\Jadwal;
use App\Models\User;
use Inertia\Inertia;
use Illuminate\Support\Facades\Storage;
use Illuminate\Support\Facades\Redirect;
use Illuminate\Support\Str;
use Illuminate\Http\Request;
use Illuminate\Validation\Rule;


class JadwalController extends Controller
{
    public function index()
    {
        $user = auth()->user(); // Ambil user yang sedang login

        // Jika user memiliki role "mahasiswa", tampilkan hanya jadwal miliknya
        if ($user->hasRole('mahasiswa')) {
            $Jadwal = Jadwal::where('user_id', $user->id)->get();
        } else {
            // Jika bukan mahasiswa, tampilkan semua jadwal
            $Jadwal = Jadwal::all();
        }

        return Inertia::render('Jadwal/index', [
            'Jadwal' => $Jadwal,
        ]);
    }


    public function create()
    {
        return Inertia::render('Jadwal/create');
    }
    public function store(Request $request)
    {
        // Validasi input
        $request->validate([

            'masalah' => 'required',
        ]);

        // Ambil nomor antrian terakhir untuk hari ini
        $nomorAntrianTerakhir = Jadwal::whereDate('created_at', today())->max('antrian');
        $nomorAntrianBaru = $nomorAntrianTerakhir ? $nomorAntrianTerakhir + 1 : 1;

        // Menyimpan Jadwal ke dalam database
        Jadwal::create([
            'user_id' => auth()->id(),
            'nama' => auth()->user()->name,
            'nim' => auth()->user()->nim,
            'waktu' => now(),
            'status' => 'antrian',
            'antrian' => $nomorAntrianBaru,
            'fakultas' => auth()->user()->fakultas,
            'jurusan' => auth()->user()->jurusan,
            'masalah' => $request->input('masalah'),
        ]);

        // Ambil user yang sedang login
        $user = auth()->user();
```

```
        return Redirect::route('Jadwal.index')->with('success', 'Jadwal berhasil ditambahkan dan status user diperbarui.');
    }

    public function edit(Jadwal $Jadwal)
    {
        return Inertia::render('Jadwal/update', [
            'Jadwal' => $Jadwal,
        ]);
    }
    public function update(Request $request, Jadwal $Jadwal)
    {
        $request->validate([
            'nim' => 'required',
            'fakultas' => 'required',
            'jurusan' => 'required',
            'masalah' => 'required',
        ]);

        $Jadwal->update([
            'nim' => $request->input('nim'),
            'fakultas' => $request->input('fakultas'),
            'jurusan' => $request->input('jurusan'),
            'masalah' => $request->input('masalah'),
        ]);
        return Redirect::route('Jadwal.index')->with('success', 'Jadwal berhasil di update');
    }
    public function selesai(Jadwal $Jadwal)
    {
        $Jadwal->update(['status' => 'selesai']);

        return Redirect::route('Jadwal.index')->with('success', 'Jadwal berhasil di selesaikan');
    }
    public function destroy(Jadwal $Jadwal)
    {
        $Jadwal->delete();

        return Redirect::route('Jadwal.index')->with('success', 'Jadwal berhasil di hapus');
    }
}
```

```
<?php

namespace App\Http\Controllers;

use App\Models\Kegiatan;
use Illuminate\Http\Request;
use Inertia\Inertia;

class KegiatanController extends Controller
{
    public function index()
    {
        $kegiatans = Kegiatan::latest()->get();
        return Inertia::render('Kegiatans/Index', [
            'kegiatans' => $kegiatans
        ]);
    }

    public function store(Request $request)
    {
        $request->validate([
            'nama' => 'required|string|max:255',
        ]);
```

```
        Kegiatan::create([
            'nama' => $request->nama,
        ]);

        return redirect()->route('kegiatans.index')->with('message', 'Kegiatan berhasil ditambahkan');
    }

    public function update(Request $request, Kegiatan $kegiatan)
    {
        $request->validate([
            'nama' => 'required|string|max:255',
        ]);

        $kegiatan->update([
            'nama' => $request->nama,
        ]);

        return redirect()->route('kegiatans.index')->with('message', 'Kegiatan berhasil diperbarui');
    }

    public function destroy(Kegiatan $kegiatan)
    {
        $kegiatan->delete();
        return redirect()->route('kegiatans.index')->with('message', 'Kegiatan berhasil dihapus');
    }
}
<?php

namespace App\Http\Controllers;

use App\Models\Matkul;
use Illuminate\Http\Request;
use Inertia\Inertia;
use Illuminate\Support\Facades\Session;

class MatkulController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        $matkuls = Matkul::latest()->get();
        return Inertia::render('Matkul/Index', [
            'matkuls' => $matkuls
        ]);
    }

    /**
     * Show the form for creating a new resource.
     */
    public function create()
    {
        return Inertia::render('Matkul/Create');
    }

    /**
     * Store a newly created resource in storage.
     */
public function store(Request $request)
{
    $validated = $request->validate([
        'kode'      => 'required|string',
        'nama'      => 'required|string',
        'sks'       => 'required|integer',
        'deskripsi' => 'nullable|string',
```

```
        'semester'  => 'required|integer',
        'jurusan'   => 'required|string',
        'tipe'      => 'required|string',
        'status'    => 'required|string',
    ]);

    // Cek apakah kode sudah ada
    if (Matkul::where('kode', $validated['kode'])->exists()) {
        return redirect()->back()
            ->withInput()
            ->with('error', 'Gagal menambahkan, kode mata kuliah sudah digunakan.');
    }

    Matkul::create($validated);

    return redirect()->route('matkuls.index')->with('success', 'Matkul berhasil ditambahkan.');
}
    /**
     * Display the specified resource.
     */
    public function show(Matkul $matkul)
    {
        return Inertia::render('Matkul/Show', [
            'matkul' => $matkul
        ]);
    }

    /**
     * Show the form for editing the specified resource.
     */
    public function edit(Matkul $matkul)
    {
        return Inertia::render('Matkul/Edit', [
            'matkul' => $matkul
        ]);
    }

    /**
     * Update the specified resource in storage.
     */
public function update(Request $request, Matkul $matkul)
{
    $validated = $request->validate([
        'kode'      => 'required|string',
        'nama'      => 'required|string',
        'sks'       => 'required|integer',
        'deskripsi' => 'nullable|string',
        'semester'  => 'required|integer',
        'jurusan'   => 'required|string',
        'tipe'      => 'required|string',
        'status'    => 'required|string',
    ]);

    // Cek jika kode baru bentrok dengan matkul lain
    if (Matkul::where('kode', $validated['kode'])->where('id', '!=', $matkul->id)->exists()) {
        return redirect()->back()
            ->withInput()
            ->with('error', 'Gagal mengupdate, kode mata kuliah sudah digunakan.');
    }

    $matkul->update($validated);

    return redirect()->route('matkuls.index')->with('success', 'Matkul berhasil diupdate.');
}
    /**
```

```
     * Remove the specified resource from storage.
     */
    public function destroy(Matkul $matkul)
    {
        $matkul->delete();

        return redirect()->route('matkuls.index')->with('success', 'Matkul berhasil dihapus.');
    }
}
<?php

namespace App\Http\Controllers;

use App\Models\User;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Hash;
use Spatie\Permission\Models\Role;

class UserController extends Controller
{
    /**
     * Tampilkan halaman user (index + form create/edit).
     */
    public function index()
    {
        $users = User::whereHas('roles', function ($q) {
            $q->whereIn('name', ['dosenPA', 'dosen', 'mahasiswa']);
        })->with('roles')->get();

        $roles = Role::whereIn('name', ['dosenPA', 'dosen', 'mahasiswa'])->pluck('name');

        return inertia('admin/Index', [
            'users' => $users,
            'roles' => $roles,
        ]);
    }

    /**
     * Simpan user baru.
     */
    public function store(Request $request)
    {
        $request->validate([
            'name'     => 'required|string|max:255',
            'email'    => 'required|email|unique:users,email',
            'no'       => 'nullable|string|max:20',
            'nim'      => 'nullable|string|max:50',
            'fakultas' => 'nullable|string|max:100',
            'jurusan'  => 'nullable|string|max:100',
            'password' => 'required|string|min:6|confirmed',
            'role'     => 'required|in:dosenPA,dosen,mahasiswa',
        ]);

