LAPORAN PRAKTIKUM

PERANCANGAN DAN PEMROGRAMAN WEB

MODUL 13

(MIGRATION)

Oleh:

Farhan Kurniawan 2311104073

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

DIREKTORAT KAMPUS PURWOKERTO

UNIVERSITAS TELKOM

2025

BAB I

PENDAHULUAN

A. Dasar Teori

Migration pada framework Laravel merupakan fitur yang digunakan untuk mengelola struktur basis data secara terprogram melalui kode. Dengan migration, pengembang dapat membuat, mengubah, dan menghapus tabel maupun kolom pada database tanpa harus menuliskan perintah SQL secara langsung. Setiap perubahan struktur database disimpan dalam bentuk file migration sehingga memudahkan proses pengelolaan database secara sistematis, konsisten, dan terdokumentasi dengan baik.

Selain itu, migration berperan sebagai *version control* untuk database karena setiap perubahan dicatat berdasarkan urutan waktu (timestamp). Hal ini memungkinkan pengembang melakukan rollback atau mengembalikan struktur database ke kondisi sebelumnya apabila terjadi kesalahan. Dengan kemampuan tersebut, migration sangat membantu dalam pengembangan aplikasi yang dinamis, terutama ketika terjadi perubahan kebutuhan sistem, tanpa harus menghapus data yang sudah ada di dalam database.

B. Tujuan

Tujuan dari praktikum ini adalah untuk memahami konsep dan fungsi migration pada framework Laravel serta mampu menerapkannya dalam pengelolaan struktur basis data aplikasi. Melalui praktikum ini, mahasiswa diharapkan dapat membuat, memodifikasi, dan mengelola tabel database secara terstruktur menggunakan migration, sehingga proses pengembangan aplikasi menjadi lebih efisien, terkontrol, dan mudah dikembangkan sesuai dengan kebutuhan sistem.

