

MODUL PRAKTIKUM PERTEMUAN 12
PROSES CRUD MENGGUNAKAN MVC

- 1. Gunakan project Laravel herd yang sudah untuk mengerjakan materi CRUD menggunakan arsitektur MVC!
- 2. Merancang Model & Migration

❖ Skema database & tabel

Diasumsikan akan merancang sebuah tabel untuk mengelola data buku dengan struktur field sebagai berikut :

No.	Field	Type(Length)	Keterangan
1	id	Int	Primary Key
2	isbn	Varchar	
3	judul	Text	
4	tahun	Varchar	
5	penulis	Varchar	
6	penerbit	Varchar	
7	cover	text	Menyimpan cover buku dalam bentuk file image

❖ Menciptakan Model & Migration

```
php artisan make:model buku -m
```

❖ Konfigurasi Migration Buku

Modifikasi file migration buku sesuai dengan perintah dibawah ini!

```
1. <?php
2. use Illuminate\Database\Migrations\Migration;
3. use Illuminate\Database\Schema\Blueprint;
4. use Illuminate\Support\Facades\Schema;
5.
6. return new class extends Migration
7. {
8.     public function up(): void
9.     {
10.         Schema::create('bukus', function (Blueprint $table) {
11.             $table->id();
12.             $table->string('isbn');
13.             $table->string('judul');
14.             $table->string('tahun');
15.             $table->text('penulis');
16.             $table->string('penerbit');
17.             $table->text('cover');
18.             $table->timestamps();
19.         });
20.     }
21.
22.     public function down(): void
23.     {
24.         Schema::dropIfExists('bukus');
25.     }
26. };
```

❖ Konfigurasi Model Buku

Modifikasi file migration buku sesuai dengan perintah dibawah ini!

```
1. <?php
2. namespace App\Models;
3. use Illuminate\Database\Eloquent\Factories\HasFactory;
4. use Illuminate\Database\Eloquent\Model;
5.
6. class Buku extends Model
7. {
```

```

8.     use HasFactory;
9.     protected $table = 'bukus';
10.    protected $primaryKey = 'id';
11.    protected $fillable = ['id', 'isbn', 'judul', 'tahun', 'penulis',
    'penerbit', 'cover'];
12. }

```

❖ Eksekusi File Migration

Tuliskan perintah dibawah ini pada terminal VsCode untuk menciptakan tabel pada database sesuai dengan migration yang ada!

```
php artisan migrate
```

3. Merancang Controller

❖ Menciptakan Controller Untuk Kelola Buku

Ciptakan file controller baru untuk aktivitas database buku dengan perintah berikut pada terminal VsCode!

```
php artisan make:controller BukuController -r
```

Fungsi “-r” pada perintah diatas yaitu untuk menciptakan resource yang dibutuhkan controller. Resource yang dibutuhkan yaitu : index, create, create, show, store, edit, update & destroy.

Rincian Fungsi resource controller :

- Index : untuk menampilkan semua data pada tabel.
- Create : untuk menampilkan view untuk tambah data pada tabel
- Store : untuk melakukan proses simpan data ke tabel
- Show : untuk menampilkan view berisikan data tertentu pada tabel
- Edit : untuk menampilkan view untuk edit data pada tabel.
- Update : untuk melakukan proses update data tertentu pada tabel
- Destroy : untuk melakukan proses hapus data tertentu pada tabel

❖ Konfigurasi Controller

Silahkan konfigurasi file controller “BukuController.php” sesuai resource yang dibutuhkan!

```

1. <?php
2.
3. namespace App\Http\Controllers;
4.
5. use App\Models\Buku;
6. use Illuminate\Http\Request;
7.
8. class BukuController extends Controller
9. {
10.     public function index()
11.     {
12.         $data['buku'] = Buku::all();
13.         return view('buku.index', $data);
14.     }
15.
16.     public function create()
17.     {
18.         return view('buku.create');
19.     }
20.
21.     public function store(Request $request)
22.     {
23.         $request->validate([
24.             'isbn' => 'required',
25.             'judul' => 'required',
26.             'penerbit' => 'required',
27.             'penulis' => 'required',
28.             'tahun' => 'required',
29.             'cover' => 'required'
30.         ]);

