LARAVEL 5.8

- Documentation : 2
- Sering Digunakan : 2
- Php artisan make: 2
- Menyalakan local server : 2
- Alur MVC laravel : 2
- Templating Engine : 2
- Database (CRUD) : 2
- Migration & Model : 3
- CRUD 3
  1. Read (getById) 3
  2. Create 3
  3. Update 5
  4. Delete 6
- Form Validation & Session (Flashdata) 6

- **Documentation :**
  https://laravel.com/docs/

- **Sering Digunakan :**
  - Blade tidak menggunakan titik koma ;
  - Php echo : <?= ... ?> atau di laravel : {{$nama}}
  - Pemeriksaan : var_dump(); atau di laravel dump(); atau dd();
  - @crsf : dibawah form pembuka

- **Php artisan make:**
  - Controller : php artisan make:controller <namaController>
  - Controller CRUD : php artisan make:controller <namaController> --resource
  - Controller + Model : php artisan make:controller <namaCon> -r -m <namaModel>
  - Jika controller dan model dibuat menggunakan cara diatas (-r -m), routenya bisa disimple kan seperti berikut :

    ```
    use App\Http\Controllers\StudentsController;
    Route::resource('students', StudentsController::class);
    ```

  -

- **Menyalakan local server :**
  1. Buka Git Bash, arahkan directori ke projek : $ cd C:\xampp\htdocs\cobaLaravel
  2. Beri perintah : $ php artisan serve

- **Alur MVC laravel :**
  1. File pertama yg diakses adalah route : cobaLaravel/routes/web.php

     ```
     use App\Http\Controllers\MahasiswaController;
     Route::get('/mahasiswa', [MahasiswaController::class, 'index']);
     ```

  2. Controller : cobaLaravel/app/Http/Controllers
  3. View : cobaLaravel/resources/views

- **Templating Engine :**
  1. main.blade.php :

     ```
     <title>@yield('title')</title>
     @yield('container') //untuk body
     ```

  2. index.blade.php :

     ```
     @extends('layout/main') //atau bisa juga (‘layout.main’)
     @section('title', 'Web Programming UNPAS')
     @section('container')
         {{-- isi content --}}
     @endsection
     ```

- **Database (CRUD) :**
  1. Lokasi Konfigurasi : cobaLaravel/.env
  2. Penggunaan Query Builder, di controller (MahasiswaController) harus menambahkan :

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

- **Migration & Model :**

1. Jika migration error di windows, penanganan ada di video ngobar #18 menit 48:25
2. Tips Pembuatan : migration itu jamak (students), model itu singular (Student) agar ORM-nya langsung konek, Jika tidak menggunakan tips tsb, di class ... extend model tambahkan :

```
Protected $table = 'namaTable';
```

3. Membuat migration : php artisan make:migration create_<namaTableDB>_table
4. Lokasi konfigurasi attribut table : cobaLaravel/database/migrations
5. Membuat model : php artisan make:migration <namaModel>
6. **Pengguanaan ORM:**
   - Di controller mahasiswa :

```
// get All Student
use App\Models\Student;
$mahasiswa = Student::all();
return view('mahasiswa.index', ['mahasiswa' => $mahasiswa]);
```

- **CRUD**

1. **Read (getById)**
   – Route/web.php :

```
// students
use App\Http\Controllers\StudentsController;
Route::get('/students/{student}', [StudentsController::class, 'show']);
```

   – Controller/StudentsController.php :

```
    public function show(Student $student) // detail (getById)
    {
        return view('students.show', compact('student'));
    }
```

2. **Create**
   – View/create.blade.php (form tambah) :

```
// harus menambahkan @csrf dibawah form pembuka
<form method="post" action="/students">
    @csrf
    // tempat : field input, button submit, form group
</form>
```

   – Route/web.php :

```
// students
use App\Http\Controllers\StudentsController;

// harus diatas route {student} agar tidak dikira id
Route::get('/students/create', [StudentsController::class, 'create']);
Route::get('/students/{student}', [StudentsController::class, 'show']);

```

```
// route::post (di form action=”/student”) tanpa tambahan url
Route::post('/students', [StudentsController::class, 'store']);
```