BAB II

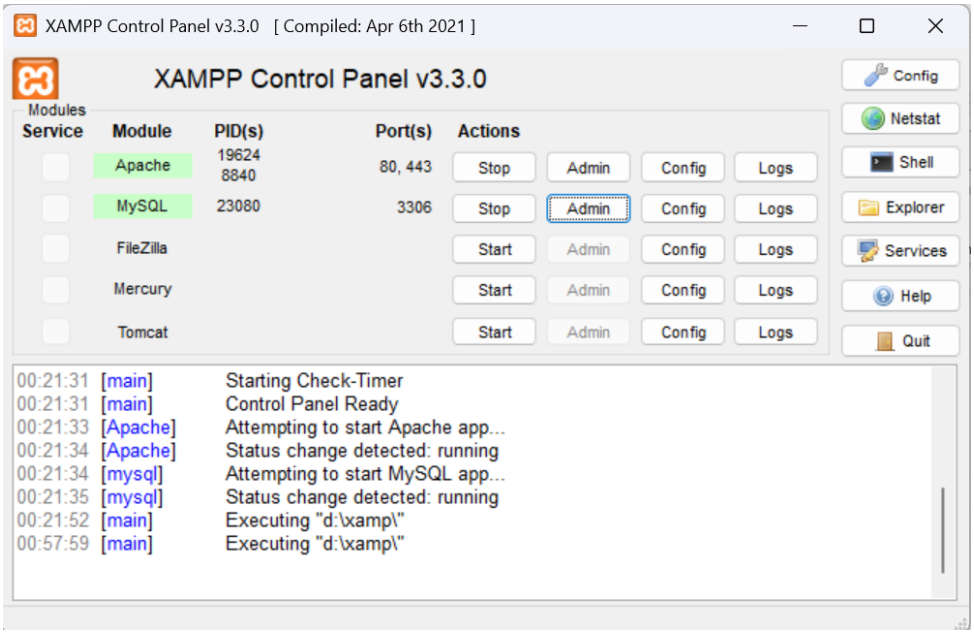
HASIL

A. GUIDED

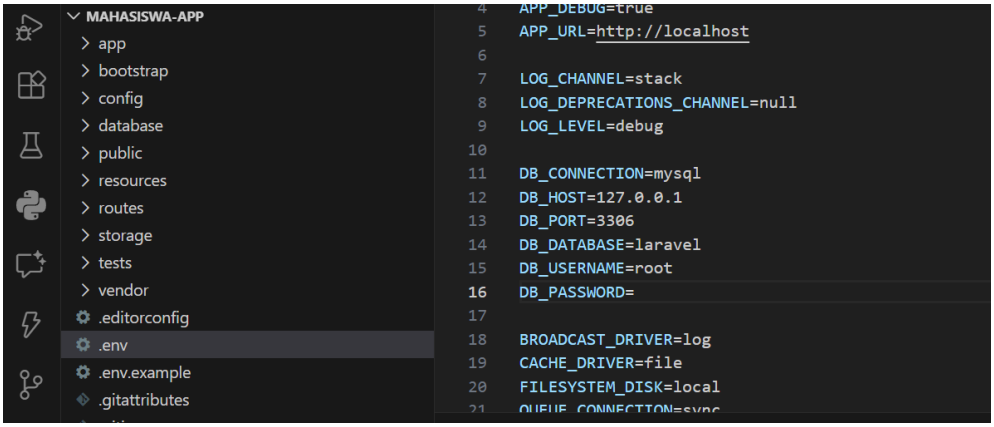
1. Migration

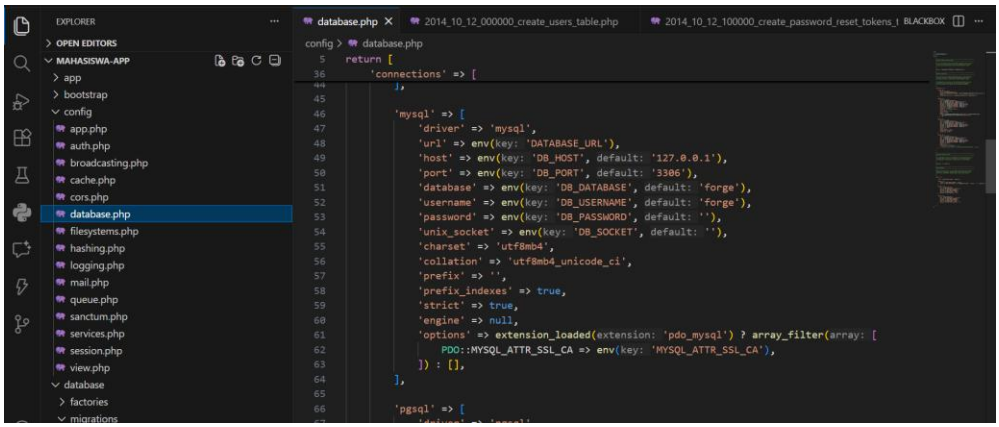
a. Konfigurasi Database

Sebelum menjalankan migration, layanan Apache dan MySQL harus diaktifkan melalui XAMPP.



Selanjutnya, Laravel memerlukan pengaturan koneksi database yang dilakukan pada dua berkas konfigurasi, yaitu file .env sebagai konfigurasi lokal dan file config/database.php sebagai konfigurasi global.

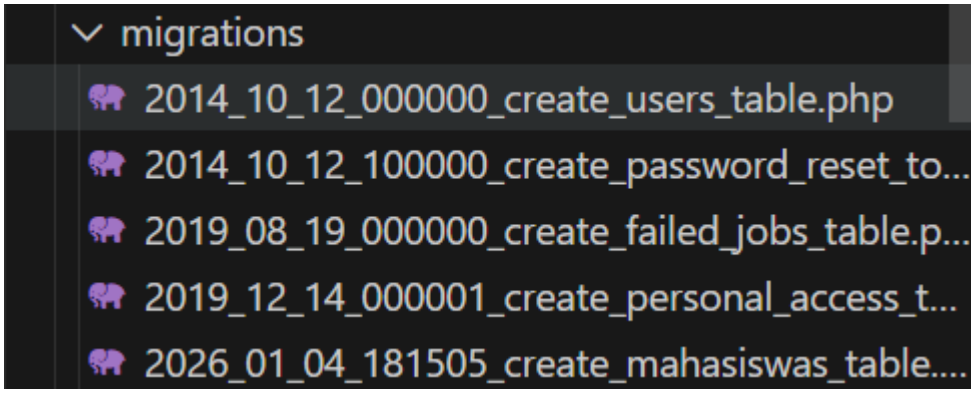




Pengaturan pada file .env digunakan untuk menentukan nama database, username, dan password yang akan digunakan oleh aplikasi. Sementara itu, file database.php berfungsi untuk memastikan pengaturan database sesuai dengan konfigurasi yang ditetapkan secara global pada framework Laravel.

