

LAPORAN PRAKTIKUM PEMROGRAMAN WEB LANJUT



Oleh :

VINA FAIZATUS SOFITA

NIM. 362458302095

2D

Dosen Pengampu : Furiansyah Dipraja, S.T., M.Kom

**PROGRAM STUDI DIPLOMA IV
TEKNOLOGI REKAYASA PERANGKAT LUNAK
POLITEKNIK NEGERI BANYUWANGI**

2025

1. Tujuan

- Membuat database baru bernama `integrasi_database` pada proyek Laravel `belajar_laravel`.
- Menghubungkan database tersebut ke Laravel melalui file konfigurasi `.env`.
- Membuat tabel mahasiswa menggunakan migration.
- Mengelola data mahasiswa dengan `MahasiswaController` menggunakan Query Builder (`insert` dan `get`).

2. Praktikum

2.2 Membuat database dari VS Code

Sebelum membuat migration, kita perlu menyiapkan database yang akan digunakan oleh Laravel

Langkah-langkah:

1. buat project Laravel baru dengan menggunakan query

`composer create-project laravel/laravel integrasi_database`

lalu aktifkan MySQL dan buat database

- buka XAMPP
- jalankan Apache dan MySQL

2. Untuk masuk ke MariaDB, masuk ke folder `C:\xampp\mysql\bin>mysql -u root -p` dan diminta memasukkan password, tetapi disini kita tidak perlu memasukkan password

```
C:\xampp\mysql\bin>mysql -u root -p
Enter password:
```

Sudah masuk ke database

```
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| integrasi_database |
| mysql          |
| performance_schema |
| phpmyadmin     |
| test           |
+-----+
6 rows in set (0.072 sec)
```

3. Setelah database siap, buka file .env di root proyek laravel dan ubah konfigurasi database seperti ini

Ubah pada bagian ini dengan **DB_DATABASE=integrasi_database**

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=integrasi_database
DB_USERNAME=root
DB_PASSWORD=
```

(Sesuaikan DB_USERNAME dan DB_PASSWORD dengan pengaturan MySQL lokal)

4. Simpan file .env. laravel sekarang sudah terhubung dengan database integrasi_database

2.2 Membuat Migration

Migration digunakan untuk membuat struktur tabel didatabase.

Jalankan perintah:

```
php artisan make:migration create_mahasiswas_table
```

Buka file migration yang dibuat di database/migrations/ dan ubah isinya menjadi:

```
<?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('mahasiswas', function (Blueprint $table) {
            $table->id();
            $table->string('nama');
            $table->string('email');
            $table->string('nim')->unique();
            $table->string('jurusan');
            $table->timestamps();
        });
    }

    public function down(): void
```

```
{  
    Schema::dropIfExists('mahasiswas');  
}  
};
```

Lalu jalankan:

php artisan migrate

2.3 Membuat Seeder

Perintah :

php artisan make:seeder MahasiswaSeeder

Isi file **database/seeder/MahasiswaSeeder.php**

```
<?php  
  
namespace Database\Seeders;  
  
use Illuminate\Database\Seeder;  
use Illuminate\Support\Facades\DB;  
  
class MahasiswaSeeder extends Seeder  
{  
    public function run()  
    {  
        DB::table('mahasiswas')->insert([  
            [  
                'nama' => 'Vina Faizatus',  
                'email' => 'vina@gmail.com',  
                'nim' => '12345678',  
                'jurusan' => 'Informatika',  
                'created_at' => now(),  
                'updated_at' => now(),  
            ],  
            [  
                'nama' => 'Dian Restu',  
                'email' => 'dian@gmail.com',  
                'nim' => '87654321',  
                'jurusan' => 'Sistem Informasi',  
                'created_at' => now(),  
                'updated_at' => now(),  
            ],  
            [  
                'nama' => 'Cheryl Aurel',  
                'email' => 'cheryl@gmail.com',  
                'nim' => '121212',  
                'jurusan' => 'Bisnis Digital',  
            ]  
        ]  
    )  
}
```

```

        'created_at' => now(),
        'updated_at' => now(),
    ],
    ];
}
}

```

Tambahkan ke DatabaseSeeder.php:

```
$this->call(MahasiswaSeeder::class);
```

Kemudian jalankan:

