

**LAPORAN PRAKTIKUM  
PERANCANGAN DAN PEMROGRAMAN WEB**

**MODUL XII  
LARAVEL 2**



**Universitas  
Telkom**

Oleh:

(Zaenarif Putra 'Ainurdin) (2311104049)

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

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Dasar Teori**

Laravel merupakan salah satu framework PHP yang menerapkan konsep MVC (Model–View–Controller) untuk mempermudah pengembangan aplikasi berbasis web. Dalam Laravel, pengelolaan database menjadi lebih terstruktur melalui fitur Migration, Model, DB Facade, Query Builder, dan Eloquent ORM.

Migration adalah fitur Laravel yang berfungsi sebagai version control untuk database. Migration memungkinkan pengembang untuk membuat, mengubah, dan menghapus struktur tabel database menggunakan kode PHP tanpa harus menulis perintah SQL secara langsung. Dengan migration, perubahan struktur database dapat dilakukan secara bertahap dan terdokumentasi dengan baik, serta dapat dijalankan kembali di lingkungan pengembangan yang berbeda.

Model pada Laravel merupakan representasi dari tabel dalam database. Model digunakan untuk mengelola data serta melakukan operasi CRUD (Create, Read, Update, Delete).

#### **1.2 Tujuan**

- a. Memahami fungsi dan cara kerja model pada framework laravel
- b. Mampu menerapkan migration pada aplikasi

#### **1.3 Manfaat**

- a. Kemudahan Pengelolaan Database  
Struktur database dapat dibuat dan dimodifikasi dengan cepat tanpa harus menghapus data yang sudah ada.
- b. Keamanan Data  
Laravel secara otomatis membantu mencegah serangan seperti SQL Injection melalui Query Builder dan Eloquent ORM.
- c. Kode Lebih Terstruktur dan Rapi  
Pemisahan antara logika aplikasi dan logika database membuat kode lebih mudah dibaca, dipahami, dan dipelihara.
- d. Efisiensi Pengembangan  
Pengembang tidak perlu menulis query SQL secara manual untuk setiap operasi, sehingga proses pengembangan menjadi lebih cepat.

e. Kemudahan Kolaborasi Tim

Migration memungkinkan setiap anggota tim memiliki struktur database yang sama melalui version control database.

## BAB II

## HASIL PRAKTIKUM

### 2.1 Guided

a. Source Code

```
.env

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=2311104049_ppw_putra
DB_USERNAME=root
DB_PASSWORD=


2025_12_21_060300_create_mahasiswa_table.php

<?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('mahasiswa', 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();
    }) ;
}

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

```

## 2025\_12\_21\_060554\_alter\_mahasiswa\_table.php

```

<?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::table('mahasiswa', function (Blueprint $table) {
            $table->renameColumn('nama', 'nama_lengkap');
            $table->text('alamat')->after('tanggal_lahir');
        });
    }
};

```

```
        $table->dropColumn('ipk');
    } );
}

/**
 * Reverse the migrations.
 */
public function down(): void
{
    Schema::table('mahasiswa', function (Blueprint $table) {
        $table->renameColumn('nama_lengkap', 'nama');
        $table->dropColumn('alamat');
        $table->decimal('ipk', 3, 2)->default(1.00);
    });
}
};
```

### web.php

```
<?php

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

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

Route::get('/insert-data',
[MahasiswaController::class, 'insertData']);
Route::get('/select-data',
[MahasiswaController::class, 'selectData']);
Route::get('/update-data',
[MahasiswaController::class, 'updateData']);
Route::get('/delete-data',
[MahasiswaController::class, 'deleteData']);
```

### MahasiswaController.php

```
<?php

namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Models\Mahasiswa;
use Illuminate\Support\Facades\DB;

class MahasiswaController extends Controller
{
    public function index()
    {
        return "Index untuk mahasiswa";
    }

    public function insertData()
    {
        // Menggunakan Eloquent ORM
        $mahasiswa = new Mahasiswa;
        $mahasiswa->nim = '20231001';
        $mahasiswa->nama_lengkap = 'Ahmad Fauzi';
        $mahasiswa->tempat_lahir = 'Surabaya';
        $mahasiswa->tanggal_lahir = '2001-05-15';
        $mahasiswa->alamat = 'Jl. Sudirman No. 45';
        $mahasiswa->fakultas = 'Fakultas Ilmu Komputer';
        $mahasiswa->jurusan = 'Sistem Informasi';

        $mahasiswa->save();

        // Tampilkan hasil
        dump($mahasiswa);
        return "Data mahasiswa berhasil disimpan!";
    }
}
?>
```

Mahasiswa.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Mahasiswa extends Model
{
    protected $fillable = [
        'nim',
        'nama_lengkap',
        'tempat_lahir',
        'tanggal_lahir',
        'alamat',
        'fakultas',
        'jurusan',
    ];
}
```

### database.php

```
'mysql' => [
    'driver' => 'mysql',
    'url' => env('DB_URL'),
    'host' => env('DB_HOST', '127.0.0.1'),
    'port' => env('DB_PORT', '3306'),
    'database' => env('DB_DATABASE',
'2311104049_ppw_putra'),
    'username' => env('DB_USERNAME', 'root'),
    'password' => env('DB_PASSWORD', ''),
    'unix_socket' => env('DB_SOCKET', ''),
    'charset' => env('DB_CHARSET', 'utf8mb4'),
    'collation' => env('DB_COLLATION',
'utf8mb4_unicode_ci'),
    'prefix' => '',
    'prefix_indexes' => true,
    'strict' => true,
    'engine' => null,
```