b. **File Migration Bawaan Laravel**

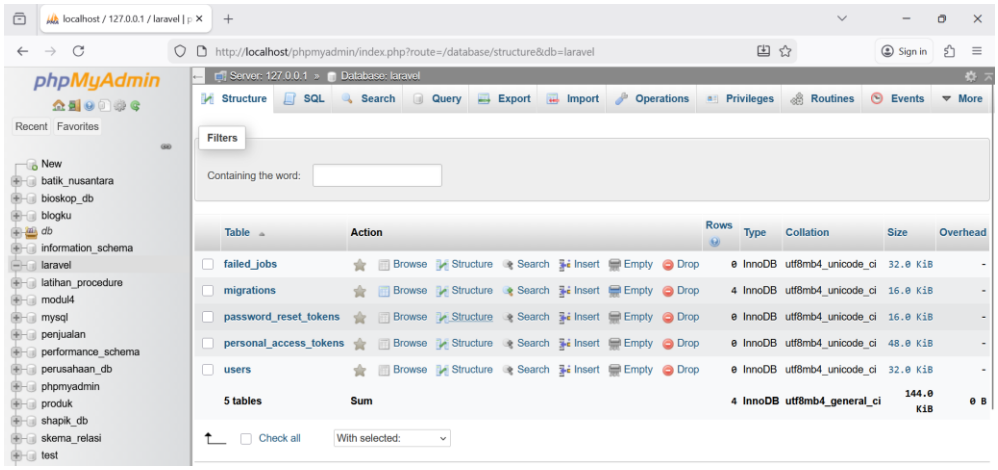
Pada beberapa versi Laravel, telah tersedia file migration bawaan yang terletak pada direktori database/migrations.



Setiap file migration digunakan untuk membentuk satu tabel pada database. Oleh karena itu, beberapa file migration bawaan akan menghasilkan beberapa tabel secara otomatis. Nama file migration diawali dengan timestamp yang menunjukkan waktu pembuatan file tersebut. Urutan timestamp ini menentukan urutan eksekusi migration ketika perintah dijalankan. Untuk menjalankan migration bawaan Laravel, digunakan perintah:

```
PS D:\xamp\htdocs\PPW\Pertemuan 13\Guided\mahasiswa-app> php artisan migrate
>>

  INFO  Preparing database.

  Creating migration table ..... 14ms DON
E

  INFO  Running migrations.

  2014_10_12_000000_create_users_table ..... 29ms DON
E
  2014_10_12_100000_create_password_reset_tokens_table ..... 10ms DON
ommit Message   Ln 1, Col 1   Spaces: 4   UTF-8   LF   { } PHP   Go Live   BLACKBOXAI: Open Chat   8.1   Prettier
```

Setelah perintah dijalankan, hasil pembuatan tabel dapat diperiksa melalui phpMyAdmin.



c. **Migration Rollback**

Laravel menyediakan fitur rollback untuk membatalkan perubahan struktur database yang telah dijalankan. Migration dijalankan berdasarkan urutan timestamp, sehingga rollback dapat digunakan untuk kembali ke kondisi database sebelumnya. Untuk melihat status migration yang telah dijalankan, digunakan perintah:

```
PS D:\xamp\htdocs\PPW\Pertemuan 13\Guided\mahasiswa-app> php artisan migrate:status

  Migration name ..... Batch / Status
  2014_10_12_000000_create_users_table ..... [1] Ran
  2014_10_12_100000_create_password_reset_tokens_table ..... [1] Ran
  2019_08_19_000000_create_failed_jobs_table ..... [1] Ran
  2019_12_14_000001_create_personal_access_tokens_table ..... [1] Ran
  2026_01_04_181505_create_mahasiswas_table ..... [2] Ran
```

Sedangkan untuk membatalkan satu migration terakhir, digunakan perintah:

```
PS D:\xamp\htdocs\PPW\Pertemuan 13\Guided\mahasiswa-app> php artisan migrate:rollback --step=1

  INFO  Rolling back migrations.

  2026_01_04_181505_create_mahasiswas_table ..... 13ms DONE
```

Open folder in new window (ctrl + click)

Perintah tersebut akan menghapus perubahan yang dihasilkan oleh satu file migration terakhir.