        $user = User::create([
            'name'     => $request->name,
            'email'    => $request->email,
            'no'       => $request->no,
            'nim'      => $request->nim,
            'fakultas' => $request->fakultas,
            'jurusan'  => $request->jurusan,
            'password' => Hash::make($request->password),
        ]);

        $user->assignRole($request->role);
```

```
        return redirect()->route('users.index')
->with('success', 'Akun berhasil ditambahkan!');
    }

    /**
     * Update user.
     */
public function update(Request $request, User $user)
{
    $request->validate([
        'name'      => 'nullable|string|max:255',
        'email'     => 'nullable|email|unique:users,email,' .
$user->id,
        'no'        => 'nullable|string|max:20',
        'nim'       => 'nullable|string|max:50',
        'fakultas'  => 'nullable|string|max:100',
        'jurusan'   => 'nullable|string|max:100',
        'password'  => 'nullable|string|min:6|confirmed',
        'role'      =>
'nullable|in:dosenPA,dosen,mahasiswa',
    ]);

    $data = [];

    // hanya ambil field yang ada isinya
    if ($request->filled('name')) {
        $data['name'] = $request->name;
    }
    if ($request->filled('email')) {
        $data['email'] = $request->email;
    }
    if ($request->filled('no')) {
        $data['no'] = $request->no;
    }
    if ($request->filled('nim')) {
        $data['nim'] = $request->nim;
    }
    if ($request->filled('fakultas')) {
        $data['fakultas'] = $request->fakultas;
    }
    if ($request->filled('jurusan')) {
        $data['jurusan'] = $request->jurusan;
    }
    if ($request->filled('password')) {
        $data['password'] = Hash::make($request-
>password);
    }

    // update hanya field yang ada isinya
    if (!empty($data)) {
        $user->update($data);
    }

    // update role kalau ada
    if ($request->filled('role')) {
        $user->syncRoles([$request->role]);
    }

    return redirect()
        ->route('users.index')
        ->with('success', 'Akun berhasil diperbarui!');
}

    /**
     * Hapus user.
     */
    public function destroy(User $user)
    {
        // pastikan tidak menghapus admin
        if ($user->hasRole('admin')) {
```

```
        return redirect()->back()->with('error', 'Admin
tidak boleh dihapus!');
    }

    $user->delete();

    return redirect()->route('users.index')-
>with('success', 'Akun berhasil dihapus!');
  }
}
```

ROUTE:

```
<?php

use Illuminate\Support\Facades\Route;
use Inertia\Inertia;
use App\Http\Controllers\MatkulController;
use App\Http\Controllers\KegiatanController;
use App\Http\Controllers\AgendaKuliahController;
use App\Http\Controllers\JadwalController;
use App\Http\Controllers\ChatController;
use App\Http\Controllers\UserController;

Route::get('/', function () {
  return Inertia::render('welcome');
})->name('home');

Route::middleware(['auth', 'verified'])->group(function
() {
  Route::get('dashboard', function () {
    return Inertia::render('dashboard');
  })->name('dashboard');
});

// index
Route::get('/matkuls', [MatkulController::class,
'index'])->name('matkuls.index');

Route::middleware(['auth',
'role:admin|dosen|dosenPA'])->group(function () {

// create
Route::get('/matkuls/create', [MatkulController::class,
'create'])->name('matkuls.create');

// store
Route::post('/matkuls', [MatkulController::class,
'store'])->name('matkuls.store');

// show
Route::get('/matkuls/{matkul}',
[MatkulController::class, 'show'])-
>name('matkuls.show');

// edit
Route::get('/matkuls/{matkul}/edit',
[MatkulController::class, 'edit'])->name('matkuls.edit');

// update
Route::put('/matkuls/{matkul}',
[MatkulController::class, 'update'])-
>name('matkuls.update');

// delete
Route::delete('/matkuls/{matkul}',
[MatkulController::class, 'destroy'])-
>name('matkuls.destroy');
});

Route::middleware(['auth',
'role:admin|dosen|dosenPA'])->group(function () {
Route::get('/kegiatans', [KegiatanController::class,
'index'])->name('kegiatans.index');
Route::post('/kegiatans', [KegiatanController::class,
'store'])->name('kegiatans.store');
```

```
Route::put('/kegiatans/{kegiatan}',
[KegiatanController::class, 'update'])-
>name('kegiatans.update');

Route::delete('/kegiatans/{kegiatan}',
[KegiatanController::class, 'destroy'])-
>name('kegiatans.destroy');

});


// menampilkan semua agenda

Route::get('/agenda-kuliah',
[AgendaKuliahController::class, 'index'])-
>name('agenda-kuliah.index');


// menyimpan agenda baru

Route::post('/agenda-kuliah',
[AgendaKuliahController::class, 'store'])-
>name('agenda-kuliah.store');


// menampilkan detail agenda tertentu

Route::get('/agenda-kuliah/{agendaKuliah}',
[AgendaKuliahController::class, 'show'])-
>name('agenda-kuliah.show');

Route::get('/hasil/{agendaKuliah}',
[AgendaKuliahController::class, 'show2'])-
>name('hasil');

// menghapus agenda

Route::delete('/agenda-kuliah/{agendaKuliah}',
[AgendaKuliahController::class, 'destroy'])-
>name('agenda-kuliah.destroy');


Route::post('/akademik/{akademik}/nilai',
[AgendaKuliahController::class, 'updateNilai'])-
>name('akademik.update-nilai');


Route::get('/Jadwal', [JadwalController::class, 'index'])-
>name('Jadwal.index');

Route::get('/Jadwal/create', [JadwalController::class,
'create'])->name('Jadwal.create');

Route::post('/Jadwal/store', [JadwalController::class,
'store'])->name('Jadwal.store');

Route::get('/Jadwal/edit/{Jadwal}',
[JadwalController::class, 'edit'])->name('Jadwal.edit');

Route::post('/Jadwal/update/{Jadwal}',
[JadwalController::class, 'update'])-
>name('Jadwal.update');

Route::post('Jadwal/selesai{jadwal}',
[JadwalController::class, 'selesai'])-
>name('Jadwal.selesai');

Route::delete('/Jadwal/delete/{Jadwal}',
[JadwalController::class, 'destroy'])-
>name('Jadwal.delete');


Route::get('/home', [ChatController::class, 'chatUsers'])-
>name('chat.users');

Route::get('/chat/{userId}', [ChatController::class,
'chatBox'])->name('chat.box');

Route::get('/users', [ChatController::class, 'index2'])-
>name('users');

Route::get('/messages', [ChatController::class,
'messages'])

    ->name('messages');

Route::post('/message', [ChatController::class,
'message'])

    ->name('message');

Route::post('/chat/end', [ChatController::class,
'deleteChat'])->name('chat.end');


Route::middleware(['auth', 'role:admin'])-
>group(function () {

    Route::get('/users', [UserController::class, 'index'])-
>name('users.index');

    Route::post('/users', [UserController::class, 'store'])-
>name('users.store');

    Route::put('/users/{user}', [UserController::class,
'update'])->name('users.update');

    Route::delete('/users/{user}', [UserController::class,
'destroy'])->name('users.destroy');

});


require __DIR__ . '/settings.php';

require __DIR__ . '/auth.php';
```

VIEWS:

