

**LAPORAN PRAKTIKUM**  
**PERANCANGAN DAN PEMROGRAMAN WEB**

**MODUL XII**

**(Laravel 2: Migration)**



Oleh:

(Faishal Arif Setiawan) (2311104066)

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK**

**DIREKTORAT KAMPUS PURWOKERTO**

**UNIVERSITAS TELKOM**

**2025**

## **BAB I**

### **PENDAHULUAN**

#### **A. Dasar Teori**

Migration pada Laravel merupakan sebuah fitur yang dapat membantu programmer mengelola database secara efisien dengan menggunakan kode. Migration membantu programmer dalam membuat (create), mengubah (edit), dan menghapus (delete) struktur tabel dan kolom pada database milik kita dengan cepat dan mudah. Dengan Migration kita dapat melakukan perubahan pada struktur database tanpa harus menghapus data yang ada. Migration membuat pengembangan aplikasi menjadi lebih mudah terutama dalam mengatasi perubahan kebutuhan yang muncul selama proses pengembangan aplikasi kita. Migration memungkinkan kita untuk mengelola perubahan pada struktur database dan menyimpannya pada file yang dapat dijalankan dengan mudah di lingkungan lain. Dalam istilah yang lebih teknis, migration disebut sebagai version control untuk database. Selain dipakai membuat tabel, kita juga bisa membuat step-by-step perubahan struktur tabel.

#### **B. Tujuan**

1. Memahami fungsi dan cara kerja model pada framework laravel
  2. Mampu menerapkan migration pada aplikasi

## **BAB II**

### **HASIL**

#### **A. GUIDED (Praktikum Terbimbing)**

File Konfigurasi:

```
23 DB_CONNECTION=mysql
24 DB_HOST=127.0.0.1
25 DB_PORT=3306
26 DB_DATABASE=db_latihan_praktikum
27 DB_USERNAME=root
28 DB_PASSWORD=
```

konfigurasi kedua, yaitu di config\database.php:

```
'mysql' => [
    'driver' => 'mysql',
    'url' => env(key: 'DB_URL'),
    'host' => env(key: 'DB_HOST', default: '127.0.0.1'),
    'port' => env(key: 'DB_PORT', default: '3306'),
    'database' => env(key: 'DB_DATABASE', default: 'db_latihan_praktikum'),
    'username' => env(key: 'DB_USERNAME', default: 'root'),
    'password' => env(key: 'DB_PASSWORD', default: ''),
    'unix_socket' => env(key: 'DB_SOCKET', default: ''),
    'charset' => env(key: 'DB_CHARSET', default: 'utf8mb4'),
    'collation' => env(key: 'DB_COLLATION', default: 'utf8mb4_unicode_ci'),
    'prefix' => '',
    'prefix_indexes' => true,
    'strict' => true,
```

Membuat Table Mahasiswa:

```
1 <?php
2 Use Illuminate\Database\Migrations\Migration;
3 use Illuminate\Database\Schema\Blueprint;
4 use Illuminate\Support\Facades\Schema;
5
6 class CreateMahasiswaTable extends Migration
7 {
8
9     public function up(): void{
10        Schema::create('mahasiswa', callback: function (Blueprint $table): void {
11            $table->id();
12            $table->char(column: 'nim', length: 8)->unique();
13            $table->string(column: 'nama');
14            $table->string(column: 'tempat_lahir');
15            $table->date(column: 'tanggal_lahir');
16            $table->string(column: 'fakultas');
17            $table->string(column: 'jurusan');
18            $table->decimal(column: 'ipk', total: 3, places: 2)->default(value: 1.00);
19        });
20    }
21 }
```