```

'options' => extension_loaded('pdo_mysql') ?
array_filter([
    (PHP_VERSION_ID >= 80500 ?
\Pdo\Mysql::ATTR_SSL_CA : \PDO::MYSQL_ATTR_SSL_CA) =>
env('MYSQL_ATTR_SSL_CA'),
]) : [],
],

```

## b. Screenshot Output

- Membuat file migration create\_mahasiswa\_table

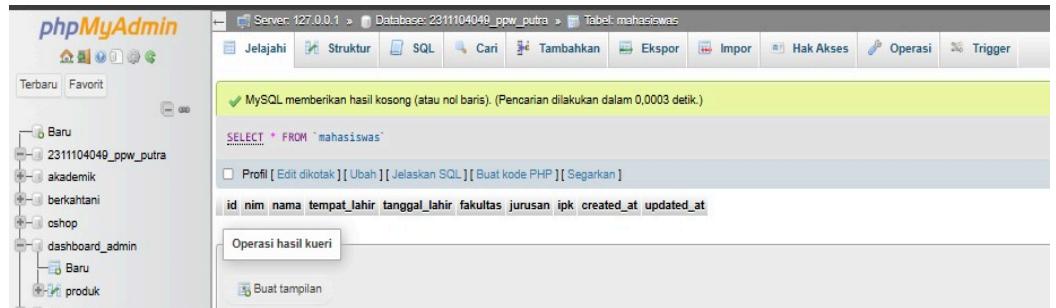
```

PS C:\xampp\htdocs\PPW_231104049_Zaenarif-Putra-Ainurdin\Pertemuan12> cd .\Praktikum
PS C:\xampp\htdocs\PPW_231104049_Zaenarif-Putra-Ainurdin\Pertemuan12\Praktikum> php artisan make:migration create_mahasiswa_table --create=mahasiswa
INFO Migration [C:\xampp\htdocs\PPW_231104049_Zaenarif-Putra-Ainurdin\Pertemuan12\Praktikum\database\migrations\2025_12_21_055923_create_mahasiswa_table.php] created successfully.

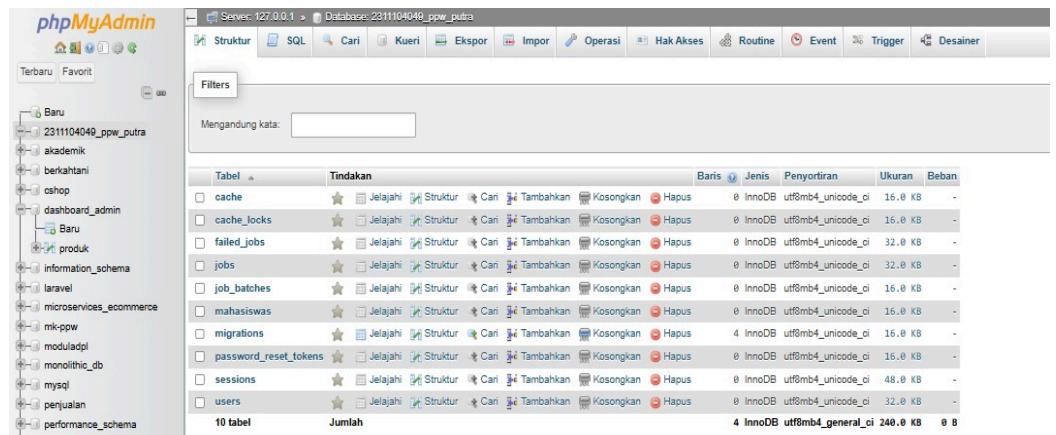
PS C:\xampp\htdocs\PPW_231104049_Zaenarif-Putra-Ainurdin\Pertemuan12\Praktikum>

```

- Membuat db pada mysql



- Menyambungkan dengan menggunakan perintah php artisan migrate



- Melihat status dari migrate yang sudah berjalan dengan perintah php artisan migrate:status

```

PS C:\xampp\htdocs\PPW_2311104049_Zaenarif-Putra-Ainurdin\Pertemuan12\Praktikum> php artisan migrate:status
Migration name ..... Batch / Status
0001_01_01_000001_create_users_table [1] Ran
0001_01_01_000001_create_cache_table [1] Ran
0001_01_01_000002_create_jobs_table [1] Ran
2025_12_21_060300_create_mahasiswa_table [1] Ran

```

- Struktur lengkap db

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	<code>id</code>	bigint(20)		UNSIGNED	Tidak	Tidak ada		AUTO_INCREMENT	Ubah  Hapus  Lainnya
2	<code>nim</code>	char(8)	utf8mb4_unicode_ci		Tidak	Tidak ada			Ubah  Hapus  Lainnya
3	<code>nama_lengkap</code>	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada			Ubah  Hapus  Lainnya
4	<code>tempat_lahir</code>	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada			Ubah  Hapus  Lainnya
5	<code>tanggal_lahir</code>	date			Tidak	Tidak ada			Ubah  Hapus  Lainnya
6	<code>alamat</code>	text	utf8mb4_unicode_ci		Tidak	Tidak ada			Ubah  Hapus  Lainnya
7	<code>fakultas</code>	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada			Ubah  Hapus  Lainnya
8	<code>jurusan</code>	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada			Ubah  Hapus  Lainnya
9	<code>created_at</code>	timestamp			Ya	NULL			Ubah  Hapus  Lainnya
10	<code>updated_at</code>	timestamp			Ya	NULL			Ubah  Hapus  Lainnya

- Membuat sebuah pergantian pada table mahasiswa

```

.env | 2025_12_21_060300_create_mahasiswa_table.php U X | 2025_12_21_060554_alter_mahasiswa_table.php U
Praktikum > database > migrations > 2025_12_21_060300_create_mahasiswa_table.php
5   use Illuminate\Support\Facades\Schema;
6
7   return new class extends Migration
8   {
9       /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('mahasiswa', function (Blueprint $table) {
15             $table->id();
16             $table->char('nim',8)->unique();
17             $table->string('nama');
18             $table->string('tempat_lahir');
19             $table->date('tanggal_lahir');
20             $table->string('fakultas');
21             $table->string('jurusan');
22             $table->decimal('ipk',3,2)->default(1.00);
23             $table->timestamps();
24         });
25     }

```

- Data setelah diubah dan di migrate rollback kembali