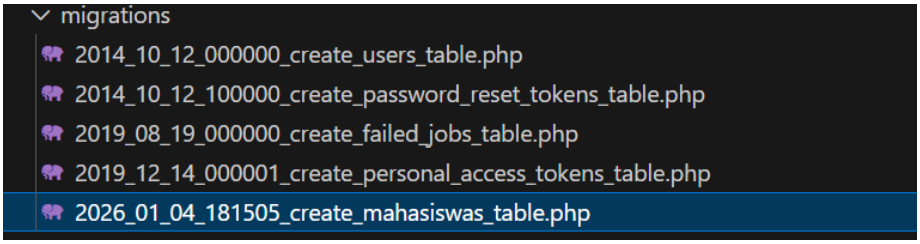
d. **Pembuatan Migration Baru**

File migration dapat dibuat secara manual, namun cara yang lebih efisien adalah menggunakan perintah Artisan. Format penamaan migration sebaiknya mengikuti pola <aksi>_<nama_tabel>_table agar mudah dipahami.

Contoh pembuatan migration tabel mahasiswas:

```
PS D:\xamp\htdocs\PPW\Pertemuan 13\Guided\mahasiswa-app> php artisan make:migration create_mahasiswas_table --create=mahasiswas
>>

  INFO  Migration [D:\xamp\htdocs\PPW\Pertemuan 13\Guided\mahasiswa-app\database\migrations/2026_01_04_181505_create_mahasiswas_table.php]
created successfully.
```

Perintah tersebut akan menghasilkan file migration baru pada folder database/migrations dengan nama yang disertai timestamp.



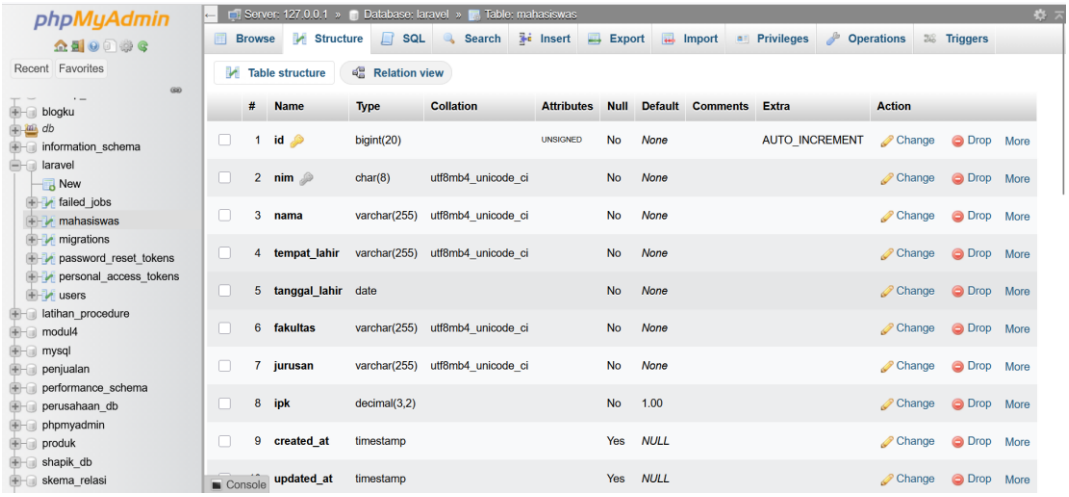
Setelah file migration dibuat, struktur tabel didefinisikan pada method up() menggunakan Schema Builder. Method down() digunakan untuk menghapus tabel apabila dilakukan rollback.

Source Code

```
<?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()
{
    Schema::create('mahasiswas', function (Blueprint $table) {
        $table->id();
        $table->char('nim',8)->unique();
        $table->string('nama');
        $table->string('tempat_lahir');
```

```
        $table->date('tanggal_lahir');
        $table->string('fakultas');
        $table->string('jurusan');
        $table->decimal('ipk',3,2)->default(1.00);
        $table->timestamps();
    });
}

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

Struktur Tabel



e. **Modifikasi dan Rollback Migration**

Struktur tabel yang telah dibuat dapat disesuaikan kembali dengan kebutuhan aplikasi. Perubahan pada file migration mengharuskan proses rollback terlebih dahulu sebelum menjalankan migration ulang agar perubahan dapat diterapkan ke database.

Source code yang di ubah