```
import '../css/app.css';


import { createInertiaApp } from '@inertiajs/react';
import { createRoot } from 'react-dom/client';
import { initializeTheme } from './hooks/use-appearance';


const pages = {
  ...(import.meta.glob('./pages/**/*.tsx') as Record<string, () => Promise<{ default: any }>>),
  ...(import.meta.glob('./pages/**/*.jsx') as Record<string, () => Promise<{ default: any }>>),
};


const appName = import.meta.env.VITE_APP_NAME || 'Laravel';


createInertiaApp({
  title: (title) => `${title} - ${appName}`,
  resolve: async (name) => {
    const pathTsx = `./pages/${name}.tsx`;
    const pathJsx = `./pages/${name}.jsx`;


    if (pages[pathTsx]) {
      return (await pages[pathTsx]!()).default;
    } else if (pages[pathJsx]) {
      return (await pages[pathJsx]!()).default;
    } else {
      throw new Error(`Page not found: ${pathTsx} or ${pathJsx}`);
    }
  },
  setup({ el, App, props }) {
    const root = createRoot(el);
    root.render(<App {...props} />);
  },
  progress: {
    color: '#4B5563',
  },
});


initializeTheme();


import { type SharedData } from '@/types';
import { Head, Link, usePage } from '@inertiajs/react';
import { Player } from '@lottiefiles/react-lottie-player';
import { Calendar, FileText, MessageCircle, BookOpen } from "lucide-react";


export default function Welcome() {
    const { auth } = usePage<SharedData>().props;


    return (
        <>
            <Head title="SIAK Fakultas Teknik">
                <link rel="preconnect" href="https://fonts.bunny.net" />
                <link


href="https://fonts.bunny.net/css?family=instrument-sans:400,500,600"
                    rel="stylesheet"
                />
            </Head>


            <div className="flex min-h-screen flex-col bg-gradient-to-br from-[#f7fafc] via-[#f0f9ff] to-[#e6fffa] text-[#1b1b18] dark:from-[#0f172a] dark:via-[#1e293b] dark:to-[#0f172a]">
                {/* Navbar */}
                <header className="w-full px-6 py-4">
                    <nav className="flex items-center justify-between max-w-7xl mx-auto">
```

```
                <h1 className="text-lg font-extrabold
bg-gradient-to-r from-cyan-600 to-blue-600 bg-clip-text
text-transparent">
                    Teknik UMPAR
                </h1>
                <div className="flex items-center gap-
4">
                  {auth.user ? (
                    <Link
                      href={route('dashboard')}
                      className="rounded-md px-5
py-2 text-sm font-medium text-white bg-gradient-to-r
from-cyan-500 to-blue-500 shadow-md hover:shadow-
lg hover:opacity-90 transition"
                    >
                      Dashboard
                    </Link>
                  ) : (
                    <>
                      <Link
                        href={route('login')}
                        className="px-5 py-2 text-sm
font-medium text-cyan-700 hover:text-blue-600
transition"
                      >
                        Log in
                      </Link>
                      <Link
                        href={route('register')}
                        className="rounded-md px-5
py-2 text-sm font-medium text-white bg-gradient-to-r
from-blue-500 to-indigo-500 shadow-md
hover:shadow-lg hover:opacity-90 transition"
                      >
                        Register
                      </Link>
                    </>
                  )}
                </div>
              </nav>
            </header>

            {/* Hero Section */}
            <section className="flex flex-col-reverse
items-center justify-between max-w-7xl mx-auto px-6
py-12 lg:flex-row lg:py-24">
              {/* Text */}
              <div className="flex flex-col items-start
gap-6 text-center lg:text-left lg:max-w-lg">
                <h2 className="text-4xl font-extrabold
bg-gradient-to-r from-cyan-600 to-blue-700 bg-clip-text
text-transparent lg:text-5xl">
                  Sistem Informasi Akademik
                </h2>
                <p className="text-gray-600 dark:text-
gray-300 text-lg">
                  Fakultas Teknik Universitas
Muhammadiyah Parepare — solusi akademik modern
                  untuk mahasiswa: jadwal, nilai, KRS,
hingga konsultasi dosen.
                </p>
                <div className="flex gap-4 justify-
center lg:justify-start">
                  <Link
                    href={route('register')}
                    className="px-6 py-3 rounded-xl
bg-gradient-to-r from-cyan-500 to-blue-600 text-white
font-medium shadow-lg hover:opacity-90 transition"
                  >
                    Daftar Sekarang
                  </Link>
                  <Link
                    href={route('login')}
                    className="px-6 py-3 rounded-xl
border border-cyan-500 text-cyan-600 font-medium
hover:bg-cyan-50 transition"
                  >
                    Masuk
                  </Link>
```

```
          </div>
        </div>

        {/* Animation */}
        <div className="w-full lg:w-[500px] mb-8 lg:mb-0">
          <Player
            autoplay
            loop
            src="/img/logo.json"
            className="w-full h-full"
          />
        </div>
      </section>

      {/* Logo Section */}
      <section className="w-full bg-gradient-to-r from-cyan-50 to-blue-50 dark:from-gray-800 dark:to-gray-900 py-12">
        <div className="max-w-7xl mx-auto px-6">
          <h3 className="text-center text-2xl font-bold mb-8 bg-gradient-to-r from-cyan-600 to-blue-700 bg-clip-text text-transparent">
            Didukung Oleh
          </h3>
          <div className="grid grid-cols-2 md:grid-cols-4 gap-8 items-center justify-items-center">
            <img src="/img/aa.png" alt="Logo 1" className="h-20 object-contain opacity-80 hover:opacity-100 transition" />
            <img src="/img/bb.png" alt="Logo 2" className="h-20 object-contain opacity-80 hover:opacity-100 transition" />
            <img src="/img/cc.png" alt="Logo 3" className="h-20 object-contain opacity-80 hover:opacity-100 transition" />
            <img src="/img/dd.png" alt="Logo 4" className="h-20 object-contain opacity-80 hover:opacity-100 transition" />
          </div>
        </div>
      </section>

      {/* Features Section */}
      <section className="py-16 bg-gradient-to-br from-white to-cyan-50 dark:from-gray-900 dark:to-gray-800">
        <div className="max-w-7xl mx-auto px-6">
          <h3 className="text-3xl font-bold text-center mb-12 bg-gradient-to-r from-cyan-600 to-blue-700 bg-clip-text text-transparent">
            Solusi untuk Mahasiswa
          </h3>
          <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-4 gap-8">
            <div className="p-6 rounded-xl shadow-md bg-white/80 dark:bg-gray-800/80 hover:shadow-xl hover:scale-[1.02] transition">
              <Calendar className="h-10 w-10 text-cyan-600 mb-4" />
              <h4 className="font-semibold text-lg mb-2 text-cyan-700">Jadwal Kuliah</h4>
              <p className="text-gray-600 dark:text-gray-300 text-sm">
                Tidak perlu bingung mencari jadwal. Semua mata kuliah
                terorganisir rapi dalam satu sistem.
              </p>
            </div>
            <div className="p-6 rounded-xl shadow-md bg-white/80 dark:bg-gray-800/80 hover:shadow-xl hover:scale-[1.02] transition">
              <FileText className="h-10 w-10 text-indigo-600 mb-4" />
              <h4 className="font-semibold text-lg mb-2 text-indigo-700">KRS & Nilai</h4>
              <p className="text-gray-600 dark:text-gray-300 text-sm">
                Mudah mengisi KRS, melihat nilai, dan memantau perkembangan
                akademik setiap semester.
```

```
                        </p>
                    </div>
                    <div className="p-6 rounded-xl shadow-md bg-white/80 dark:bg-gray-800/80 hover:shadow-xl hover:scale-[1.02] transition">
                        <MessageCircle className="h-10 w-10 text-purple-600 mb-4" />
                        <h4 className="font-semibold text-lg mb-2 text-purple-700">Konsultasi Dosen</h4>
                        <p className="text-gray-600 dark:text-gray-300 text-sm">
                            Bingung dengan perkuliahan? Hubungi dosen pembimbing langsung
                            melalui sistem.
                        </p>
                    </div>
                    <div className="p-6 rounded-xl shadow-md bg-white/80 dark:bg-gray-800/80 hover:shadow-xl hover:scale-[1.02] transition">
                        <BookOpen className="h-10 w-10 text-pink-500 mb-4" />
                        <h4 className="font-semibold text-lg mb-2 text-pink-600">Materi & E-Learning</h4>
                        <p className="text-gray-600 dark:text-gray-300 text-sm">
                            Akses materi kuliah, tugas, dan e-learning kapan saja,
                            di mana saja.
                        </p>
                    </div>
                </div>
            </div>
        </section>