## Struktur Table:

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	<b>id</b> 🛡	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2	<b>nim</b>	char(8)	utf8mb4_unicode_ci		No	None			Change  Drop  More
<input type="checkbox"/>	3	<b>nama</b>	varchar(255)	utf8mb4_unicode_ci		No	None			Change  Drop  More
<input type="checkbox"/>	4	<b>tempat_lahir</b>	varchar(255)	utf8mb4_unicode_ci		No	None			Change  Drop  More
<input type="checkbox"/>	5	<b>tanggal_lahir</b>	date			No	None			Change  Drop  More
<input type="checkbox"/>	6	<b>fakultas</b>	varchar(255)	utf8mb4_unicode_ci		No	None			Change  Drop  More
<input type="checkbox"/>	7	<b>jurusan</b>	varchar(255)	utf8mb4_unicode_ci		No	None			Change  Drop  More
<input type="checkbox"/>	8	<b>ipk</b>	decimal(3,2)			No	None			Change  Drop  More
<input type="checkbox"/>	9	<b>created_at</b>	timestamp			Yes	NULL			Change  Drop  More
<input type="checkbox"/>	10	<b>updated_at</b>	timestamp			Yes	NULL			Change  Drop  More

## Modifikasi File Migration:

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

## Struktur table:

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	<b>id</b> 🛡	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2	<b>nim</b> 🔑	char(8)	utf8mb4_unicode_ci		No	None			Change  Drop  More
<input type="checkbox"/>	3	<b>nama</b>	varchar(255)	utf8mb4_unicode_ci		No	None			Change  Drop  More
<input type="checkbox"/>	4	<b>tempat_lahir</b>	varchar(255)	utf8mb4_unicode_ci		No	None			Change  Drop  More
<input type="checkbox"/>	5	<b>tanggal_lahir</b>	date			No	None			Change  Drop  More
<input type="checkbox"/>	6	<b>fakultas</b>	varchar(255)	utf8mb4_unicode_ci		No	None			Change  Drop  More
<input type="checkbox"/>	7	<b>jurusan</b>	varchar(255)	utf8mb4_unicode_ci		No	None			Change  Drop  More
<input type="checkbox"/>	8	<b>ipk</b>	decimal(3,2)			No	1.00			Change  Drop  More
<input type="checkbox"/>	9	<b>created_at</b>	timestamp			Yes	NULL			Change  Drop  More
<input type="checkbox"/>	10	<b>updated_at</b>	timestamp			Yes	NULL			Change  Drop  More

## Install composer require/dbal:

```
PS C:\PPW_2311104066_FaishalArifSetiawan\Pertemuan  
12\Praktikum\Guided> composer require doctrine/db  
al  
.composer.json has been updated  
Running composer update doctrine/dbal  
Loading composer repositories with package informa  
tion
```

Membuat Table Alter Mahasiswa:

```
12\Praktikum\Guided> php artisan make:migration a  
lter__mahasiswa_table --create=mahasiswa  
  
[INFO] Migration [C:\PPW_2311104066_FaishalArifs  
etiawan\Pertemuan 12\Praktikum\Guided\database\mig  
rations\2026_01_04_172952_alter__mahasiswa_table.  
php] created successfully.
```

Modifikasi file alter\_mahasiswa:

```
Praktikum > Guided > database > migrations > 2026_01_04_172952_alter__mahasiswa_table.php > PHP > AlterMahasiswa  
1  <?php  
2  Use Illuminate\Database\Migrations\Migration;  
3  use Illuminate\Database\Schema\Blueprint;  
4  use Illuminate\Support\Facades\Schema;  
5  0 references | 0 implementations  
6  class AlterMahasiswaTable extends Migration  
7  {  
8  0 references | 0 overrides | Trae: Explain | Doc | X  
9  public function up(): void{  
10 Schema::table('mahasiswa', callback: function (Blueprint $table): void {  
11 $table->renameColumn(from: 'nama',to: 'nama_lengkap');  
12 $table->text(column: 'alamat')->after(column: 'tanggal_lahir');  
13 $table->dropColumn(columns: 'ipk');  
14 );  
15 }  
16 0 references | 0 overrides | Trae: Explain | Doc | X  
17 public function down(): void{  
18 Schema::table('mahasiswa', callback: function (Blueprint $table): void {  
19 $table->renameColumn(from: 'nama_lengkap',to: 'nama');  
20 $table->dropColumn(columns: 'alamat');  
21 $table->decimal(column: 'ipk',total: 3,places: 2)->default(value: 1.00);  
22 });  
23 }
```