```
        public function up(){
            Schema::create('mahasiswas', function (Blueprint $table) {
                $table->id();
                $table->char('nim',8)->unique();
                $table->string('nama');
                $table->string('tempat_lahir');
                $table->date('tanggal_lahir');
                $table->string('fakultas');
                $table->string('jurusan');
                $table->decimal('ipk',3,2)default(1.00);
                $table->timestamps();
            });
    }
```

B. UNGUIDED

Soal:

1. Buatlah file migration untuk database yang akan digunakan pada tugas besar di kelas teori.
2. Buatlah fungsi-fungsi berikut untuk menyelesaikan tugas besar di matakuliah teori:
    a. Buat fungsi untuk insert data menggunakan raw SQL Queries
    b. Buat fungsi untuk insert data menggunakan Query Builder
    c. Buat fungsi untuk insert data menggunakan Eloquent ORM

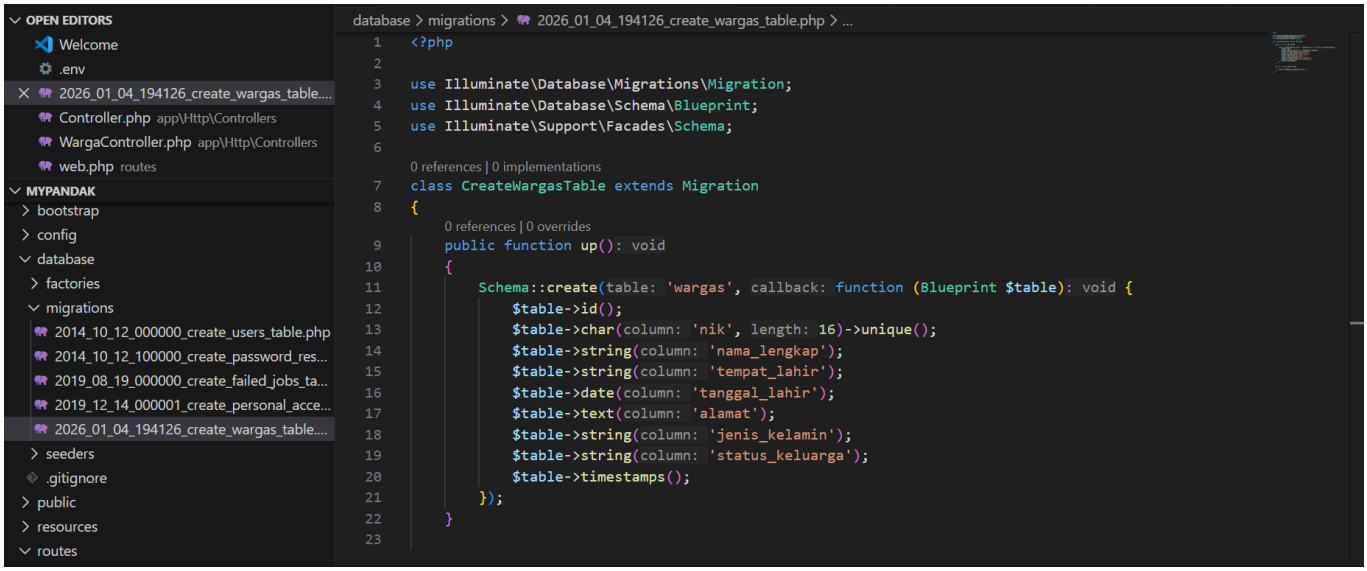
Jawaban:

1. Berikut:

    Buat file migration:

```
PS D:\xamp\htdocs\PPW\Pertemuan 13\Unguided\mypandak> php artisan make:migration create_wargas_table --create=wargas
>>

  INFO  Migration [D:\xamp\htdocs\PPW\Pertemuan 13\Unguided\mypandak\database\migrations/2026_01_04_191747_create_wargas_
table.php] created successfully.
```

    Edit File Migration



Kemudian

```
PS D:\xamp\htdocs\PPW\Pertemuan 13\Unguided\mypandak> php artisan migrate
>>

  INFO  Preparing database.

  Creating migration table ..................................... 18ms DONE

  INFO  Running migrations.