```

● PS C:\xampp\htdocs\PPW_2311104049_Zaenarif-Putra-Ainurdin\Pertemuan12\Praktikum> php artisan migrate:status
Migration name ..... Batch / Status
0001_01_01_000000_create_users_table ..... [1] Ran
0001_01_01_000001_create_cache_table ..... [1] Ran
0001_01_01_000002_create_jobs_table ..... [1] Ran
2025_12_21_060300_create_mahasiswa_table ..... Pending
2025_12_21_060554_alter_mahasiswa_table ..... Pending

● PS C:\xampp\htdocs\PPW_2311104049_Zaenarif-Putra-Ainurdin\Pertemuan12\Praktikum> php artisan migrate
INFO Running migrations.

2025_12_21_060300_create_mahasiswa_table ..... 37.63ms DONE
2025_12_21_060554_alter_mahasiswa_table ..... 53.67ms DONE

❖ PS C:\xampp\htdocs\PPW_2311104049_Zaenarif-Putra-Ainurdin\Pertemuan12\Praktikum>

```

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	<b>id</b>	bigint(20)		UNSIGNED	Tidak	Tidak ada		AUTO_INCREMENT	Ubah  Hapus Lainnya
2	<b>nim</b>	char(8)	utf8mb4_unicode_ci		Tidak	Tidak ada			Ubah  Hapus Lainnya
3	<b>nama_lengkap</b>	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada			Ubah  Hapus Lainnya
4	<b>tempat_lahir</b>	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada			Ubah  Hapus Lainnya
5	<b>tanggal_lahir</b>	date			Tidak	Tidak ada			Ubah  Hapus Lainnya
6	<b>alamat</b>	text	utf8mb4_unicode_ci		Tidak	Tidak ada			Ubah  Hapus Lainnya
7	<b>fakultas</b>	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada			Ubah  Hapus Lainnya
8	<b>jurusan</b>	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada			Ubah  Hapus Lainnya
9	<b>created_at</b>	timestamp			Ya	NULL			Ubah  Hapus Lainnya
10	<b>updated_at</b>	timestamp			Ya	NULL			Ubah  Hapus Lainnya

Jelajahi Struktur SQL Cari Tambahkan Eksport Impor

Tampilkan kotak kueri

Kueri SQL Anda berhasil dieksekusi.

```
DESC mahasiswa;
```

[ Edit dikotak ] [ Ubah ] [ Buat kode PHP ]

Extra options

Field	Type	Null	Key	Default	Extra
<b>id</b>	bigint(20) unsigned	NO	PRI	NULL	auto_increment
<b>nim</b>	char(8)	NO	UNI	NULL	
<b>nama_lengkap</b>	varchar(255)	NO		NULL	
<b>tempat_lahir</b>	varchar(255)	NO		NULL	
<b>tanggal_lahir</b>	date	NO		NULL	
<b>alamat</b>	text	NO		NULL	
<b>fakultas</b>	varchar(255)	NO		NULL	
<b>jurusan</b>	varchar(255)	NO		NULL	
<b>created_at</b>	timestamp	YES		NULL	
<b>updated_at</b>	timestamp	YES		NULL	

Operasi hasil kueri

Cetak Salin ke clipboard Buat tampilan

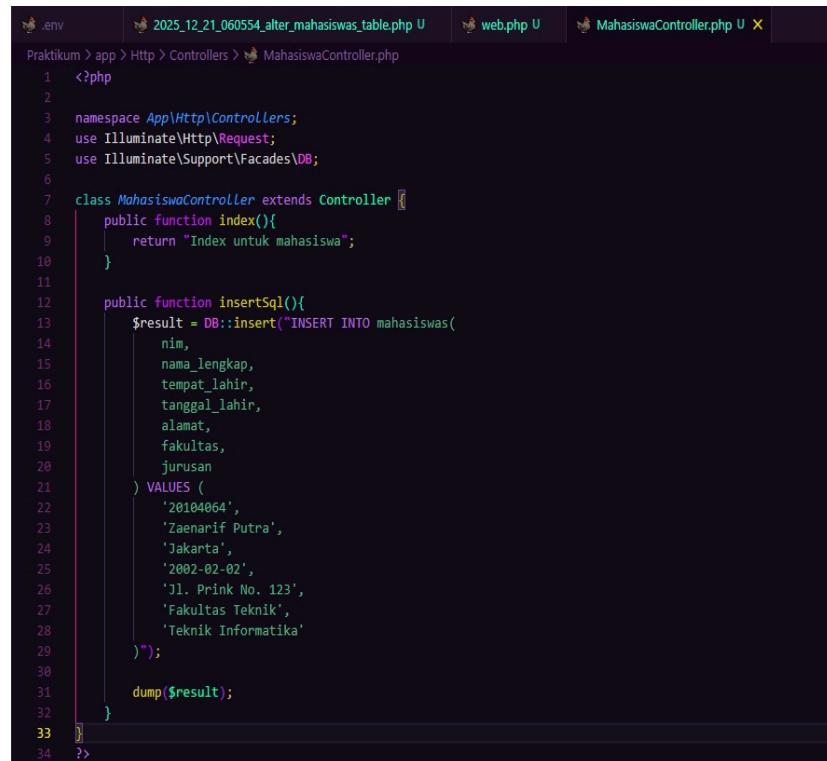
- Membuat sebuah migrations baru juga yang diberi nama 2025\_12\_21\_060554\_alter\_mahasiswa\_table.php