Struktur table:

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	<b>id</b> 	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	 Change  Drop More
<input type="checkbox"/>	2	<b>nim</b> 	char(8)	utf8mb4_unicode_ci		No	None			 Change  Drop More
<input type="checkbox"/>	3	<b>nama_lengkap</b>	varchar(255)	utf8mb4_unicode_ci		No	None			 Change  Drop More
<input type="checkbox"/>	4	<b>tempat_lahir</b>	varchar(255)	utf8mb4_unicode_ci		No	None			 Change  Drop More
<input type="checkbox"/>	5	<b>tanggal_lahir</b>	date			No	None			 Change  Drop More
<input type="checkbox"/>	6	<b>alamat</b>	text	utf8mb4_unicode_ci		No	None			 Change  Drop More
<input type="checkbox"/>	7	<b>fakultas</b>	varchar(255)	utf8mb4_unicode_ci		No	None			 Change  Drop More
<input type="checkbox"/>	8	<b>jurusan</b>	varchar(255)	utf8mb4_unicode_ci		No	None			 Change  Drop More
<input type="checkbox"/>	9	<b>created_at</b>	timestamp			Yes	NUL			 Change  Drop More
<input type="checkbox"/>	10	<b>updated_at</b>	timestamp			Yes	NUL			 Change  Drop More

Buat controller mahasiswa:

```

1 <?php
2 namespace App\Http\Controllers;
3 use Illuminate\Http\Request;
4 use Illuminate\Support\Facades\DB;
5 use App\Models\Mahasiswa;
6 class MahasiswaController extends Controller{
7     public function index(): string{
8         return "Index untuk mahasiswa";
9     }
10    public function insertSql(): string {
11        $result1 = DB::insert(query: "
12            INSERT INTO mahasiswas
13            (nim, nama_lengkap, tempat_lahir, tanggal_lahir, alamat, fakultas, jurusan)
14            VALUES (?, ?, ?, ?, ?, ?, ?)
15        ", bindings: ['11112222', 'Budi Santoso', 'Jakarta', '2001-03-15', 'Jl. Sudirman No. 5
16
17        return "Data mahasiswa berhasil ditambahkan";
18    }

```

Buat root untuk insert data di web.php:

```

1 <?php
2 use Illuminate\Support\Facades\Route;
3 use App\Http\Controllers\MahasiswaController;
4
5 // INSERT
6 Route::get(uri: '/insert-data1', action: [MahasiswaController::class, 'insertsql']);

```

Output:

← → ⌂ ⓘ 127.0.0.1:8000/insert-data1



Data mahasiswa berhasil ditambahkan

## B. UNGUIDED (Tugas Mandiri)

### 1. Soal 1:

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

Jawaban:

```
PS C:\PPW_2311104066_FaishalArifSetiawan\Pertemuan  
12\Latihan> cd "c:\PPW_2311104066_FaishalArifSetiawan\Pertemuan 12\Latihan\Unguided"; php artisan m  
ake:migration create_categories_table --create=cat  
egories  
  
[INFO] Migration [C:\PPW_2311104066_FaishalArifSetiawan\Pertemuan 12\Latihan\Unguided\database\mig  
rations\2026_01_04_180322_create_categories_table.  
php] created successfully.
```

```
PS C:\PPW_2311104066_FaishalArifSetiawan\Pertemuan  
12\Latihan\Unguided> cd "c:\PPW_2311104066_FaishalArifSetiawan\Pertemuan 12\Latihan\Unguided"; php  
artisan make:migration create_products_table --cre  
ate=products  
  
[INFO] Migration [C:\PPW_2311104066_FaishalArifSetiawan\Pertemuan 12\Latihan\Unguided\database\mig  
rations\2026_01_04_180327_create_products_table.ph  
p] created successfully.
```

```
🐘 0001_01_01_000000_create_users_ta... U
🐘 0001_01_01_000001_create_cache_t... U
🐘 0001_01_01_000002_create_jobs_ta... U
🐘 2026_01_04_180322_create_categori... U
🐘 2026_01_04_180327_create_product... U
```

Penjelasan:

Pada soal 12.1, saya membuat dua file migration untuk database e-commerce AMitra Furniture yang menjual berbagai jenis furniture. Migration pertama adalah tabel categories yang berfungsi untuk mengelompokkan produk furniture berdasarkan jenisnya seperti Sofa, Meja, Lemari, dan lain-lain, dimana setiap kategori memiliki nama dan deskripsi untuk memudahkan pengelolaan produk.