  2014_10_12_000000_create_users_table ........................ 26ms DONE
  2014_10_12_100000_create_password_reset_tokens_table ........ 9ms DONE
  2019_08_19_000000_create_failed_jobs_table ................... 28ms DONE
  2019_12_14_000001_create_personal_access_tokens_table ....... 33ms DONE
  2026_01_04_194126_create_wargas_table ........................ 24ms DONE
```

2. Membuat Controller

A. INSERT DATA MENGGUNAKAN RAW SQL

edit WargaController.php

```
use Illuminate\Support\Facades\DB;


public function insertRaw()
{
   $result = DB::insert("
      INSERT INTO wargas
      (nik, nama_lengkap, tempat_lahir, tanggal_lahir, alamat, jenis_kelamin,
status_keluarga)
      VALUES
      ('3305123456780001',
       'Budi Santoso',
       'Purwokerto',
       '2001-05-12',
       'Jl. Merdeka No. 10',
       'Laki-laki',
       'Kepala Keluarga')
   ");


   dump($result);
```

```
}
```

Tambahkan Route

```
use App\Http\Controllers\WargaController;

Route::get('/insert-raw', [WargaController::class, 'insertRaw']);
```

B. INSERT DATA MENGGUNAKAN QUERY BUILDER

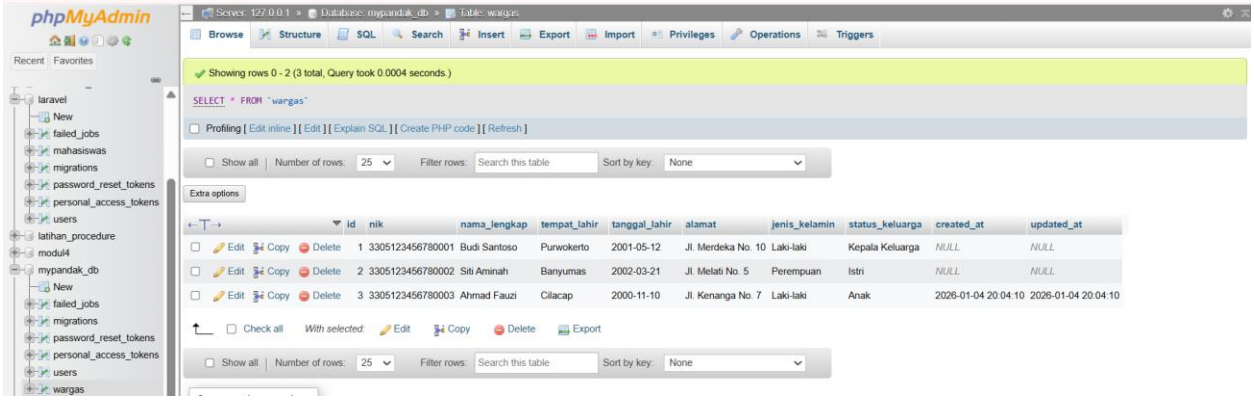
```
public function insertQB()
{
    $result = DB::table('wargas')->insert([
        'nik' => '3305123456780002',
        'nama_lengkap' => 'Siti Aminah',
        'tempat_lahir' => 'Banyumas',
        'tanggal_lahir' => '2002-03-21',
        'alamat' => 'Jl. Melati No. 5',
        'jenis_kelamin' => 'Perempuan',
        'status_keluarga' => 'Istri',
    ]);

    dump($result);
}
```

C. INSERT DATA MENGGUNAKAN ELOQUENT ORM

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Warga extends Model
{
    protected $table = 'wargas';
```

```
    protected $fillable = [
        'nik',
        'nama_lengkap',
        'tempat_lahir',
        'tanggal_lahir',
        'alamat',
        'jenis_kelamin',
        'status_keluarga'
    ];
}
```

Maka Hasil setelah di migrasi:



BAB III

PENUTUP

A. Kesimpulan

Modul 12 membahas penggunaan Migration pada Laravel sebagai cara untuk mengelola struktur database secara teratur melalui kode program. Dengan migration, pembuatan, perubahan, dan penghapusan tabel dapat dilakukan dengan lebih mudah dan aman tanpa harus mengubah database secara langsung. Praktikum ini juga menunjukkan bahwa Laravel menyediakan tiga cara untuk mengelola data, yaitu Raw SQL, Query Builder, dan Eloquent ORM. Ketiga metode tersebut dapat digunakan untuk melakukan proses input data ke dalam database sesuai dengan kebutuhan aplikasi. Melalui penerapan migration dan berbagai metode pengolahan data tersebut, pengelolaan database menjadi lebih rapi, terkontrol, dan mudah dikembangkan dalam pembuatan aplikasi berbasis Laravel.