```
php artisan migrate:fresh --seed
```

Perintah ini akan menghapus tabel lama, membuat ulang, lalu mengisi data otomatis dari Seeder

2.4 Membuat Controller

Controller digunakan untuk emnagtur logika aplikasi.

Buat controller baru:

```
php artisan make:controller MahasiswaController
```

buka file **app/Http/Controllers/MahasiswaController.php** dan isi dengan code berikut:

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

class MahasiswaController extends Controller
{
    public function insertData()
    {
        DB::table('mahasiswas')->insert([
            'nama' => 'Faizatus',
            'email' => 'faiiza@gmail.com',
            'nim' => '11223345',
            'jurusan' => 'Teknik Imformatika',
            'created_at' => now(),
            'updated_at' => now(),
        ]);
    }
}

```

```

        return "Data berhasil ditambahkan!";
    }

    public function getData()
    {
        $mahasiswas = DB::table('mahasiswas')->get();
        return view('mahasiswa.index', compact('mahasiswas'));
    }
}

```

- InsertData() digunakan untuk menambahkan data baru ke tabel.
- getData digunakan untuk mengambil semua data mahasiswa dari database

2.5 Membuat Model

Model digunakan agar tabell mahasiswas

Perintah :

php artisan make:model Mahasiswa

isi file **app/Models/Mahasiswa.php**

```

namespace App\Models;

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

class Mahasiswa extends Model
{
    use HasFactory;

    protected $table = 'mahasiswas';

    protected $fillable = ['nama', 'email', 'telepon', 'alamat'];
}

```

2.6 Menambahkan Routing dengan Query Builder

Tambahkan di **routes/web.php**:

```

<?php

//use Illuminate\Support\Facades\Route;
use Illuminate\Support\Facades\DB;
use App\Http\Controllers\MahasiswaController;

// Route::resource('mahasiswa', MahasiswaController::class);

```

```
// Route::get('/', function () {
//     return view('welcome');
// });

// // Insert data mahasiswa
// Route::get('/insert', function () {
//     DB::table('mahasiswa')->insert([
//         'nama' => 'Vina Faizatus',
//         'nim' => '25252525',
//         'jurusan' => 'Teknik Komputer',
//         'created_at' => now(),
//         'updated_at' => now(),
//     ]);

//     return "Data berhasil ditambahkan!";
// });

// // Ambil semua data mahasiswa
// Route::get('/mahasiswa', function () {
//     $mahasiswa = DB::table('mahasiswa')->get();
//     return response()->json($mahasiswa);
// });

Route::get('/insert', [MahasiswaController::class, 'insertData']);
Route::get('/mahasiswas', [MahasiswaController::class, 'getData']);
```

2.5 Membuat Tampilan (View)

Buat folder baru:

resources/views/mahasiswa

lalu buat file **indeks.blade.php** dengan isi:

akses pada **resources/views/mahasiswa/indeks.blade.php**

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

@section('content')
    <h2 class="text-2xl font-bold mb-4">Daftar Mahasiswa</h2>

    @if(session('success'))
        <div class="bg-green-100 text-green-700 p-3 mb-4 rounded">
            {{ session('success') }}
        </div>
    @endif

    <table class="table-auto border-collapse border w-full">
        <thead>
```

```

<th class="border px-4 py-2">ID</th>
  <th class="border px-4 py-2">Nama</th>
  <th class="border px-4 py-2">Email</th>
  <th class="border px-4 py-2">Nim</th>
  <th class="border px-4 py-2">Jurusan</th>
</tr>
</thead>
<tbody>
  @foreach($mahasiswas as $m)
    <tr>
      <td class="border px-4 py-2">{{ $m->id }}</td>
      <td class="border px-4 py-2">{{ $m->nama }}</td>
      <td class="border px-4 py-2">{{ $m->email }}</td>
      <td class="border px-4 py-2">{{ $m->nim }}</td>
      <td class="border px-4 py-2">{{ $m->jurusan }}</td>
    </tr>
  @endforeach
</tbody>
</table>
@endsection

```

2.6 Periksa tabel mahasiswa

Setelah menambahkan email