```

1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::table('mahasiswa', function (Blueprint $table) {
15             $table->renameColumn('nama', 'nama_lengkap');
16             $table->text('alamat')->after('tanggal_lahir');
17             $table->dropColumn('ipk');
18         });
19     }
20
21     /**
22      * Reverse the migrations.
23      */
24     public function down(): void
25     {
26         Schema::table('mahasiswa', function (Blueprint $table) {
27             $table->renameColumn('nama_lengkap', 'nama');
28             $table->dropColumn('alamat');
29             $table->decimal('ipk', 3, 2)->default(1.00);
30         });
31     }
32 };
33

```

- Test menginputkan sebuah data



```

.phpenv | 2025_12_21_060554_alter_mahasiswa_table.php U | web.php U | MahasiswaController.php U X
Praktikum > app > Http > Controllers > MahasiswaController.php
1 <?php
2
3 namespace App\Http\Controllers;
4 use Illuminate\Http\Request;
5 use Illuminate\Support\Facades\DB;
6
7 class MahasiswaController extends Controller {
8     public function index(){
9         return "Index untuk mahasiswa";
10    }
11
12    public function insertSql(){
13        $result = DB::insert("INSERT INTO mahasiswa(
14            nim,
15            nama_lengkap,
16            tempat_lahir,
17            tanggal_lahir,
18            alamat,
19            fakultas,
20            jurusan
21        ) VALUES (
22            '20104064',
23            'Zaenarif Putra',
24            'Jakarta',
25            '2002-02-02',
26            'Jl. Prink No. 123',
27            'Fakultas Teknik',
28            'Teknik Informatika'
29        )");
30
31        dump($result);
32    }
33}
34 ?>

```

	<input type="checkbox"/>	Ubah	<input type="checkbox"/> Salin	<input checked="" type="checkbox"/> Hapus	id	nim	nama_lengkap	tempat_lahir	tanggal_lahir	alamat	fakultas	jurusan	created_at	updated_at
	<input type="checkbox"/>				1	20104084	Zaenarif Putra	Jakarta	2002-02-02	Jl. Prnk No. 123	Fakultas Teknik	Teknik Informatika	NULL	NULL
<hr/>														
<hr/>														
<input type="checkbox"/> Tampilkan semua    Jumlah baris: <select>25</select> Saring baris: <input type="text"/> Cari di tabel ini														
<a href="#"> Cetak</a> <a href="#"> Salin ke clipboard</a> <a href="#"> Eksport</a> <a href="#"> Tampilkan bagan</a> <a href="#"> Buat tampilan</a>														

- Dengan menambahkan route pada web.php

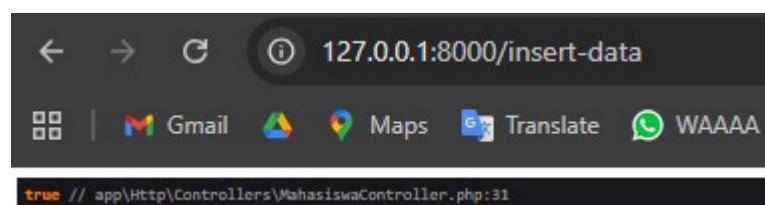
```

Praktikum > routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\MahasiswaController;
5
6  Route::get('/', function () {
7      return view('welcome');
8  });
9
10 Route::get('/insert-data', [MahasiswaController::class,'insertSql']);
11 Route::get('/select-data', [MahasiswaController::class,'selectData']);
12 Route::get('/update-data', [MahasiswaController::class,'updateData']);
13 Route::get('/delete-data', [MahasiswaController::class,'deleteData']);

```

- Mengubah function tujuan yang tadinya insertSql menjadi insertData

```
.env | 2025_12_21_060554_alter_mahasiswa_table.php U | web.php U | MahasiswaController.php U X
Praktikum > app > Http > Controllers > MahasiswaController.php
1 <?php
2
3 namespace App\Http\Controllers;
4 use Illuminate\Http\Request;
5 use Illuminate\Support\Facades\DB;
6
7 class MahasiswaController extends Controller {
8     public function index(){
9         return "Index untuk mahasiswa";
10    }
11
12    public function insertData(){
13        $result = DB::insert("INSERT INTO mahasiswas(
14            nim,
15            nama_lengkap,
16            tempat_lahir,
17            tanggal_lahir,
18            alamat,
19            fakultas,
20            jurusan
21        ) VALUES (
22            '20104099',
23            'Muhammad Hamada',
24            'Jakarta',
25            '2002-02-05',
26            ''Jl. Merdeka No. 10',
27            'Fakultas Teknik',
28            'Sistem Informasi'
29        )" );
30
31        dump($result);
32    }
33 }
34 ?>
```



	<input type="checkbox"/> Profil	[Edit dikotak]	[Ubah]	[Jelaskan SQL]	[Buat kode PHP]	[Segarkan]		
	<input type="checkbox"/>	Tampilkan semua	Jumlah baris:	25	Saring baris:	Cari di tabel ini	Sort by key:	Tidak ada
		Extra options						
	<input type="checkbox"/>		Ubah		Salin		Hapus	
1	<input type="checkbox"/>		Ubah		Salin		Hapus	
2	<input type="checkbox"/>		Ubah		Salin		Hapus	
	<input type="checkbox"/>	Pilih Semua	Dengan pilihan:		Ubah		Salin	
								Eksport
		<input type="checkbox"/>	Tampilkan semua	Jumlah baris:	25	Saring baris:	Cari di tabel ini	Sort by key:
								Tidak ada
		Operasi hasil kueri						
			Cetak		Salin ke clipboard		Eksport	
								Tampilkan bagan
								Buat tampilan

- Menambahkan model mahasiswa