        {/* Footer */}
        <footer className="py-6 text-center text-sm text-gray-600 dark:text-gray-400">
            © {new Date().getFullYear()} Fakultas Teknik Universitas Muhammadiyah parepare. Semua Hak Dilindungi.
        </footer>
    </>
  );
}

import { Head, useForm } from '@inertiajs/react';
import { LoaderCircle } from 'lucide-react';
import { FormEventHandler, useState } from 'react';

import InputError from '@/components/input-error';
import TextLink from '@/components/text-link';
import { Button } from '@/components/ui/button';
import { Checkbox } from '@/components/ui/checkbox';
import { Input } from '@/components/ui/input';
import { Label } from '@/components/ui/label';
import AuthLayout from '@/layouts/auth-layout';

type LoginForm = {
  nim: string;
  password: string;
  remember: boolean;
};

interface LoginProps {
  status?: string;
  canResetPassword: boolean;
}

export default function Login({ status, canResetPassword }: LoginProps) {
  const { data, setData, post, processing, errors, reset } = useForm<Required<LoginForm>>({
    nim: '',
    password: '',
    remember: false,
```

```
    });

    const [role, setRole] = useState<'mahasiswa' | 'dosen'>('mahasiswa');

    const submit: FormEventHandler = (e) => {
        e.preventDefault();
        post(route('login'), {
            onFinish: () => reset('password'),
        });
    };

    return (
        <AuthLayout title="Log in" description="Masukkan Nim/NIDN dan password untuk log in">
            <Head title="Log in" />

            <form className="flex flex-col gap-6" onSubmit={submit}>
                <div className="grid gap-6">
                    {/* Pilih role */}
                    <div className="grid gap-2">
                        <Label htmlFor="role">Login sebagai</Label>
                        <select
                            id="role"
                            value={role}
                            onChange={(e) => setRole(e.target.value as 'mahasiswa' | 'dosen')}
                            className="rounded-xl border p-2"
                        >
                            <option value="mahasiswa">Mahasiswa</option>
                            <option value="dosen">Dosen</option>
                        </select>
                    </div>

                    {/* Input NIM atau NIDN */}
                    <div className="grid gap-2">
                        <Label htmlFor="nim">{role === 'mahasiswa' ? 'NIM' : 'NIDN'}</Label>
                        <Input
                            id="nim"
                            type="number"
                            required
                            autoFocus
                            tabIndex={1}
                            autoComplete="nim"
                            value={data.nim}
                            onChange={(e) => setData('nim', e.target.value)}
                            placeholder={`Masukkan ${role === 'mahasiswa' ? 'NIM' : 'NIDN'} anda`}
                        />
                        <InputError message={errors.nim} />
                    </div>

                    {/* Password */}
                    <div className="grid gap-2">
                        <Input
                            id="password"
                            type="password"
                            required
                            tabIndex={2}
                            autoComplete="current-password"
                            value={data.password}
                            onChange={(e) => setData('password', e.target.value)}
                            placeholder="Password"
                        />
                        <InputError message={errors.password} />
                    </div>
```

```
                {/* Remember me */}
                <div className="flex items-center space-x-3">
                  <Checkbox
                    id="remember"
                    name="remember"
                    checked={data.remember}
                    onClick={() => setData('remember', !data.remember)}
                    tabIndex={3}
                  />
                  <Label htmlFor="remember">Remember me</Label>
                </div>

                {/* Submit button */}
                <Button type="submit" className="mt-4 w-full" tabIndex={4} disabled={processing}>
                  {processing && <LoaderCircle className="h-4 w-4 animate-spin" />}
                  Log in
                </Button>
              </div>

              <div className="text-muted-foreground text-center text-sm">
                Belum mempunyai akun?{' '}
                <TextLink href={route('register')} tabIndex={5}>
                  daftar
                </TextLink>
              </div>
            </form>

            {status && <div className="mb-4 text-center text-sm font-medium text-green-600">{status}</div>}
        </AuthLayout>
    );
}

import { Head, useForm } from '@inertiajs/react';
import { LoaderCircle } from 'lucide-react';
import { FormEventHandler, useState } from 'react';

import InputError from '@/components/input-error';
import TextLink from '@/components/text-link';
import { Button } from '@/components/ui/button';
import { Input } from '@/components/ui/input';
import { Label } from '@/components/ui/label';
import AuthLayout from '@/layouts/auth-layout';

type RegisterForm = {
    name: string;
    email: string;
    password: string;
    password_confirmation: string;
    nim: string;
    fakultas: string;
    jurusan: string;
    no: string;
};

export default function Register() {
    const { data, setData, post, processing, errors, reset } = useForm<Required<RegisterForm>>({
        name: '',
        email: '',
        password: '',
        password_confirmation: '',
        nim: '',
        fakultas: '',
        jurusan: '',
```

```
    no: '',
  });

  const [role, setRole] = useState<'mahasiswa' | 'dosen'>('mahasiswa');

  const submit: FormEventHandler = (e) => {
    e.preventDefault();
    post(route('register'), {
      onFinish: () => reset('password', 'password_confirmation'),
    });
  };

  // Nomor WhatsApp (dengan format +62)
  const handleNoWaChange = (e: React.ChangeEvent<HTMLInputElement>) => {
    if (!e.target.value.startsWith('62')) {
      setData('no', '62' + e.target.value.replace(/[^0-9]/g, '').slice(2));
    } else {
      setData('no', e.target.value);
    }
  };

  return (
    <AuthLayout title="Buat Akun" description="Silahkan buat akun jika belum mempunyai akun">
      <Head title="Register" />
      <form className="flex flex-col gap-6" onSubmit={submit}>
        <div className="grid gap-6">
          {/* Pilih role */}
          <div className="grid gap-2">
            <Label htmlFor="role">Daftar sebagai</Label>
            <select
              id="role"
              value={role}
              onChange={(e) => setRole(e.target.value as 'mahasiswa' | 'dosen')}
              className="rounded-md border px-3 py-2 text-sm"
              disabled={processing}
            >
              <option value="mahasiswa">Mahasiswa</option>
              <option value="dosen">Dosen</option>
            </select>
          </div>

          <div className="grid gap-2">
            <Label htmlFor="name">Nama lengkap</Label>
            <Input
              id="name"
              type="text"
              required
              autoFocus
              tabIndex={1}
              autoComplete="name"
              value={data.name}
              onChange={(e) => setData('name', e.target.value)}
              disabled={processing}
              placeholder="Nama lengkap"
            />
            <InputError message={errors.name} className="mt-2" />
          </div>

          <div className="grid gap-2">
            <Label htmlFor="email">Email</Label>
```

```
                <Input
                    id="email"
                    type="email"
                    required
                    tabIndex={2}
                    autoComplete="email"
                    value={data.email}
                    onChange={(e) => setData('email', e.target.value)}
                    disabled={processing}
                    placeholder="Masukkan email anda"
                />
                <InputError message={errors.email} />
            </div>

            <div className="grid gap-2">
                <Label htmlFor="no">Nomor WA</Label>
                <Input
                    id="no_wa"
                    type="number"
                    required
                    tabIndex={3}
                    autoComplete="no"
                    value={data.no}
                    onChange={handleNoWaChange}
                    disabled={processing}
                    placeholder="Nomor WhatsApp"
                />
                <InputError message={errors.no} className="mt-2" />
                <p className="text-xs text-gray-500 mt-2">
                    Masukkan nomor WhatsApp Anda dengan format setelah: <strong>+62</strong> di awal, tanpa 0
                    (contoh: <strong>81234567890</strong>).
                </p>
            </div>

            {/* NIM/NIDN */}
            <div className="grid gap-2">
                <Label htmlFor="nim">{role === 'mahasiswa' ? 'NIM' : 'NIDN'}</Label>
                <Input
                    id="nim"
                    type="text"
                    required
                    tabIndex={4}
                    value={data.nim}
                    onChange={(e) => setData('nim', e.target.value)}
                    disabled={processing}
                    placeholder={role === 'mahasiswa' ? 'Nomor Induk Mahasiswa' : 'Nomor Induk Dosen Nasional'}
                />
                <InputError message={errors.nim} />
            </div>