```

MariaDB [integrasi_database]> desc mahasiswa;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id     | bigint(20) unsigned | NO   | PRI | NULL    | auto_increment |
| nama   | varchar(255)      | NO   |     | NULL    |                |
| email  | varchar(255)      | NO   |     | NULL    |                |
| nim    | varchar(255)      | NO   | UNI | NULL    |                |
| jurusan | varchar(255)      | NO   |     | NULL    |                |
| created_at | timestamp        | YES  |     | NULL    |                |
| updated_at | timestamp        | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.049 sec)

```

```

MariaDB [integrasi_database]> select*from mahasiswa;
+----+-----+-----+-----+-----+-----+-----+
| id | nama          | email          | nim    | jurusan    | created_at      | updated_at      |
+----+-----+-----+-----+-----+-----+-----+
| 1  | Vina Faizatus | vina@gmail.com | 12345678 | Informatika | 2025-10-03 03:58:13 | 2025-10-03 03:58:13 |
| 2  | Dian Restu    | dian@gmail.com | 87654321 | Sistem Informasi | 2025-10-03 03:58:13 | 2025-10-03 03:58:13 |
| 3  | Cheryl Aurel  | dian@gmail.com | 121212  | Bisnis Digital | 2025-10-03 03:58:13 | 2025-10-03 03:58:13 |
+----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.001 sec)

```

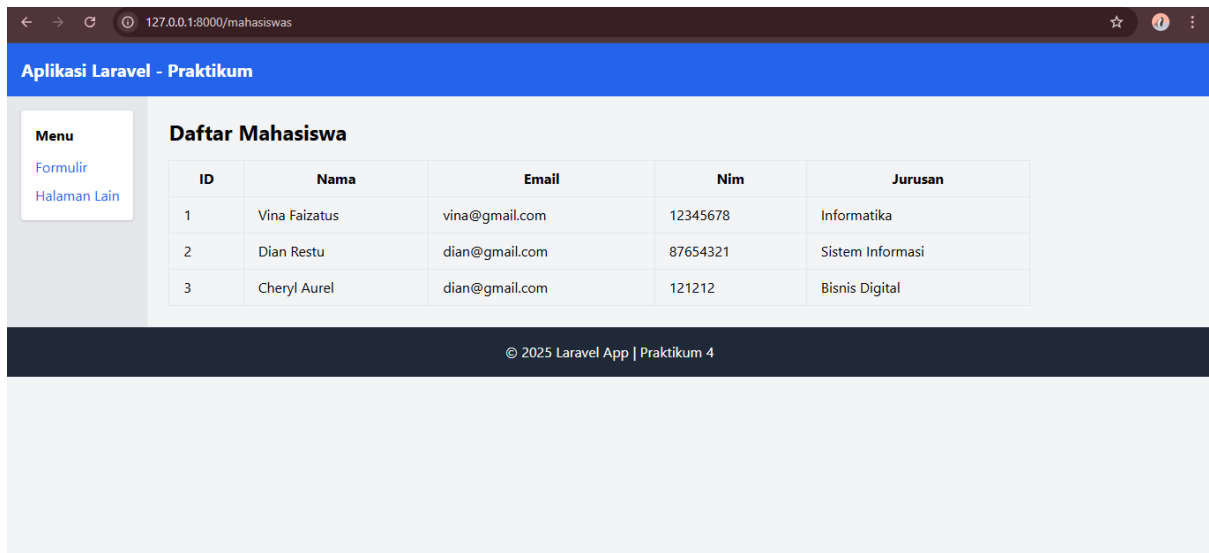
2.7 Uji Coba Aplikasi

1. jalankan server Laravel:

`php artisan serve`

2. Buka browser dan akses:

- <http://localhost:8000/mahasiswa> (tampil semua data)



3. Kesimpulan

- **Migration** mempermudah proses pembuatan struktur tabel tanpa harus mengetik SQL secara manual.
- **Controller** mengatur alur logika aplikasi dan dapat berinteraksi langsung dengan database melalui Query Builder.
- **View (Blade Template)** menampilkan hasil query dalam format yang lebih mudah dibaca pengguna.
- Dengan kombinasi **Migration + Controller + Query Builder + View**, Laravel dapat mengelola data secara efisien, dinamis, dan terstruktur.