

# MATERI PEMBELAJARAN

## MVC (MODEL VIEW CONTROLLER) PADA LARAVEL

*Mata Pelajaran: Rekayasa Perangkat Lunak*

### A. Pengertian MVC

MVC adalah singkatan dari Model, View, dan Controller. MVC merupakan pola arsitektur yang digunakan Laravel untuk mengatur struktur aplikasi agar kode lebih rapi, terorganisir, dan mudah dikembangkan.

- Model (M): mengelola data dan berhubungan dengan database.
- View (V): menampilkan informasi atau antarmuka kepada pengguna.
- Controller (C): mengatur logika aplikasi dan menjadi penghubung antara Model dan View.

Alur sederhana MVC Laravel adalah:

```
User → Route → Controller → Model → Database → Controller → View → User
```

### B. Model

Model adalah bagian yang bertugas untuk berhubungan dengan database. Model dapat digunakan untuk mengambil, menyimpan, mengubah, dan menghapus data.

Lokasi Model:

```
app/Models
```

Membuat Model Bunga:

```
php artisan make:model Bunga
```

Contoh Model:

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Bunga extends Model
{
    protected $fillable = [
        'nama',
        'harga',
        'stock'
    ];
}
```

Properti \$fillable digunakan untuk menentukan kolom yang dapat diisi secara massal menggunakan metode create() atau update().

## C. View

View adalah bagian yang bertugas menampilkan data kepada pengguna. View dapat berisi HTML, CSS, JavaScript, dan Blade Template.

Lokasi View:

```
resources/views
```

Laravel menggunakan Blade Template dengan ekstensi `.blade.php`, misalnya:

```
resources/views/bunga/index.blade.php
```

Contoh menampilkan variabel:

```
// Controller
$nama = "Mawar";
return view('bunga.index', compact('nama'));

// View
<h1>{{ $nama }}</h1>
```

Contoh perulangan menggunakan Blade:

```
@foreach($bungas as $bunga)
    <p>{{ $bunga->nama }}</p>
@endforeach
```

## D. Controller

Controller berfungsi mengatur logika aplikasi serta menjadi penghubung antara Model dan View.

Lokasi Controller:

```
app/Http/Controllers
```

Membuat Controller:

```
php artisan make:controller BungaController
```

Contoh Controller:

```
<?php

namespace App\Http\Controllers;

use App\Models\Bunga;

class BungaController extends Controller
{
    public function index()
    {
        $bungas = Bunga::all();

        return view('bunga.index', compact('bungas'));
    }
}
```

## E. Route

Route digunakan untuk menghubungkan URL dengan Controller. Route web Laravel berada pada file:

```
routes/web.php
```

Contoh Route:

```
use App\Http\Controllers\BungaController;

Route::get('/bunga', [BungaController::class, 'index']);
```

Ketika pengguna membuka URL /bunga, Laravel akan menjalankan method index() pada BungaController.

## F. Alur Kerja MVC Laravel

- User membuka halaman atau mengirimkan permintaan.
- Route menerima dan mengarahkan permintaan.
- Controller menerima permintaan dan menjalankan logika aplikasi.
- Controller menggunakan Model untuk mengambil atau mengolah data.
- Model berhubungan dengan Database.
- Data dikembalikan kepada Controller.
- Controller mengirimkan data ke View.
- View menampilkan hasil kepada User.