            {/* Fakultas & Jurusan hanya untuk mahasiswa */}
            {role === 'mahasiswa' && (
                <>
                    <div className="grid gap-2">
                        <Label htmlFor="fakultas">Fakultas</Label>
                        <select
                            id="fakultas"
                            required
                            tabIndex={5}
                            value={data.fakultas}
```

```
                        onChange={(e) => {
                            setData('fakultas',
e.target.value);
                            setData('jurusan', '');
                        }}
                        disabled={processing}
                        className="border-input
rounded-md border px-3 py-2 text-sm"
                    >
                        <option value="">Pilih
Fakultas</option>
                        <option value="Fakultas Agama
Islam">Fakultas Agama Islam</option>
                        <option value="Fakultas
Ekonomi dan Bisnis">Fakultas Ekonomi dan
Bisnis</option>
                        <option value="Fakultas
Hukum">Fakultas Hukum</option>
                        <option value="Fakultas Ilmu
Kesehatan">Fakultas Ilmu Kesehatan</option>
                        <option value="Fakultas
Keguruan dan Ilmu Pendidikan">
                            Fakultas Keguruan dan Ilmu
Pendidikan
                        </option>
                        <option value="Fakultas
Pertanian, Peternakan, dan Perikanan">
                            Fakultas Pertanian,
Peternakan, dan Perikanan
                        </option>
                        <option value="Fakultas
Teknik">Fakultas Teknik</option>
                    </select>
                    <InputError
message={errors.fakultas} />
                </div>

                <div className="grid gap-2">
                    <Label htmlFor="jurusan">Jurusan
/ Prodi</Label>
                    <select
                        id="jurusan"
                        required
                        tabIndex={6}
                        value={data.jurusan}
                        onChange={(e) =>
setData('jurusan', e.target.value)}
                        disabled={processing}
                        className="border-input
rounded-md border px-3 py-2 text-sm"
                    >
                        <option value="">Pilih
Jurusan</option>
                        {data.fakultas === 'Fakultas
Agama Islam' && (
                            <>
                                <option value="Pendidikan
Agama Islam">Pendidikan Agama Islam</option>
                                <option value="Dakwah dan
BPI">Dakwah dan Bimbingan Penyuluhan
Islam</option>
                                <option value="Perbankan
Syariah">Perbankan Syariah</option>
                                <option
value="PGRA">Pendidikan Guru Raudhatul
Athfal</option>
                            </>
                        )}
                        {data.fakultas === 'Fakultas
Ekonomi dan Bisnis' && (
                            <>
                                <option
value="Akuntansi">Akuntansi</option>
                                <option
value="Manajemen">Manajemen</option>
                                <option
value="Pembangunan">Pembangunan</option>
                            </>
                        )}
                        {data.fakultas === 'Fakultas
Hukum' && (
```

```
                                <option value="Ilmu Hukum">Ilmu Hukum</option>

                            )}

                            {data.fakultas === 'Fakultas Ilmu Kesehatan' && (

                                <option value="Ilmu Kesehatan Masyarakat">Ilmu Kesehatan Masyarakat</option>

                            )}

                            {data.fakultas === 'Fakultas Keguruan dan Ilmu Pendidikan' && (

                                <>

                                <option value="Pendidikan Matematika">Pendidikan Matematika</option>

                                <option value="Pendidikan Bahasa Inggris">Pendidikan Bahasa Inggris</option>

                                <option value="Pendidikan Luar Sekolah">Pendidikan Luar Sekolah (PLS)</option>

                                <option value="Biologi">Biologi</option>

                                </>

                            )}

                            {data.fakultas === 'Fakultas Pertanian, Peternakan, dan Perikanan' && (

                                <>

                                <option value="Agroteknologi">Agroteknologi</option>

                                <option value="Agribisnis">Agribisnis</option>

                                <option value="Ilmu Peternakan">Ilmu Peternakan</option>

                                <option value="Budidaya Perairan">Budidaya Perairan</option>

                                </>

                            )}

                            {data.fakultas === 'Fakultas Teknik' && (

                                <>

                                <option value="Teknik Sipil">Teknik Sipil</option>

                                <option value="Teknik Elektro">Teknik Elektro</option>

                                <option value="Teknik Informatika">Teknik Informatika</option>

                                </>

                            )}

                        </select>

                        <InputError message={errors.jurusan} />

                    </div>

                </>

            )}


            <div className="grid gap-2">

                <Label htmlFor="password">Password</Label>

                <Input

                    id="password"

                    type="password"

                    required

                    tabIndex={7}

                    autoComplete="new-password"

                    value={data.password}

                    onChange={(e) => setData('password', e.target.value)}

                    disabled={processing}

                    placeholder="Password"

                />

                <InputError message={errors.password} />

            </div>


            <div className="grid gap-2">

                <Label htmlFor="password_confirmation">Confirm password</Label>

                <Input

                    id="password_confirmation"
```

```
                    type="password"
                    required
                    tabIndex={8}
                    autoComplete="new-password"
                    value={data.password_confirmation}
                    onChange={(e) =>
setData('password_confirmation', e.target.value)}
                    disabled={processing}
                    placeholder="Confirm password"
                  />
                  <InputError
message={errors.password_confirmation} />
                </div>

                <Button type="submit" className="mt-2
w-full" tabIndex={9} disabled={processing}>
                  {processing && <LoaderCircle
className="h-4 w-4 animate-spin" />}
                  Simpan
                </Button>
              </div>

              <div className="text-muted-foreground
text-center text-sm">
                Sudah punya akun?{' '}
                <TextLink href={route('login')}
tabIndex={10}>
                  Log in
                </TextLink>
              </div>
            </form>
          </AuthLayout>
    );
}

import { Head } from '@inertiajs/react';
import AppLayout from '@/layouts/app-layout';
import { type BreadcrumbItem } from '@/types';
import { GraduationCap, Calendar, FileText,
MessageCircle, BookOpen, Users } from "lucide-react";
import { Player } from '@lottiefiles/react-lottie-player';

const breadcrumbs: BreadcrumbItem[] = [
    { title: 'Dashboard', href: '/dashboard' },
];

export default function Dashboard() {
    return (
        <AppLayout breadcrumbs={breadcrumbs}>
            <Head title="Dashboard" />

            <div className="flex flex-col gap-10 p-6 bg-
gradient-to-br from-[#f7fafc] via-[#f0f9ff] to-[#e6fffa]
dark:from-[#0f172a] dark:via-[#1e293b] dark:to-
[#0f172a] rounded-xl shadow-md min-h-[80vh]">

                {/* Header Section */}
                <div className="text-center lg:text-left">
                    <h1 className="text-3xl font-extrabold
bg-gradient-to-r from-cyan-600 to-blue-700 bg-clip-text
text-transparent">
                        Selamat Datang di SIAK Fakultas
Teknik
                    </h1>
                    <p className="mt-2 text-gray-600
dark:text-gray-300">
                        Sistem Informasi Akademik modern
untuk mendukung kebutuhan mahasiswa Fakultas
Teknik Universitas Muhammadiyah Parepare.
                    </p>
                </div>

                {/* Animation Section */}
                <div className="grid grid-cols-1 md:grid-
cols-3 gap-6 justify-items-center">
                    <div className="w-full max-w-xs">
```

```
                <img  src="/img/cc.png" className="w-full " />

              </div>

              <div className="w-full max-w-xs">

                <Player autoplay loop src="/img/logo.json" className="w-full h-full" />

              </div>

              <div className="w-full max-w-xs">

                <img src="/img/dd.png" className="w-full " />

              </div>

            </div>


            {/* About App Section */}

            <div className="grid grid-cols-1 lg:grid-cols-2 gap-8">

              {/* Tentang Aplikasi */}

              <div className="p-6 bg-white/80 dark:bg-gray-800/80 rounded-xl shadow hover:shadow-lg transition">

                <h2 className="text-xl font-semibold text-cyan-700 mb-3 flex items-center gap-2">

                  <GraduationCap className="h-6 w-6 text-cyan-600" />

                  Tentang Aplikasi

                </h2>

                <p className="text-gray-600 dark:text-gray-300 text-sm leading-relaxed">

                  Aplikasi ini membantu mahasiswa dalam mengelola aktivitas akademik, mulai dari jadwal kuliah,

                  pengisian KRS, pemantauan nilai, hingga konsultasi dengan dosen. Semua dirancang agar perkuliahan lebih mudah,

                  terorganisir, dan transparan.

