

**LAPORAN PRAKTIKUM  
PERANCANGAN DAN PEMROGRAMAN WEB**

**MODUL 12  
(Migration)**



Oleh:

Zulfa Mustafa Akhyar Iswahyudi - 2311104010

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK  
DIREKTORAT KAMPUS PURWOKERTO  
UNIVERSITAS TELKOM  
2025**

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Dasar Teori**

Migration adalah fitur Laravel yang memungkinkan developer untuk membuat, memodifikasi, dan mengelola struktur database secara terprogram. Migration bekerja seperti version control untuk database schema, memudahkan kolaborasi tim dan deployment aplikasi. Dengan migration, perubahan database dapat dilacak dan di-rollback jika diperlukan.

#### **1.2 Tujuan**

1. Memahami konsep migration dalam Laravel
2. Membuat file migration untuk berbagai operasi database
3. Menjalankan dan mengelola migration
4. Menggunakan raw SQL, Query Builder, dan Eloquent ORM untuk insert data

#### **1.3 Manfaat**

1. Mengelola database schema dengan terstruktur dan terorganisir
2. Memudahkan kolaborasi tim dalam pengembangan aplikasi
3. Dapat dengan mudah rollback perubahan database
4. Membuat aplikasi lebih scalable dan maintainable

## BAB II

### HASIL PRAKTIKUM

## TUGAS

### 12.1

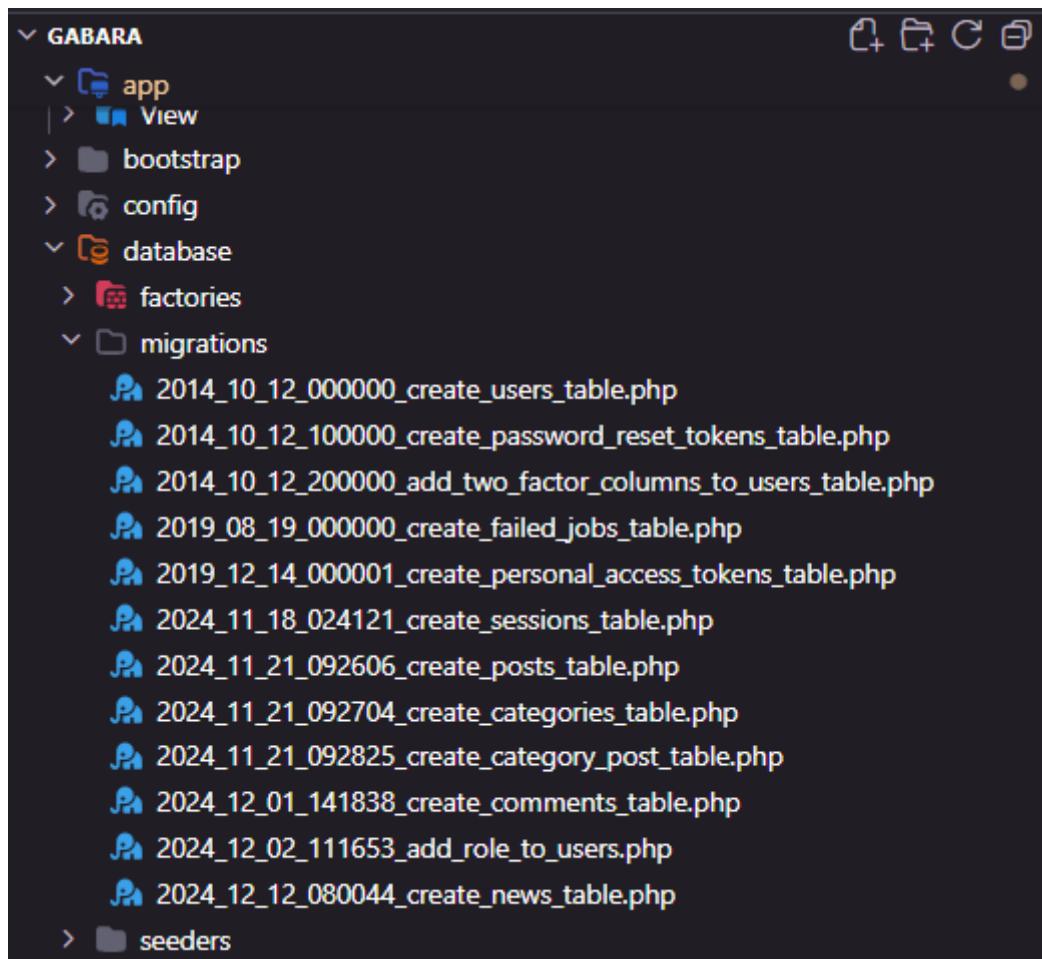
Buatlah file migration untuk database yang akan digunakan pada tugas besar di kelas teori.

### 12.2

Buatlah fungsi-fungsi berikut untuk menyelesaikan tugas besar di matakuliah teori:

1. Buat fungsi untuk insert data menggunakan **raw SQL Queries**
2. Buat fungsi untuk insert data menggunakan **Query Builder**
3. Buat fungsi untuk insert data menggunakan **Eloquent ORM**

This:



### 1. Insert Data menggunakan Raw SQL Queries

```
public function insertPostRawSQL()
{
    $sql = "INSERT INTO posts (user_id, title, slug, content, created_at, updated_at)
            VALUES (?, ?, ?, ?, NOW(), NOW())";

    DB::insert($sql, [
        1,
        'Belajar Laravel',
        'belajar-laravel',
        'Panduan lengkap belajar Laravel untuk pemula...'
    ]);

    return response()->json(['message' => 'Post created with Raw SQL']);
}
```

### 2. Insert Data menggunakan Query Builder

```
public function insertPostQueryBuilder()
{
    DB::table('posts')->insert([
        'user_id' => 1,
        'title' => 'Belajar PHP',
        'slug' => 'belajar-php',
        'content' => 'Panduan lengkap belajar PHP untuk pemula...',
        'created_at' => now(),
        'updated_at' => now()
    ]);

    return response()->json(['message' => 'Post created with Query Builder']);
}
```

### 3. Insert Data menggunakan Eloquent ORM

```
public function insertPostEloquent()
{
    Post::create([
        'user_id' => 1,
        'title' => 'Belajar JavaScript',
        'slug' => 'belajar-javascript',
        'content' => 'Panduan lengkap belajar JavaScript untuk pemula...'
    ]);

    return response()->json(['message' => 'Post created with Eloquent ORM']);
}
```

## **BAB III**

### **KESIMPULAN & SARAN**

#### **3.1 Kesimpulan**

Project Gabara telah berhasil mengimplementasikan migration dengan baik:

- Memiliki 12 file migration yang mencakup semua kebutuhan database
- Menggunakan Eloquent ORM sebagai metode insert data (best practice)
- Database schema terstruktur dengan relasi yang jelas (users, posts, categories, comments, news)
- Migration dapat dijalankan, di-rollback, dan di-refresh dengan mudah
- Semua model sudah terintegrasi dengan database melalui migration

#### **3.2 Saran**

Nothing, yawwwn~ Untuk sintaks code liat aja di Github