```
USER
↓
ROUTE
↓
CONTROLLER
↓
MODEL
↓
DATABASE
↓
CONTROLLER
↓
VIEW
↓
USER
```

## G. Contoh Implementasi MVC: Data Bunga

Contoh tabel database: bungas dengan kolom id, nama, harga, stock, created\_at, dan updated\_at.

### 1. Model Bunga

```
class Bunga extends Model
{
    protected $fillable = [
        'nama',
        'harga',
        'stock'
    ];
}
```

## 2. Controller Bunga

```
public function index()
{
    $bungas = Bunga::all();

    return view('bunga.index', compact('bungas'));
}
```

## 3. Route

```
Route::get('/bunga', [BungaController::class, 'index']);
```

## 4. View

```
<h1>Data Bunga</h1>

<table border="1">
    <tr>
        <th>No</th>
        <th>Nama</th>
        <th>Harga</th>
        <th>Stock</th>
    </tr>

    @foreach($bungas as $bunga)
    <tr>
        <td>{{ $loop->iteration }}</td>
        <td>{{ $bunga->nama }}</td>
        <td>{{ $bunga->harga }}</td>
        <td>{{ $bunga->stock }}</td>
    </tr>
    @endforeach
</table>
```

## H. CRUD Menggunakan MVC

CRUD terdiri dari Create, Read, Update, dan Delete. Laravel menyediakan Resource Controller untuk mempermudah pembuatan fitur CRUD.

```
php artisan make:controller BungaController --resource
```

Resource Controller menyediakan method berikut:

| Method    | Fungsi                       |
|-----------|------------------------------|
| index()   | Menampilkan seluruh data     |
| create()  | Menampilkan form tambah data |
| store()   | Menyimpan data baru          |
| show()    | Menampilkan detail data      |
| edit()    | Menampilkan form edit        |
| update()  | Memperbarui data             |
| destroy() | Menghapus data               |

## I. Contoh Route Resource

```
use App\Http\Controllers\BungaController;

Route::resource('bunga', BungaController::class);
```

Untuk melihat daftar Route Laravel, gunakan perintah:

```
php artisan route:list
```

## J. Contoh Form Tambah Data

```
<form action="{{ route('bunga.store') }}" method="POST">

    @csrf

    <input type="text" name="nama" placeholder="Nama Bunga">
    <input type="number" name="harga" placeholder="Harga">
    <input type="number" name="stock" placeholder="Stock">

    <button type="submit">Simpan</button>

</form>
```

## K. Struktur Folder MVC

```
project-laravel
├── app
│   ├── Http
│   │   └── Controllers
│   │       └── BungaController.php
│   └── Models
│       └── Bunga.php
├── resources
│   └── views
│       └── bunga
│           ├── index.blade.php
│           ├── create.blade.php
│           ├── edit.blade.php
│           └── show.blade.php
├── routes
│   └── web.php
└── database
    └── migrations
```

## L. Perbedaan Model, View, dan Controller

| Bagian     | Fungsi                         | Contoh              |
|------------|--------------------------------|---------------------|
| Model      | Mengelola data dan database    | Bunga.php           |
| View       | Menampilkan antarmuka dan data | index.blade.php     |
| Controller | Mengatur logika aplikasi       | BungaController.php |

## M. Keuntungan Menggunakan MVC

- Kode aplikasi lebih rapi dan terstruktur.
- Memudahkan pengembangan aplikasi.
- Memudahkan proses perbaikan dan pemeliharaan.
- Memudahkan pembagian tugas dalam kerja tim.
- Komponen aplikasi dapat digunakan kembali.

## N. Kesimpulan

MVC merupakan pola arsitektur penting dalam Laravel. Model bertugas mengelola data dan berhubungan dengan database, View bertugas menampilkan data kepada pengguna, sedangkan Controller bertugas mengatur logika dan menghubungkan Model dengan View. Dengan memahami MVC, pengembangan aplikasi Laravel menjadi lebih terstruktur, mudah dipelajari, dan mudah dikembangkan.

## O. Latihan Praktik

Buatlah aplikasi sederhana Data Siswa menggunakan konsep MVC dengan ketentuan:

- Model: Siswa
- Controller: SiswaController
- View: siswa/index.blade.php
- Database: siswas
- Data yang ditampilkan: Nama, NIS, Kelas, dan Alamat.

Tantangan: Kembangkan aplikasi tersebut menjadi CRUD lengkap menggunakan `Route::resource()`.