                </p>

              </div>


              {/* Fitur Utama */}

              <div className="grid grid-cols-2 gap-4">

                <a href='/matkuls' className="p-4 rounded-xl bg-white/80 dark:bg-gray-800/80 shadow hover:scale-[1.02] transition">

                  <Calendar className="h-8 w-8 text-cyan-600 mb-2" />

                  <h3 className="font-semibold text-cyan-700">Mata Kuliah</h3>

                  <p className="text-xs text-gray-600 dark:text-gray-300">Mudah akses jadwal kapan saja.</p>

                </a>

                <a href='/agenda-kuliah' className="p-4 rounded-xl bg-white/80 dark:bg-gray-800/80 shadow hover:scale-[1.02] transition">

                  <FileText className="h-8 w-8 text-indigo-600 mb-2" />

                  <h3 className="font-semibold text-indigo-700">KRS & Nilai</h3>

                  <p className="text-xs text-gray-600 dark:text-gray-300">Pantau akademik dengan mudah.</p>

                </a>

                <a href='/home' className="p-4 rounded-xl bg-white/80 dark:bg-gray-800/80 shadow hover:scale-[1.02] transition">

                  <MessageCircle className="h-8 w-8 text-purple-600 mb-2" />

                  <h3 className="font-semibold text-purple-700">Konsultasi online</h3>

                  <p className="text-xs text-gray-600 dark:text-gray-300">Langsung hubungi dosen pembimbing.</p>

                </a>

                <a href='/Jadwal' className="p-4 rounded-xl bg-white/80 dark:bg-gray-800/80 shadow hover:scale-[1.02] transition">

                  <Users className="h-8 w-8 text-pink-500 mb-2" />

                  <h3 className="font-semibold text-pink-600">Konsultasi ofline</h3>

                  <p className="text-xs text-gray-600 dark:text-gray-300">Atur pertemuan.</p>

                </a>

              </div>
```

```
            </div>
          </div>
        </AppLayout>
    );
}

import { useState } from "react";
import Swal from "sweetalert2";
import { Link, router } from "@inertiajs/react";
import AppLayout from '@/layouts/app-layout';
import { Button } from '@/components/ui/button';
import { IconClock, IconTrash, IconEdit, IconUser, IconList, IconCheck } from "@tabler/icons-react";

const Index = ({ Jadwal }) => {
    const [isDeleting, setIsDeleting] = useState(false);

    const Delete = (id) => {
        Swal.fire({
            title: "Apakah Anda yakin?",
            text: "Anda tidak akan dapat mengembalikan ini!",
            icon: "warning",
            showCancelButton: true,
            confirmButtonColor: "#3085d6",
            cancelButtonColor: "#d33",
            confirmButtonText: "Ya, hapus saja!",
        }).then((result) => {
            if (result.isConfirmed) {
                setIsDeleting(true);
                router.delete(route("Jadwal.delete", id), {
                    onSuccess: () => {
                        setIsDeleting(false);
                        Swal.fire("Dihapus!", "Jadwal telah dihapus.", "success");
                    },
                    onError: () => {
                        setIsDeleting(false);
                        Swal.fire("Gagal!", "Terjadi kesalahan saat menghapus jadwal.", "error");
                    },
                });
            }
        });
    };
    const handleSelesai = (id) => {
        Swal.fire({
            title: "Tandai sebagai selesai?",
            text: "Data ini akan diupdate menjadi 'Selesai'.",
            icon: "question",
            showCancelButton: true,
            confirmButtonColor: "#16a34a",
            cancelButtonColor: "#d33",
            confirmButtonText: "Ya, selesai",
        }).then((result) => {
            if (result.isConfirmed) {
                router.post(route("Jadwal.selesai", id), {
                    onSuccess: () => {
                        Swal.fire("Berhasil!", "Status telah diubah menjadi selesai.", "success");
                    },
                    onError: () => {
                        Swal.fire("Gagal!", "Gagal menyelesaikan jadwal.", "error");
                    },
                });
            }
        });
    };

    return (
        <AppLayout>
```

```
        <div className="container mx-auto px-4 py-8">

            <div className="text-center text-gray-900 dark:text-white">

                <h1 className="text-3xl xl:text-5xl font-bold mb-6">Layanan Tiket</h1>

                <p className="text-gray-600 dark:text-gray-400 mb-4">

                    Kelola jadwal helpdesk dengan mudah dan cepat.

                </p>

                <Link href={route("Jadwal.create")}>

                    <Button className="flex items-center justify-center gap-2 bg-blue-600 hover:bg-blue-700 text-white font-semibold py-2 px-4 rounded-lg shadow-lg transition-all duration-300">

                        <IconList className="w-5 h-5" />

                        Tambah Jadwal

                    </Button>

                </Link>

            </div>


            {/* Daftar Jadwal */}

            <div className="mt-8 grid grid-cols-1 md:grid-cols-2 xl:grid-cols-3 gap-6">

                {Jadwal && Jadwal.length > 0 ? (

                    Jadwal.map((item, index) => (

                        <div

                            key={index}

                            className="bg-white dark:bg-gray-800 shadow-lg rounded-lg overflow-hidden transition-transform transform hover:scale-105 duration-300"

                        >

                            <div className="p-6">

                                {/* Nama */}

                                <h2 className="text-xl font-semibold text-gray-800 dark:text-gray-200 flex items-center gap-2">

                                    <IconUser className="w-6 h-6 text-blue-500" />

                                    {item.nama}

                                </h2>


                                {/* NIM & Fakultas/Jurusan */}

                                <p className="text-gray-600 dark:text-gray-400 mt-2">NIM: {item.nim}</p>

                                <p className="text-gray-600 dark:text-gray-400">Fakultas: {item.fakultas} - {item.jurusan}</p>


                                {/* Waktu */}

                                <div className="mt-3 flex items-center gap-2 text-sm">

                                    <IconClock className="w-5 h-5 text-yellow-500" />

                                    <span className="font-semibold text-gray-700 dark:text-gray-300">{item.waktu}</span>

                                </div>


                                {/* Antrian & Status */}

                                <div className="mt-4 space-y-2">

                                    {/* Antrian */}

                                    <span

                                        className={`px-3 py-1 text-xs font-semibold rounded-full w-fit ${item.antrian <= 5

                                            ? "bg-green-500 text-white"

                                            : item.antrian <= 9

                                                ? "bg-yellow-500 text-white"

                                                : "bg-red-500 text-white"

                                            }`}

                                    >

                                        Antrian: {item.antrian}

                                    </span>
```

```
                        {/* Status */}
                        <span
                            className={`px-3 py-1 text-xs font-semibold rounded-full w-fit flex items-center gap-1 ${item.status.toLowerCase() === "selesai"
                                ? "bg-green-600 text-white"
                                : "bg-yellow-500 text-white"
                                }`}
                        >
                            Status:
                            {item.status.toLowerCase() === "selesai" ? (
                                <>
                                    <IconCheck className="w-4 h-4 text-white" />
                                    Selesai
                                </>
                            ) : (
                                <>
                                    <IconClock className="w-4 h-4 text-white" />
                                    {item.status}
                                </>
                            )}
                        </span>
                    </div>

                    {/* Masalah */}
                    <p className="text-gray-700 dark:text-gray-300 mt-4 text-sm">
                        {item.masalah.length > 100 ? `${item.masalah.substring(0, 100)}...` : item.masalah}
                    </p>
                </div>

                {/* Footer: Edit & Hapus */}
                <div className="flex justify-between items-center bg-gray-100 dark:bg-gray-900 p-4">
                    <Link
                        href={route("Jadwal.edit", item.id)}
                        className="text-blue-500 hover:text-blue-700 font-medium flex items-center"
                    >
                        <IconEdit className="w-5 h-5 mr-1" /> Edit
                    </Link>
                    <button
                        onClick={() => Delete(item.id)}
                        disabled={isDeleting}
                        className="text-red-500 hover:text-red-700 font-medium flex items-center"
                    >
                        <IconTrash className="w-5 h-5 mr-1" /> Hapus
                    </button>
                </div>