Migration kedua adalah tabel products yang menyimpan informasi detail tentang setiap produk furniture yang dijual, termasuk kode produk yang unik, nama produk, harga, dan jumlah stok yang tersedia. Tabel products juga dilengkapi dengan kolom-kolom khusus untuk furniture seperti material (bahan furniture seperti Kayu Jati atau Mahoni), dimensions (ukuran furniture), color (warna), dan image (untuk menyimpan foto produk).

Kedua tabel ini memiliki relasi dimana setiap produk terhubung dengan satu kategori melalui foreign key category\_id, sehingga kita bisa dengan mudah mengetahui produk tertentu masuk dalam kategori apa. Struktur database ini dirancang khusus untuk kebutuhan toko furniture online karena menyimpan informasi penting yang dibutuhkan pelanggan saat membeli furniture seperti bahan, ukuran, dan warna.

Dengan menggunakan migration Laravel, perubahan struktur database dapat dilacak dengan baik dan mudah untuk di-rollback jika terjadi kesalahan, serta memudahkan kolaborasi tim karena semua developer menggunakan struktur database yang sama. Migration ini juga memastikan data produk furniture AMitra dapat tersimpan dengan terorganisir dan siap digunakan untuk fitur-fitur e-commerce seperti katalog produk, pencarian, dan manajemen stok.

## 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

### Jawaban:

Insert data menggunakan raw sql:

```
public function insertRawSQL(): string
{
    // Insert category terlebih dahulu
    DB::insert(query: "
        INSERT INTO categories (name, description, created_at, updated_at)
        VALUES (?, ?, NOW(), NOW())
    ", bindings: ['Sofa', 'Koleksi sofa berkualitas tinggi']);
```

Insert data menggunakan QB:

```
public function insertQueryBuilder(): string
{
    // Insert category terlebih dahulu
    $categoryId = DB::table(table: 'categories')->insertGetId(values: [
        'name' => 'Meja',
        'description' => 'Berbagai jenis meja untuk rumah dan kantor',
        'created_at' => now(),
        'updated_at' => now(),
    ]);

    $result = DB::table(table: 'products')->insert(values: [
        'category_id' => $categoryId,
        'product_code' => 'FRN-MJ-001',
        'name' => 'Meja Makan Kayu Jati',
        'description' => 'Meja makan minimalis dengan bahan kayu jati solid',
        'price' => 3500000,
        'stock' => 15,
        'material' => 'Kayu Jati Solid',
        'dimensions' => '150x90x75 cm'
    ]);
```

Insert data menggunakan Eloquent ORM:

```
public function insertEloquent(): string
{
    $category = Category::create(attributes: [
        'name' => 'Lemari',
        'description' => 'Lemari pakaian dan lemari hias',
    ]);

    $category->save();
    $product = Product::create(attributes: [
        'category_id' => $category->id,
        'product_code' => 'FRN-LM-001',
        'name' => 'Lemari Pakaian 3 Pintu',
        'description' => 'Lemari pakaian dengan 3 pintu sliding dan cermin',
        'price' => 4200000,
        'stock' => 8,
        'material' => 'Kayu Mahoni',
        'dimensions' => '180x60x200 cm',
        'color' => 'Coklat Mahoni',
    ]);
}
```

Penjelasan:

Penjelasan Soal 12.2 - Fungsi Insert untuk E-commerce AMitra Furniture

Pada soal 12.2, saya membuat tiga fungsi berbeda untuk melakukan insert data produk furniture ke dalam database, masing-masing menggunakan metode yang berbeda yaitu Raw SQL Queries, Query Builder, dan Eloquent ORM. Ketiga fungsi ini dibuat di dalam ProductController yang berfungsi untuk menangani operasi database terkait produk furniture di toko online AMitra.