```

.env 2025_12_21_060554_alter_mahasiswas_table.php U web.php U MahasiswaController
Praktikum > app > Models > Mahasiswa.php
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Mahasiswa extends Model
8 {
9     protected $fillable = [
10         'nim',
11         'nama_lengkap',
12         'tempat_lahir',
13         'tanggal_lahir',
14         'alamat',
15         'fakultas',
16         'jurusan',
17     ];
18 }

```

- Kemudian menyesuaikan yang ada di model agar lasung di panggil di controller

```

.env | 2025_12_21_060554_alter_mahasiswa_table.php U | web.php U | MahasiswaController.php X
Praktikum > app > Http > Controllers > MahasiswaController.php
1  <?php
2
3  namespace App\Http\Controllers;
4  use Illuminate\Http\Request;
5  use App\Models\Mahasiswa;
6  use Illuminate\Support\Facades\DB;
7
8  class MahasiswaController extends Controller
9  {
10     public function index()
11     {
12         return "Index untuk mahasiswa";
13     }
14
15     public function insertData()
16     {
17         // Menggunakan Eloquent ORM
18         $mahasiswa = new Mahasiswa;
19         $mahasiswa->nim = '20231001';
20         $mahasiswa->nama_lengkap = 'Ahmad Fauzi';
21         $mahasiswa->tempat_lahir = 'Surabaya';
22         $mahasiswa->tanggal_lahir = '2001-05-15';
23         $mahasiswa->alamat = 'Jl. Sudirman No. 45';
24         $mahasiswa->fakultas = 'Fakultas Ilmu Komputer';
25         $mahasiswa->jurusan = 'Sistem Informasi';
26
27         $mahasiswa->save();
28
29         // Tampilkan hasil
30         dump($mahasiswa);
31         return "Data mahasiswa berhasil disimpan!";
32     }
33 }
34 ?>

```

- Dan data berhasil ditambahkan

**App\Models\Mahasiswa {#267 ► }**

Data mahasiswa berhasil disimpan!

<input type="checkbox"/> Profil	<input type="checkbox"/> Edit dikotak	<input type="checkbox"/> Ubah	<input type="checkbox"/> Jelaskan SQL	<input type="checkbox"/> Buat kode PHP	<input type="checkbox"/> Segarkan						
<input type="checkbox"/> Tampilkan semua   Jumlah baris: 25   Saring baris: Cari di tabel ini   Sort by key: Tidak ada											
Extra options											
<input type="checkbox"/>	<input type="checkbox"/> Ubah	<input type="checkbox"/> Salin	<input type="checkbox"/> Hapus	1	20104084	Zaenarif Putra	Jakarta	2002-02-02	Jl. Prink No. 123	Fakultas Teknik	Teknik Informatika
<input type="checkbox"/>	<input type="checkbox"/> Ubah	<input type="checkbox"/> Salin	<input type="checkbox"/> Hapus	2	20104099	Muhammad Hamada	Jakarta	2002-02-05	Jl.Merdeka No. 10	Fakultas Teknik	Sistem Informasi
<input type="checkbox"/>	<input type="checkbox"/> Ubah	<input type="checkbox"/> Salin	<input type="checkbox"/> Hapus	3	20231001	Ahmad Fauzi	Surabaya	2001-05-15	Jl. Sudirman No. 45	Fakultas Ilmu Komputer	Sistem Informasi

↑  Pilih Semua Dengan pilihan:  Ubah  Salin  Hapus  Eksport

### c. Dekripsi Program

Program yang dibuat merupakan aplikasi Laravel sederhana untuk pengelolaan data mahasiswa yang terhubung dengan database MySQL menggunakan fitur Migration dan Eloquent ORM. Konfigurasi koneksi database diatur melalui file .env dan database.php, yang berfungsi menentukan host, port, nama database, username, dan password agar aplikasi dapat berkomunikasi dengan database. Struktur tabel mahasiswa dibuat menggunakan file migration create\_mahasiswa\_table, di mana setiap syntax seperti \$table->id() berfungsi sebagai primary key, \$table->char('nim',8)->unique() sebagai identitas unik mahasiswa, serta \$table->timestamps() untuk mencatat waktu pembuatan dan pembaruan data, kemudian dimodifikasi melalui migration alter\_mahasiswa\_table yang mengubah nama kolom, menambah alamat, dan menghapus IPK. Alur program dimulai dari route di web.php yang mengarahkan URL /insert-data ke method insertData pada MahasiswaController, lalu controller tersebut menggunakan Eloquent ORM dengan membuat objek Mahasiswa, mengisi setiap atribut sesuai kolom tabel, dan menyimpannya ke database menggunakan \$mahasiswa->save(). Model Mahasiswa.php berperan sebagai representasi tabel database yang menjembatani controller dan database, sedangkan output akhir dari program adalah data mahasiswa yang berhasil tersimpan di database serta ditampilkan menggunakan dump(\$mahasiswa) dan pesan konfirmasi bahwa data berhasil disimpan.

## 2.2 Unguided

### 1. Latihan

#### a. Source Code

##### KapsterController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use App\Models\Kapster;
```

```
class KapsterController extends Controller
{
    public function index()
    {
        $kapsters = Kapster::latest()->get();
        return view('kapster.index', compact('kapsters'));
    }

    public function storeRaw(Request $r)
    {
        DB::insert(
            "INSERT INTO kapsters
                (nama,email,no_hp,umur,tanggal_masuk,admin_id,outlet_id,shift_id,created_at,updated_at)
                VALUES  (?, ?, ?, ?, ?, ?, ?, ?, ?, NOW() ,NOW() )",
            [
                $r->nama,
                $r->email,
                $r->no_hp,
                $r->umur,
                $r->tanggal_masuk,
                1,
                $r->outlet_id,
                $r->shift_id
            ]
        );
        return back()->with('success', 'RAW SQL berhasil');
    }

    public function storeQuery(Request $r)
    {
        DB::table('kapsters')->insert([
            'nama' => $r->nama,
            'email' => $r->email,
            'no_hp' => $r->no_hp,
```

```

        'umur' => $r->umur,
        'tanggal_masuk' => $r->tanggal_masuk,
        'admin_id' => 1,
        'outlet_id' => $r->outlet_id,
        'shift_id' => $r->shift_id,
        'created_at' => now(),
        'updated_at' => now()
    ]);

    return back()->with('success', 'Query Builder berhasil');
}

public function storeEloquent(Request $r)
{
    Kapster::create([
        'nama' => $r->nama,
        'email' => $r->email,
        'no_hp' => $r->no_hp,
        'umur' => $r->umur,
        'tanggal_masuk' => $r->tanggal_masuk,
        'admin_id' => 1,
        'outlet_id' => $r->outlet_id,
        'shift_id' => $r->shift_id
    ]);

    return back()->with('success', 'Eloquent ORM berhasil');
}