                {/* Tambahkan tombol selesaikan jika belum selesai */}
                {item.status.toLowerCase() !== "selesai" && (
                    <div className="flex justify-center bg-gray-50 dark:bg-gray-800 p-3 border-t border-gray-200 dark:border-gray-700">
                        <button
                            onClick={() => handleSelesai(item.id)}
                            className="text-green-600 hover:text-green-800 font-medium flex items-center gap-1"
                        >
                            <IconCheck className="w-5 h-5" />
                            Tandai Selesai
```

```
                                </button>
                            </div>
                        )}
                    </div>
                ))
            ) : (
                <p className="text-gray-500 text-center col-span-full">Tidak ada jadwal yang tersedia</p>
            )}
        </div>
    </div>
  </AppLayout>
  );
};

export default Index;
```

```
import React from 'react'
import { Link, usePage, router } from '@inertiajs/react'
import AppLayout from '@/layouts/app-layout';

import { useState } from 'react'
export default function Index() {
  const { kegiatans, flash } = usePage().props

  const handleDelete = (id) => {
    if (confirm('Yakin ingin menghapus data ini?')) {
      router.delete(`/kegiatans/${id}`)
    }
  }

  const [showCreate, setShowCreate] = useState(false);
  const [showEdit, setShowEdit] = useState(false);
  const [editData, setEditData] = useState(null);
  return (
    <AppLayout>
      <div className="p-6">
        <h1 className="text-2xl font-bold mb-4">Daftar Kegiatan</h1>

        {flash.message && (
          <div className="bg-green-100 text-green-700 p-2 mb-4 rounded">
            {flash.message}
          </div>
        )}

        <button onClick={() => setShowCreate(true)} className="bg-blue-500 text-white px-4 py-2 rounded mb-4 inline-block">Tambah</button>

        <table className="w-full mt-4 border">
          <thead>
            <tr className="bg-gray-200">
              <th className="p-2 border">No</th>
              <th className="p-2 border">Nama</th>
              <th className="p-2 border">Aksi</th>
            </tr>
          </thead>
          <tbody>
            {kegiatans.map((item, index) => (
              <tr key={item.id}>
                <td className="p-2 border text-center">{index + 1}</td>
                <td className="p-2 border">{item.nama}</td>
                <td className="p-2 border text-center space-x-2">
                  <button onClick={() => {
```

```
                        setEditData(item)
                        setShowEdit(true)
                    }} className="bg-yellow-500 text-white px-3 py-1 rounded">Edit</button>
                    <button onClick={() => handleDelete(item.id)} className="bg-red-500 text-white px-3 py-1 rounded">Hapus</button>
                </td>
            </tr>
        ))}
    </tbody>
</table>

{/* Modal Create */}
{showCreate && (
    <div className="fixed inset-0 bg-black/50 flex items-center justify-center z-50">
        <div className="bg-white p-6 rounded-lg w-11/12 sm:w-2/3 md:w-1/2 lg:w-1/3">
            <h2 className="text-xl font-bold mb-4">Tambah Kegiatan</h2>
            <form onSubmit={(e) => { e.preventDefault(); router.post('/kegiatans', { nama: e.target.nama.value }); setShowCreate(false); }}>
                <div className="mb-4">
                    <label className="block mb-1">Nama</label>
                    <input type="text" name="nama" className="border p-2 w-full" />
                </div>
                <div className="text-right space-x-2">
                    <button type="button" onClick={() => setShowCreate(false)} className="px-4 py-2 bg-gray-400 text-white rounded">Batal</button>
                    <button type="submit" className="px-4 py-2 bg-blue-500 text-white rounded">Simpan</button>
                </div>
            </form>
        </div>
    </div>
)}

{/* Modal Edit */}
{showEdit && editData && (
    <div className="fixed inset-0 bg-black/50 flex items-center justify-center z-50">
        <div className="bg-white p-6 rounded-lg w-11/12 sm:w-2/3 md:w-1/2 lg:w-1/3">
            <h2 className="text-xl font-bold mb-4">Edit Kegiatan</h2>
            <form onSubmit={(e) => { e.preventDefault(); router.put(`/kegiatans/${editData.id}`, { nama: e.target.nama.value }); setShowEdit(false); }}>
                <div className="mb-4">
                    <label className="block mb-1">Nama</label>
                    <input defaultValue={editData.nama} type="text" name="nama" className="border p-2 w-full" />
                </div>
                <div className="text-right space-x-2">
                    <button type="button" onClick={() => setShowEdit(false)} className="px-4 py-2 bg-gray-400 text-white rounded">Batal</button>
                    <button type="submit" className="px-4 py-2 bg-yellow-500 text-white rounded">Update</button>
                </div>
            </form>
        </div>
    </div>
)}
</div>
</AppLayout>
)
}
```

```
// Create.jsx

export function Create() {

  return (

    <div className="p-6">

      <h1 className="text-2xl font-bold mb-
4">Tambah Kegiatan</h1>

      <form method="post" onSubmit={(e) => {
e.preventDefault(); router.post('/kegiatans', { nama:
e.target.nama.value }) }}>

        <div className="mb-4">

          <label className="block">Nama</label>

          <input type="text" name="nama"
className="border p-2 w-full" />

        </div>

        <button className="bg-blue-500 text-white
px-4 py-2 rounded">Simpan</button>

      </form>

    </div>

  )

}


// Edit.jsx

export function Edit({ kegiatan }) {

  return (

    <div className="p-6">

      <h1 className="text-2xl font-bold mb-4">Edit
Kegiatan</h1>

      <form method="post" onSubmit={(e) => {
e.preventDefault();
router.put(`/kegiatans/${kegiatan.id}`, { nama:
e.target.nama.value }) }}>

        <div className="mb-4">

          <label className="block">Nama</label>

          <input defaultValue={kegiatan.nama}
type="text" name="nama" className="border p-2 w-
full" />

        </div>

        <button className="bg-yellow-500 text-
white px-4 py-2 rounded">Update</button>

      </form>

    </div>

  )

}

import React, { useState } from 'react';

import { Link, router, usePage } from '@inertiajs/react';

import { Eye, Edit, Trash2, Plus } from 'lucide-react';

import AppLayout from '@/layouts/app-layout';


export default function Index({ matkuls }) {

 const { auth } = usePage().props;

 const [selectedSemester, setSelectedSemester] =
useState(1);


 const handleDelete = (id) => {

  if (confirm('Yakin ingin menghapus mata kuliah
ini?')) {

   router.delete(`/matkuls/${id}`);

  }

 };


 // cek role

 const allowedRoles = ['dosen', 'dosenPA', 'admin'];

 const hasPermission = auth?.roles?.some((role) =>
allowedRoles.includes(role));


 const semesters = Array.from({ length: 8 }, (_, i) => i
+ 1);

const filteredMatkuls = matkuls.filter(

 (m) => Number(m.semester) ===
Number(selectedSemester)

);

console.log("SELECTED SEMESTER:",
selectedSemester);

console.log("Contoh MATKUL:",
matkuls[0]?.semester, typeof matkuls[0]?.semester);

console.log("FILTERED:", filteredMatkuls);
```

```
 console.log('MATKULS di produksi:', matkuls);

 return (

   <AppLayout>

     <div className="min-h-screen bg-gradient-to-br
from-[#f7fafc] via-[#f0f9ff] to-[#e6fffa]

                 dark:from-[#0f172a] dark:via-[#1e293b]
dark:to-[#0f172a] py-8 px-6">


      <div className="max-w-6xl mx-auto bg-white
dark:bg-gray-900 shadow-xl rounded-2xl p-6 md:p-8">


        {/* Header */}

        <div className="flex flex-wrap justify-between
items-center gap-3 mb-6">

          <h1 className="text-2xl font-extrabold bg-
gradient-to-r from-cyan-600 to-blue-700

                    bg-clip-text text-transparent">

            Daftar Mata Kuliah

          </h1>

          {hasPermission && (

            <Link

              href="/matkuls/create"

              className="inline-flex items-center px-4 py-
2 rounded-xl font-medium

                    bg-gradient-to-r from-cyan-500 to-
blue-600 text-white shadow-md

                    hover:opacity-90 transition"