Untuk Tambah Data / Controller Create ada 2 cara, Pertama :

– Controller/StudentsController.php

```
    // function yg menampilkan view (form tambah data)
    public function create()
    {
        return view('students.create');
    }

    // function yg mengeksekusi tambah data
    public function store(Request $request)
    {
        $student = new Student;
        $student->nama = $request->nama;
        $student->nrp = $request->nrp;
        $student->email = $request->email;
        $student->jurusan = $request->jurusan;

        $student->save();
        return redirect('/students'); // harus return redirect();
    }
```

Untuk Tambah Data / Controller Create ada 2 cara, Kedua :

– Controller/StudentsController.php

```
// function yg menampilkan view (form tambah data)
public function create()
{
    return view('students.create');
}

// function yg mengeksekusi tambah data
public function store(Request $request)
{
    Student::create($request->all());
    return redirect('/students');
}
```

– Model/Student.php

```
class Student extends Model
{
    use HasFactory;
    // Pilih salah satu antara :
```

```
    $guarded (untuk yg [tidak boleh diisi seperti id, role_id, dll])
    $fillable (untuk yg [hanya boleh diisi])
    Untuk mencegah hacking

    protected $guarded = ['id'];
    protected $fillable = ['nama', 'nrp', 'email', 'jurusan'];
}
```

3. Update

- View/edit.blade.php

```
<form method="post" action="/students/{{ $student->id }}">
    @csrf
    @method('patch') // mengelabui method menjadi (‘patch’)
</form>
```

- Route/web.php

```
use App\Http\Controllers\StudentsController;

Route::get('/students/{student}/edit', [StudentsController::class, 'edit']);
Route::patch('/students/{student}', [StudentsController::class, 'update']);
```

- Controller/StudentsController.php

```
// function yg menampilkan view (form ubah data)
public function edit(Student $student)
{
    return view('students.edit', compact('student'));
}

// function yg mengeksekusi perubahan data
public function update(Request $request, Student $student)
{
    $request->validate([
        'nama'      => 'required',
        'nrp'       => 'required|size:8',
        'email'     => 'required|email',
        'jurusan'   => 'required'
    ]);

    Student::where('id', $student->id)->update([
        'nama'      => $request->nama,
        'nrp'       => $request->nrp,
        'email'     => $request->email,
        'jurusan'   => $request->jurusan
    ]);
    return redirect('/students')->with('status', ' Diubah!');
}
```

4. **Delete**
   - View/edit.blade.php

```
<form action="/students/{{ $student->id }}" method="post">
    @csrf
    @method('delete') ) // mengelabui method menjadi (‘patch’)
    <button type="submit" class="btn btn-danger">Delete</button>
</form>
```

   - Route/web.php

```
use App\Http\Controllers\StudentsController;

Route::delete('/students/{student}', [StudentsController::class, 'destroy']);
```

   - Controller/StudentsController.php

```
public function destroy(Student $student)
{
    Student::destroy($student->id);
    return redirect('/students')->with('status', 'Berhasil Dihapus!');

}
```

- **Form Validation & Session (Flashdata)**

1. **Session (Flashdata)**
   - Controller/StudentsController.php : Tambahkan with();

```
return redirect('/students')->with('status', 'Berhasil Ditambahkan!');
```

   - View/create.blade.php :

```
@if (session('status'))
    <div class="alert alert-success">
        {{ session('status') }}
    </div>
@endif
```

2. **Form Validation**
   - Controller/StudentsController.php :

```
public function store(Request $request)
{
    // rulesnya :
    $request->validate([
        'nama'       => 'required',
        'nrp'        => 'required|size:8',
        'email'      => 'required|email',
        'jurusan'    => 'required'
    ]);

    Student::create($request->all());
    return redirect('/students')->with('status', 'Ditambahkan!');
}
```

– View/create.blade.php :

```
<div class="form-group">
    <label for="nama">Nama</label>
    // Di <input> tambahkan class is-invalid (untuk border jadi merah)
    <input class="@error('nama') is-invalid @enderror" value="{{ old('nama')
}}">
    // Setelah <input> tambahkan :
    @error('nama') <div class="invalid-feedback">{{ $message }}</div> @enderror
</div>
```