Fungsi pertama adalah insertRawSQL yang menggunakan perintah SQL murni dengan DB::insert untuk memasukkan data kategori Sofa dan produk Sofa Minimalis Modern beserta spesifikasi lengkapnya seperti harga Rp 5.500.000, material Kayu Jati, dan dimensi furniture. Fungsi kedua adalah insertQueryBuilder yang menggunakan Query Builder Laravel dengan sintaks yang lebih mudah dibaca, dimana saya insert kategori Meja dan produk Meja Makan Kayu Jati dengan harga Rp 3.500.000 beserta detail material dan dimensinya.

Fungsi ketiga adalah insertEloquent yang merupakan cara paling modern dan mudah menggunakan Eloquent ORM, dimana saya cukup memanggil method create pada Model Category dan Product untuk insert kategori Lemari dan produk Lemari Pakaian 3 Pintu seharga Rp 4.200.000. Setiap fungsi tidak hanya insert produk saja tetapi juga insert

kategorinya terlebih dahulu, sehingga relasi antara produk dan kategori terjaga dengan baik melalui foreign key category\_id.

Ketiga metode ini menunjukkan fleksibilitas Laravel dalam menangani operasi database, dimana Raw SQL memberikan kontrol penuh, Query Builder memberikan kemudahan dengan method yang chainable, dan Eloquent ORM memberikan cara paling elegan dengan pemanfaatan Model dan relationship. Setiap fungsi dapat diakses melalui route yang berbeda untuk testing, yaitu product/insert-sql, product/insert-qb, dan product/insert-eloquent yang semuanya akan mengembalikan pesan sukses setelah data berhasil disimpan ke database.

## BAB III

### PENUTUP

#### A. Kesimpulan

Pada praktikum Pertemuan 12 ini, saya berhasil mempelajari dan mengimplementasikan berbagai cara melakukan operasi database menggunakan Laravel framework, khususnya operasi insert data dengan tiga metode yang berbeda. Praktikum ini memberikan pemahaman mendalam tentang bagaimana Laravel menyediakan berbagai tingkat abstraksi untuk berinteraksi dengan database, mulai dari tingkat paling rendah dengan Raw SQL hingga tingkat paling tinggi dengan Eloquent ORM.

Saya telah membuat migration file untuk mendefinisikan struktur tabel database dengan kolom-kolom yang sesuai kebutuhan, serta mempelajari cara membuat relasi antar tabel menggunakan foreign key. Melalui praktikum ini, saya memahami pentingnya migration dalam Laravel untuk mengelola perubahan struktur database secara terorganisir, mudah di-tracking, dan dapat di-rollback jika terjadi kesalahan.

Implementasi tiga metode insert yang berbeda memberikan pengalaman langsung dalam memahami kelebihan dan kekurangan masing-masing pendekatan. Raw SQL Queries

memberikan kontrol penuh namun rentan terhadap SQL injection, Query Builder memberikan keamanan lebih baik dengan sintaks yang mudah dibaca, sedangkan Eloquent ORM memberikan cara paling elegan dan aman dengan memanfaatkan Model dan mass assignment protection.

Saya juga belajar membuat Model dengan property fillable untuk keamanan data, membuat Controller untuk menangani logika bisnis, dan membuat routes untuk mengakses fungsi-fungsi yang telah dibuat. Praktikum ini mengajarkan bahwa pemilihan metode yang tepat dalam operasi database sangat tergantung pada kebutuhan spesifik aplikasi, kompleksitas query, dan tingkat keamanan yang diinginkan.

Secara keseluruhan, praktikum ini memberikan fondasi yang kuat dalam memahami database operation di Laravel dan mempersiapkan saya untuk mengembangkan aplikasi web yang lebih kompleks dengan operasi CRUD yang lengkap dan terstruktur dengan baik.