```

## Admin.php

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

```

```
class Admin extends Model
{
    protected $fillable = ['nama', 'email'];
    public function kapsters() {
        return $this->hasMany(Kapster::class);
    }
}
```

### Kapster.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Kapster extends Model
{
    protected $fillable = [
        'nama', 'email', 'no_hp', 'umur', 'tanggal_masuk',
        'admin_id', 'outlet_id', 'shift_id'
    ];

    public function admin(){ return
$this->belongsTo(Admin::class); }
    public function outlet(){ return
$this->belongsTo(Outlet::class); }
    public function shift(){ return
$this->belongsTo(Shift::class); }
}
```

### Outlet.php

```
<?php

namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Model;

class Outlet extends Model
{
    protected $fillable = ['nama_outlet', 'alamat'];
}
```

### Shift.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Shift extends Model
{
    protected $fillable =
['nama_shift', 'jam_masuk', 'jam_keluar'];
}
```

### User.php

```
<?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;

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

    /**
     * The attributes that should be mutated in the database
     * as they are not always directly set by models.
     *
     * @var array
     */
    protected $fillable = [
        'name', 'email', 'password',
    ];
    protected $hidden = [
        'password', 'remember_token',
    ];
    protected $casts = [
        'email_verified_at' => 'datetime',
    ];
}
```

```
 * The attributes that are mass assignable.
 *
 * @var list<string>
 */
protected $fillable = [
    'name',
    'email',
    'password',
];

/**
 * 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',
    ];
}
```

## Database.php

```
'mysql' => [
```

```
        'driver' => 'mysql',
        'url' => env('DB_URL'),
        'host' => env('DB_HOST', '127.0.0.1'),
        'port' => env('DB_PORT', '3306'),
        'database' => env('DB_DATABASE',
            'db_adabarbershop'),
        'username' => env('DB_USERNAME', 'root'),
        'password' => env('DB_PASSWORD', ''),
        'unix_socket' => env('DB_SOCKET', ''),
        'charset' => env('DB_CHARSET', 'utf8mb4'),
        'collation' => env('DB_COLLATION',
            'utf8mb4_unicode_ci'),
        'prefix' => '',
        'prefix_indexes' => true,
        'strict' => true,
        'engine' => null,
        'options' => extension_loaded('pdo_mysql') ?
array_filter([
    (PHP_VERSION_ID >= 80500 ?
\Pdo\Mysql::ATTR_SSL_CA : \PDO::MYSQL_ATTR_SSL_CA) =>
env('MYSQL_ATTR_SSL_CA'),
]) : [],
],
],
```

## 2025\_12\_25\_163939\_create\_admins\_table.php

```
<?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('admins', function (Blueprint $table)  
{  
        $table->id();  
        $table->string('nama');  
        $table->string('email')->unique();  
        $table->timestamps();  
    });  
}  
  
/**  
 * Reverse the migrations.  
 */  
public function down(): void  
{  
    Schema::dropIfExists('admins');  
}  
};
```

## 2025\_12\_25\_164012\_create\_outlets\_table.php

```
<?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('outlets', function (Blueprint  
$table) {  
            $table->id();  
            $table->string('nama_outlet');
```

```
        $table->text('alamat');
        $table->timestamps();
    } );
}

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

2025\_12\_25\_164053\_create\_shifts\_table.php

```
<?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('shifts', function (Blueprint $table)
        {
            $table->id();
            $table->string('nama_shift');
            $table->time('jam_masuk');
            $table->time('jam_keluar');
            $table->timestamps();
        });
    }
};
```

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

2025\_12\_25\_164117\_create\_kapsters\_table.php

```
<?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('kapsters', function (Blueprint
$table) {
            $table->id();
            $table->string('nama');
            $table->string('email')->unique();
            $table->string('no_hp');
            $table->integer('umur');
            $table->date('tanggal_masuk');
            $table->foreignId('admin_id')->constrained();
            $table->foreignId('outlet_id')->constrained();
            $table->foreignId('shift_id')->constrained();
            $table->timestamps();
        });
    }
};
```

```
}

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

};
```

### Index.blade.php (Dashboard view)

```
<!DOCTYPE html>
<html>
<head>
    <title>Dashboard Admin - Adabarbershop</title>
    <style>
        body { font-family: Arial; margin: 40px; }
        .menu a {
            display: inline-block;
            margin-right: 15px;
            padding: 10px 15px;
            background: #333;
            color: white;
            text-decoration: none;
        }
        .menu a:hover {
            background: #555;
        }
    </style>
</head>
<body>

    <h1>Dashboard Admin</h1>
    <p>Selamat datang di sistem Adabarbershop</p>

    <div class="menu">
```

```
</a href="/kapster">Manajemen Kapster</a>
</div>

</body>
</html>
```

### form.blade.php (Kapster view)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Form</title>
</head>
<body>
    <input type="text" name="nama" placeholder="Nama
Kapster" required>
    <input type="email" name="email" placeholder="Email"
required>
    <input type="text" name="no_hp" placeholder="No HP"
required>
    <input type="number" name="umur" placeholder="Umur"
required>
    <input type="date" name="tanggal_masuk" required>