            >

              <Plus className="w-4 h-4 mr-2" />

              Tambah

            </Link>

          )}

        </div>


        {/* Tabs Semester */}

        <div className="flex flex-wrap gap-2 mb-6">

          {semesters.map((smt) => (

            <button

              key={smt}

              onClick={() => setSelectedSemester(smt)}

              className={`px-4 py-2 rounded-xl text-sm
font-medium shadow-sm transition

                ${selectedSemester === smt

                  ? 'bg-gradient-to-r from-cyan-500 to-blue-
600 text-white shadow-md'

                  : 'bg-white dark:bg-gray-800 text-gray-700
dark:text-gray-300 border hover:bg-gray-50
dark:hover:bg-gray-700'

                }`}

            >

              Semester {smt}

            </button>

          ))}

        </div>


        {/* Tabel */}

        <div className="overflow-x-auto">

          <table className="min-w-full border border-
gray-300 dark:border-gray-700 text-sm rounded-lg
overflow-hidden">

            <thead className="bg-gradient-to-r from-
cyan-500 to-blue-600 text-white">

              <tr>

                <th className="px-4 py-2 text-
left">Kode</th>

                <th className="px-4 py-2 text-
left">Nama</th>

                <th className="px-4 py-2 text-
center">SKS</th>

                <th className="px-4 py-2 text-
center">Jurusan</th>

                <th className="px-4 py-2 text-
center">Tipe</th>

                <th className="px-4 py-2 text-
center">Status</th>

                <th className="px-4 py-2 text-
center">Aksi</th>
```

```
            </tr>
          </thead>
          <tbody className="bg-white dark:bg-gray-900">
            {filteredMatkuls.length === 0 ? (
              <tr>
                <td colSpan="7" className="px-4 py-6 text-center text-gray-500 dark:text-gray-400">
                  Tidak ada mata kuliah untuk semester ini.
                </td>
              </tr>
            ) : (
              filteredMatkuls.map((matkul) => (
                <tr
                  key={matkul.id}
                  className={`border-t dark:border-gray-700 hover:bg-cyan-50 dark:hover:bg-gray-800 transition`}
                >
                  <td className="px-4 py-2">{matkul.kode}</td>
                  <td className="px-4 py-2">{matkul.nama}</td>
                  <td className="px-4 py-2 text-center">{matkul.sks}</td>
                  <td className="px-4 py-2 text-center">{matkul.jurusan}</td>
                  <td className="px-4 py-2 text-center">{matkul.tipe}</td>
                  <td className="px-4 py-2 text-center">{matkul.status}</td>
                  <td className="px-4 py-2 text-center space-x-2">
                    <Link
                      href={`/matkuls/${matkul.id}`}
                      className="text-blue-600 hover:text-blue-800"
                    >
                      <Eye className="inline w-5 h-5" />
                    </Link>
                    {hasPermission && (
                      <>
                        <Link
                          href={`/matkuls/${matkul.id}/edit`}
                          className="text-yellow-600 hover:text-yellow-800"
                        >
                          <Edit className="inline w-5 h-5" />
                        </Link>
                        <button
                          onClick={() => handleDelete(matkul.id)}
                          className="text-red-600 hover:text-red-800"
                        >
                          <Trash2 className="inline w-5 h-5" />
                        </button>
                      </>
                    )}
                  </td>
                </tr>
              ))
            )}
          </tbody>
        </table>
      </div>
    </div>
  </AppLayout>
 );
}
import React, { useEffect, useRef, useState } from "react";
import Message from "./Message.jsx";
import MessageInput from "./MessageInput.jsx";
import axios from "axios";
```

```
import { Link, usePage } from "@inertiajs/react";

const ChatBox = ({ rootUrl, recipient }) => {
  const { csrf_token } = usePage().props;
  const [messages, setMessages] = useState([]);
  const scroll = useRef();

  const fetchMessages = async () => {
    try {
      const { data } = await axios.get(`${rootUrl}/messages`, { params: { recipient_id: recipient.id } });
      setMessages(data);
      scroll.current?.scrollIntoView({ behavior: "smooth" });
    } catch (err) {
      console.error("Error fetching messages:", err.message);
    }
  };

  useEffect(() => {
    fetchMessages();
    return () => setMessages([]);
  }, [rootUrl, recipient.id]);

  const endChat = async () => {
    const keepHistory = window.confirm("Apakah Anda ingin menyimpan riwayat chat?");
    try {
      await axios.post(`${rootUrl}/chat/end`, {
        recipient_id: recipient.id,
        keep_history: keepHistory
      }, {
        headers: { 'X-CSRF-TOKEN': csrf_token }
      });
      alert(keepHistory ? "Chat berakhir. Riwayat disimpan." : "Chat berakhir. Riwayat dihapus.");
      window.location.href = "/home";
    } catch (err) {
      console.error("Error ending chat:", err.message);
    }
  };

  return (
    <div className="min-h-screen flex flex-col bg-gradient-to-br from-[#f7fafc] via-[#f0f9ff] to-[#e6fffa] dark:from-[#0f172a] dark:via-[#1e293b] dark:to-[#0f172a]">
      {/* Header */}
      <div className="p-4 border-b bg-white/80 dark:bg-gray-800/80 backdrop-blur rounded-b-xl shadow flex justify-between items-center">
        <Link href="/home" className="text-cyan-600 hover:text-blue-600 transition">← Kembali</Link>
        <h2 className="font-bold bg-gradient-to-r from-cyan-600 to-blue-700 bg-clip-text text-transparent">{recipient.name}</h2>
        <button onClick={endChat} className="px-4 py-1 rounded-lg bg-gradient-to-r from-red-500 to-pink-600 text-white text-sm shadow hover:opacity-90 transition">
          End Chat
        </button>
      </div>

      {/* Chat Messages */}
      <div className="flex-1 overflow-y-auto p-6 space-y-3">
        {messages.map((message, index) => (
          <Message key={index} message={message} userId={usePage().props.auth.user.id} />
        ))}
        <span ref={scroll}></span>
      </div>
```

```
        {/* Input Chat */}
        <div className="p-4 border-t bg-white/80 dark:bg-gray-800/80 backdrop-blur rounded-t-xl shadow">
            <MessageInput rootUrl={rootUrl} recipientId={recipient.id} />
        </div>
      </div>
    );
};

export default ChatBox;

import React from "react";
import { Link, usePage } from "@inertiajs/react";
import AppLayout from '@/layouts/app-layout';

const ChatUsers = ({ users }) => {
    const { auth } = usePage().props;
    const isMahasiswa = auth.roles.includes("mahasiswa");

const filteredUsers = isMahasiswa
  ? users.filter(user => user.roles.includes("dosen") || user.roles.includes("dosenPA"))
  : users;

    return (
        <AppLayout>
            <div className="max-w-lg mx-auto p-6 bg-white/80 dark:bg-gray-800/80 rounded-2xl shadow-lg backdrop-blur">
                <h2 className="text-xl font-bold text-center mb-6 bg-gradient-to-r from-cyan-600 to-blue-700 bg-clip-text text-transparent">
                    Pilih Pengguna untuk Chat
                </h2>
                {filteredUsers.length > 0 ? (
                    <ul className="space-y-3">
                        {filteredUsers.map((user) => (
                            <li key={user.id} className="p-4 border rounded-xl hover:bg-gradient-to-r from-cyan-50 to-blue-50 dark:hover:from-gray-700 dark:hover:to-gray-800 transition">
                                <Link href={`/chat/${user.id}`} className="font-medium text-cyan-700 dark:text-cyan-300 hover:text-blue-600">
                                    {user.name} <span className="text-sm text-gray-500">({user.roles.join(", ")})</span>
                                </Link>
                            </li>
                        ))}
                    </ul>
                ) : (
                    <p className="text-gray-500 text-center">Tidak ada Dosen yang tersedia.</p>
                )}
            </div>
        </AppLayout>
    );
};

export default ChatUsers;
```