```

```

31.     $cover = $request->file('cover');
32.     $coverName = $cover->hashName();
33.     $request->cover->move(public_path('/cover'), $coverName);
34.     Buku::create([
35.         'isbn' => $request->isbn,
36.         'judul' => $request->judul,
37.         'penulis' => $request->penulis,
38.         'penerbit' => $request->penerbit,
39.         'tahun' => $request->tahun,
40.         'cover' => $coverName
41.     ]);
42.     return redirect()->route('buku.index');
43. }
44.
45. public function show(string $id)
46. {
47.     //
48. }
49. public function edit(string $id)
50. {
51.     $data['buku'] = Buku::find($id);
52.     return view('buku.edit', $data);
53. }
54. public function update(Request $request, string $id)
55. {
56.     $request->validate([
57.         'isbn' => 'required',
58.         'judul' => 'required',
59.         'penerbit' => 'required',
60.         'penulis' => 'required',
61.         'tahun' => 'required'
62.     ]);
63.     $buku = Buku::find($id);
64.     $buku->update([
65.         'isbn' => $request->isbn,
66.         'judul' => $request->judul,
67.         'penulis' => $request->penulis,
68.         'penerbit' => $request->penerbit,
69.         'tahun' => $request->tahun,
70.     ]);
71.     return redirect()->route('buku.index');
72. }
73. public function destroy(string $id)
74. {
75.     $buku = Buku::find($id);
76.     $buku->delete();
77.     unlink(public_path('cover/' . $buku->cover));
78.     return redirect()->route('buku.index');
79. }
80. }

```

4. Merancang Routes

Untuk mempermudah proses perpindahan dari setiap aktivitas mengelola data buku, silahkan tambah route pada file “web.php” yang berada pada folder “routes” sesuai perintah dibawah ini!

```

1. <?php
2.
3. use App\Http\Controllers\BukuController;
4. use Illuminate\Support\Facades\Route;
5.
6. Route::get('/', function () {
7.     return view('welcome');
8. });
9. Route::resource('/buku', BukuController::class);

```

10.

5. Merancang Layout Website

❖ Membuat Layout

Layout pada blade template sangat membantu kita dalam mengatur tata letak konten pada website. Ciptakan file view baru bernama “template.blade.php” melalui terminal VsCode sesuai dengan perintah dibawah ini.

```
php artisan make:view template
```

❖ Modifikasi Template Website

Tuliskan perintah berikut pada file view “template.blade.php” untuk mengelola tata letak view!

```
1. <!doctype html>
2. <html lang="en">
3.
4. <head>
5.     <title>website laravel 11</title>
6.     <meta charset="utf-8" />
7.     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
8.     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet"/>
9. </head>
10.
11. <body>
12.     <header>
13.         <!-- place navbar here -->
14.     </header>
15.     <main>
16.         @yield('content')
17.     </main>
18.     <footer>
19.         <!-- place footer here -->
20.     </footer>
21.     <!-- bootstrap javascript libraries -->
22.     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.min.js">
23.     </script>
24. </body>
25.
26. </html>
```

6. Merancang View

❖ Membuat View Untuk Melihat Semua Data Buku

Ciptakan file view baru dengan nama "index.blade.php" pada folder "buku" melalui terminal VsCode

```
php artisan make:view buku/index
```

dan tuliskan perintah berikut pada file index!

```
1. @extends('template')
2. @section('content')
3.     <div class="container">
4.         <div class="row">
5.             <div class="col-md-12">
6.                 <h2>Halaman Buku</h2>
7.                 <table class="table">
8.                     <thead>
9.                         <tr>
10.                             <th>No</th>
11.                             <th>ISBN</th>
12.                             <th>Judul</th>
13.                             <th>Penulis</th>
14.                             <th>Penerbit</th>
15.                             <th>Tahun</th>
16.                             <th>Cover</th>
17.                             <th>
18.                                 Proses
19.                             </th>
20.                         </tr>
21.                     </thead>
22.                     <tbody>
23.                         @foreach ($buku as $data)
24.                             <tr>
25.                                 <td>{{ $loop->iteration }}</td>
26.                                 <td>{{ $data->isbn }}</td>
```

```

27.         <td>{{ $data->judul }}</td>
28.         <td>{{ $data->penulis }}</td>
29.         <td>{{ $data->penerbit }}</td>
30.         <td>{{ $data->tahun }}</td>
31.         <td>
32.             
34.         </td>
35.         <td>
36.             <form action="{{ route('buku.destroy', $data->id) }}" method="POST">
37.                 @csrf
38.                 @method('DELETE')
39.                 <button type="submit" class="btn btn-danger">Hapus</button>
40.                 <a href="{{ route('buku.edit', $data->id) }}" class="btn btn-info">Ubah</a>
41.             </form>
42.         </td>
43.     </tr>
44. @endforeach
45. </tbody>
46. </table>
47. <a href="{{ route('buku.create') }}" class="btn btn-primary">Tambah</a>
48. </div>
49. </div>
50. </div>
51. @endsection

```

❖ Membuat View Untuk Tambah Data Buku

Ciptakan file view baru dengan nama “create.blade.php” pada folder “buku” melalui terminal VsCode

```
php artisan make:view buku/create
```

dan tuliskan perintah berikut pada file create!

```

1. @extends('template')
2.
3. @section('content')