    <input type="hidden" name="outlet_id" value="1">
    <input type="hidden" name="shift_id" value="1">
</body>
</html>
```

### Index.blade.php (Kapster view)

```
<!DOCTYPE html>
<html>
<head>
    <title>Manajemen Kapster</title>
```

```

<style>
    body { font-family: Arial; margin: 40px; }
    table { width: 100%; border-collapse: collapse;
margin-top: 20px; }
    th, td { border: 1px solid #ccc; padding: 8px; }
    th { background: #f0f0f0; }
    form { margin-bottom: 30px; padding: 20px; border:
1px solid #ddd; }
    input, button { margin-top: 8px; padding: 8px;
width: 100%; }
    h2 { margin-top: 0; }
</style>
</head>
<body>

<h1>Manajemen Kapster</h1>

@if(session('success'))
    <p style="color: green">{{ session('success') }}</p>
@endif

{ {-- FORM RAW SQL --} }
<form method="POST" action="/kapster/raw">
    @csrf
    <h2>Insert Data Kapster (Raw SQL)</h2>
    @include('kapster.form')
    <button type="submit">Simpan Raw SQL</button>
</form>

{ {-- FORM QUERY BUILDER --} }
<form method="POST" action="/kapster/query">
    @csrf
    <h2>Insert Data Kapster (Query Builder)</h2>
    @include('kapster.form')
    <button type="submit">Simpan Query Builder</button>
</form>

```

```

{ {-- FORM ELOQUENT ORM --} }

<form method="POST" action="/kapster/eloquent">
    @csrf
    <h2>Insert Data Kapster (Eloquent ORM)</h2>
    @include('kapster.form')
    <button type="submit">Simpan Eloquent ORM</button>
</form>

<h2>Data Kapster</h2>

<table>
    <tr>
        <th>Nama</th>
        <th>Email</th>
        <th>No HP</th>
        <th>Umur</th>
        <th>Tanggal Masuk</th>
    </tr>
    @foreach($kapsters as $k)
    <tr>
        <td>{{ $k->nama }}</td>
        <td>{{ $k->email }}</td>
        <td>{{ $k->no_hp }}</td>
        <td>{{ $k->umur }}</td>
        <td>{{ $k->tanggal_masuk }}</td>
    </tr>
    @endforeach
</table>

</body>
</html>

```

web.php

```

<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\KapsterController;

```

```

Route::get('/', fn () => view('dashboard.index'));

Route::get('/kapster', [KapsterController::class, 'index']);
Route::post('/kapster/raw', [KapsterController::class,
    'storeRaw']);
Route::post('/kapster/query', [KapsterController::class,
    'storeQuery']);
Route::post('/kapster/eloquent', [KapsterController::class,
    'storeEloquent']);

```

.env

```

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=db_adabarbershop
DB_USERNAME=root
DB_PASSWORD=

```

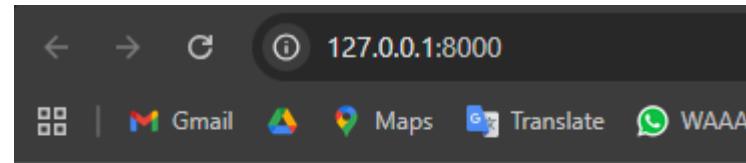
## b. Screenshot Output

### 1. Database

The screenshot shows the MySQL Workbench interface with the database 'db\_adabarbershop' selected. The main window displays a list of 13 tables: admins, cache, cache\_locks, failed\_jobs, jobs, job\_batches, kapsters, migrations, outlets, password\_reset\_tokens, sessions, shifts, and users. Each table row includes icons for Jelajahi (Browse), Struktur (Structure), Cari (Search), Tambahkan (Add), Kosongkan (Empty), and Hapus (Delete). A summary at the bottom indicates there are 13 tables with a total size of 368.0 KB.

Tabel	Tindakan	Baris	Jenis	Penyortiran	Ukuran	Beban
admins		0	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
cache		0	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
cache_locks		0	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
failed_jobs		0	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
jobs		0	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
job_batches		0	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
kapsters		0	InnoDB	utf8mb4_unicode_ci	80.0 KB	-
migrations		7	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
outlets		0	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
password_reset_tokens		0	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
sessions		2	InnoDB	utf8mb4_unicode_ci	48.0 KB	-
shifts		0	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
users		0	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
13 tabel	Jumlah	9	InnoDB	utf8mb4_general_ci	368.0 KB	0 B

### 2. Halaman Awal



# Dashboard Admin

Selamat datang di sistem Adabarbershop

Manajemen Kapster

### 3. Raw Sql

Manajemen Kapster

RAW SQL berhasil

## Insert Data Kapster (Raw SQL)

Nama Kapster

Email

No HP

Umur

dd / mm / yyyy

	Ubah	Salin	Hapus	id	nama	email	no_hp	umur	tanggal_masuk	admin_id	outlet_id	shift_id	created_at	updated_at
<input type="checkbox"/>	Ubah	Salin	Hapus	2	Arif	2345555@gmail.com	757572382	22	2005-03-22	1	1	1	2025-12-27 15:49:30	2025-12-27 15:49:30
<input type="checkbox"/>	 Ubah	 Salin	 Hapus	3	Arif	sddadasaa@gmail.com	087647575757	22	2032-03-22	1	1	1	2025-12-27 15:49:53	2025-12-27 15:49:53

#### 4. Query Builder

Manajemen Kapster

Query Builder berhasil

**Insert Data Kapster (Query Builder)**

---

<input type="text" value="Nama Kapster"/>
<input type="text" value="Email"/>
<input type="text" value="No HP"/>
<input type="text" value="Umur"/>
<input type="text" value="dd/mm/yyyy"/>
<input type="text"/>

	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	id	nama	email	no_hp	umur	tanggal_masuk	admin_id	outlet_id	shift_id	created_at	updated_at
<input type="checkbox"/>	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	2	Anif	2345555@gmail.com	757572382	22	2005-03-22	1	1	1	2025-12-27 15:49:30	2025-12-27 15:49:30
<input type="checkbox"/>	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	3	Anif	sdddasaae@gmail.com	087847575757	22	2032-03-22	1	1	1	2025-12-27 15:49:53	2025-12-27 15:49:53
<input type="checkbox"/>	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	4	melvin	mel@gmail.com	656565373223	19	2017-04-02	1	1	1	2025-12-27 08:52:19	2025-12-27 08:52:19

## 5. Eloquent ORM

### Manajemen Kapster

Eloquent ORM berhasil

rey	rey@gmail.com	6565653443222												
ilham	ham@gmail.com	447637567245												
Extra options														
	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	id	nama	email	no_hp	umur	tanggal_masuk	admin_id	outlet_id	shift_id	created_at	updated_at
<input type="checkbox"/>	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	2	Anif	2345555@gmail.com	757572382	22	2005-03-22	1	1	1	2025-12-27 15:49:30	2025-12-27 15:49:30
<input type="checkbox"/>	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	3	Anif	sdddasaae@gmail.com	087847575757	22	2032-03-22	1	1	1	2025-12-27 15:49:53	2025-12-27 15:49:53
<input type="checkbox"/>	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	4	melvin	mel@gmail.com	656565373223	19	2017-04-02	1	1	1	2025-12-27 08:52:19	2025-12-27 08:52:19
<input type="checkbox"/>	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	5	ilham	ham@gmail.com	447637567245	20	2019-05-04	1	1	1	2025-12-27 08:54:36	2025-12-27 08:54:36
<input type="checkbox"/>	<input type="button" value="Ubah"/>	<input type="button" value="Salin"/>	<input type="button" value="Hapus"/>	6	rey	rey@gmail.com	6565653443222	21	2018-02-11	1	1	1	2025-12-27 08:55:47	2025-12-27 08:55:47

#### c. Deksripsi Program

Program yang dibuat adalah aplikasi web Manajemen Kapster Adabarbershop berbasis framework Laravel yang menerapkan arsitektur MVC (Model–View–Controller) untuk mengelola data kapster secara terstruktur dan terhubung dengan database relasional MySQL. Alur kerja program dimulai dari route (web.php) yang menerima request HTTP dari pengguna, kemudian meneruskannya ke KapsterController sebagai pengendali logika aplikasi; di dalam controller terdapat tiga algoritma insert data yang berbeda sesuai

tugas, yaitu menggunakan Raw SQL (DB::insert), Query Builder (DB::table()->insert), dan Eloquent ORM (Kapster::create), yang masing-masing menunjukkan cara kerja Laravel dalam berinteraksi dengan database pada level rendah, menengah, dan tinggi. Data yang dikirim dari View (Blade) berupa form HTML diproses oleh controller, divalidasi secara implisit oleh database melalui foreign key (admin\_id, outlet\_id, shift\_id), lalu disimpan ke tabel kapsters, sedangkan Model (Kapster, Admin, Outlet, Shift) berfungsi sebagai representasi tabel database sekaligus penghubung relasi antar data menggunakan Eloquent (belongsTo, hasMany). Migration berperan sebagai algoritma pembentukan struktur database yang konsisten (tabel, kolom, relasi), sementara Seeder berfungsi mengisi data awal (admin, outlet, shift) agar constraint database terpenuhi sebelum proses insert kapster dilakukan. Output akhir dari program ini adalah halaman web interaktif yang menampilkan form input kapster dan tabel daftar kapster yang tersimpan, sehingga pengguna dapat langsung melihat hasil eksekusi algoritma insert data melalui antarmuka web secara real time.

## **BAB III**

### **KESIMPULAN & SARAN**

#### **3.1 Kesimpulan**

Berdasarkan program yang dibuat, dapat disimpulkan bahwa aplikasi Adabarbershop berbasis Laravel ini berhasil menerapkan konsep manajemen data kapster dengan memanfaatkan migration, model, controller, dan view secara terstruktur. Program ini menggunakan tiga pendekatan akses database, yaitu Raw SQL, Query Builder, dan Eloquent ORM, sehingga menunjukkan perbedaan cara kerja serta fleksibilitas Laravel dalam mengelola data. Relasi antar tabel seperti admin, outlet, shift, dan kapster telah diatur menggunakan foreign key sehingga integritas data tetap terjaga. Alur kerja aplikasi dimulai dari input data melalui form, diproses oleh controller, disimpan ke database, lalu ditampilkan kembali dalam bentuk tabel pada halaman view. Dengan demikian, program ini sudah mencerminkan implementasi CRUD dasar yang sesuai dengan konsep MVC pada framework Laravel.

### **3.2 Saran**

Untuk pengembangan selanjutnya, aplikasi ini dapat ditingkatkan dengan menambahkan validasi input menggunakan fitur validation Laravel agar data yang masuk lebih aman dan konsisten. Selain itu, disarankan untuk menerapkan sistem autentikasi sehingga nilai admin\_id tidak ditulis statis, melainkan diambil dari admin yang sedang login. Penggunaan seeder juga sebaiknya disesuaikan dengan struktur tabel agar tidak terjadi error duplikasi atau ketidaksesuaian kolom. Dari sisi antarmuka, tampilan dapat diperbaiki menggunakan Blade layout dan CSS framework seperti Bootstrap agar lebih responsif dan user-friendly. Terakhir, fitur update dan delete data kapster dapat ditambahkan agar sistem manajemen data menjadi lebih lengkap dan optimal.