```

```

4.     <div class="container">
5.         <div class="row">
6.             <div class="col-md-12">
7.                 <h4>Tambah Mahasiswa</h4>
8.                 <form action="{{ route('buku.store') }}" method="post" enctype="multipart/form-data">
9.                     @csrf
10.                    <div class="mb-3">
11.                        <label class="form-label">ISBN</label>
12.                        <input type="text" class="form-control" name="isbn" value="{{ @old('isbn') }}" />
13.                        @error('isbn')
14.                            <small class="form-text text-danger">{{ $message }}</small>
15.                        @enderror
16.                    </div>
17.                    <div class="mb-3">
18.                        <label class="form-label">Judul Buku</label>
19.                        <input type="text" class="form-control" name="judul" value="{{ @old('judul') }}" />
20.                        @error('judul')
21.                            <small class="form-text text-danger">{{ $message }}</small>
22.                        @enderror
23.                    </div>
24.                    <div class="mb-3">
25.                        <label class="form-label">Penulis</label>
26.                        <input type="text" class="form-control" name="penulis" value="{{ @old('penulis') }}" />
27.                        @error('penulis')
28.                            <small class="form-text text-danger">{{ $message }}</small>
29.                        @enderror
30.                    </div>
31.                    <div class="mb-3">
32.                        <label class="form-label">Tahun</label>
33.                        <input type="text" class="form-control" name="tahun" value="{{ @old('tahun') }}" />
34.                        @error('tahun')
35.                            <small class="form-text text-danger">{{ $message }}</small>
36.                        @enderror
37.                    </div>
38.                    <div class="mb-3">
39.                        <label class="form-label">Penerbit</label>

```



```

40.         <input type="text" class="form-control" name="penerbit" value="{{ @old('penerbit') }}" />
41.         @error('penerbit')
42.             <small class="form-text text-danger">{{ $message }}</small>
43.         @enderror
44.     </div>
45.     <div class="mb-3">
46.         <label class="form-label">Cover</label>
47.         <input type="file" class="form-control" name="cover" value="{{ @old('cover') }}" />
48.         @error('cover')
49.             <small class="form-text text-danger">{{ $message }}</small>
50.         @enderror
51.     </div>
52.     <button type="submit" class="btn btn-primary">Simpan</button>
53. </form>
54. </div>
55. </div>
56. </div>
57. @endsection

```

❖ Membuat View Untuk Edit Data Buku

Ciptakan file view baru dengan nama "create.blade.php" pada folder "buku" melalui terminal VsCode

```
php artisan make:view buku/edit
```

dan tuliskan perintah berikut pada file edit!

```

1. @extends('template')
2.
3. @section('content')
4.     <div class="container">
5.         <div class="row">
6.             <div class="col-md-12">
7.                 <h4>Ubah Mahasiswa</h4>
8.                 <form action="{{ route('buku.update', $buku->id) }}" method="post">
9.                     @csrf
10.                    @method('PUT')

```

```

11. <div class="mb-3">
12.     <label class="form-label">ISBN</label>
13.     <input type="text" class="form-control" name="isbn" value="{{ $buku->isbn }}" />
14.     @error('isbn')
15.         <small class="form-text text-danger">{{ $message }}</small>
16.     @enderror
17. </div>
18. <div class="mb-3">
19.     <label class="form-label">Judul Buku</label>
20.     <input type="text" class="form-control" name="judul" value="{{ $buku->judul }}" />
21.     @error('judul')
22.         <small class="form-text text-danger">{{ $message }}</small>
23.     @enderror
24. </div>
25. <div class="mb-3">
26.     <label class="form-label">Penulis</label>
27.     <input type="text" class="form-control" name="penulis" value="{{ $buku->penulis }}" />
28.     @error('penulis')
29.         <small class="form-text text-danger">{{ $message }}</small>
30.     @enderror
31. </div>
32. <div class="mb-3">
33.     <label class="form-label">Tahun</label>
34.     <input type="text" class="form-control" name="tahun" value="{{ $buku->tahun }}" />
35.     @error('tahun')
36.         <small class="form-text text-danger">{{ $message }}</small>
37.     @enderror
38. </div>
39. <div class="mb-3">
40.     <label class="form-label">Penerbit</label>
41.     <input type="text" class="form-control" name="penerbit" value="{{ $buku->penerbit }}" />
42.     @error('penerbit')
43.         <small class="form-text text-danger">{{ $message }}</small>
44.     @enderror
45. </div>
46. <button type="submit" class="btn btn-primary">Simpan</button>

```

```
47.          </form>
48.      </div>
49.  </div>
50. </div>
51. @endsection
```