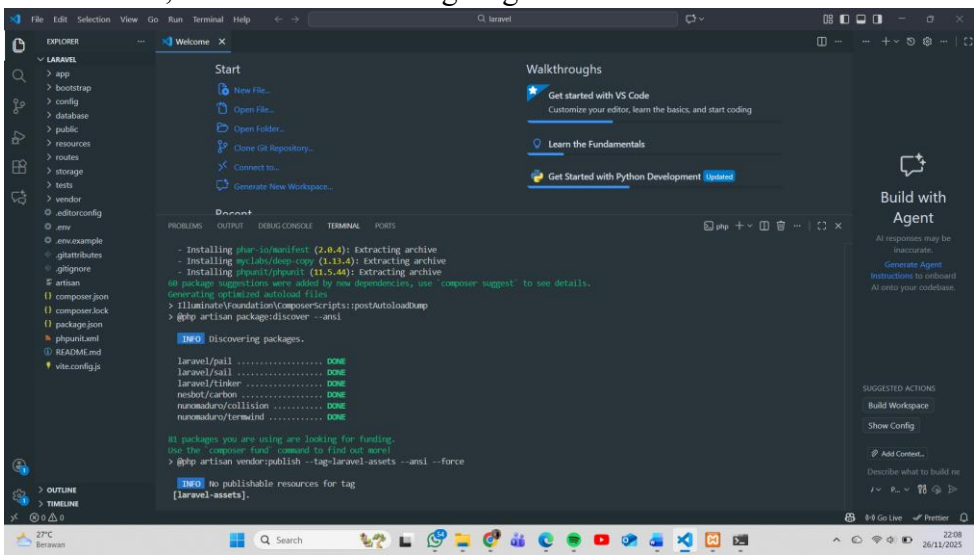
PRAKTIKUM PERTEMUAN 12

Anisa Intania Putri

2024071010

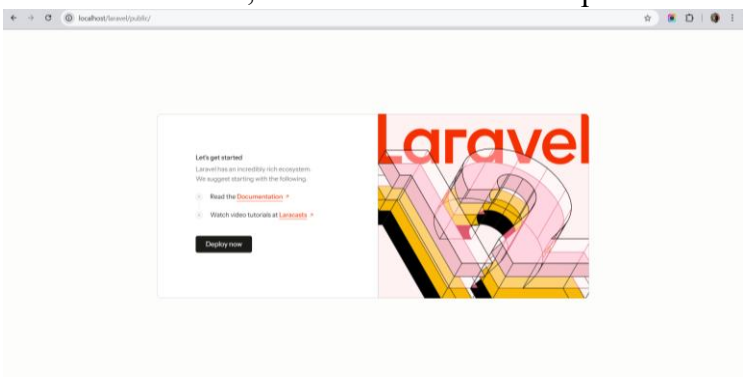
Notes: Gambar tidak ada karena lupa screeshot

1. Persiapkan XAMPP dan VSCode
2. Download Composer di https://getcomposer.org/download
3. Install Composer, Pilih For All User. Klik Next.
4. Tentukan lokasi php yang ada di komputer.
5. Pilih Add php, Klik Next, Install. Klik Finish
6. Untuk memastikan Composer telah terinstal, buka Command prompt, ketikkan composer –v
7. Buka XAMPP, Di bagian Apache, Klik Config, buka PHP.ini. Cari intl. Hilangkan komen “;”. Simpan, Start Apache dan MySQL
8. Buka VS Code, Terminal, New Terminal.
9. Install extension php intelliphense dan laravel blade snippets untuk mempermudah pengodean.
10. Buat folder project baru di htdocs bernama “laravel”
11. Buka folder laravel di VS Code
12. .Di terminal VS Code, pindahkan lokasi ke folder htdocs\laravel dengan command cd C:\xampp\htdocs\laravel
13. Instalasi Laravel dengan mengetikan
14. Jika berhasil, instalasi akan berlangsung



15. Jika error, cek php.ini, bagian zip, uncomment dengan menghapus “;”, lalu restart Apache
16. Isi folder Laravel
17. Command php artisan serve di terminal untuk menyalakan server
18. Buka localhost/laravel/public atau localhost:8000 untuk mengakses halaman lokal laravel

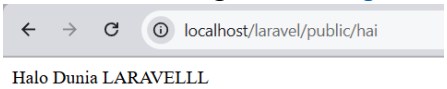
19. Jika error di key generation, ketikkan di terminal: php artisan key:generate
20. Jika error di basis data, ubah basis data dari SQLite ke MySQL di .env. Buat database bernama ‘laravel’ di mysql. Lalu di terminal, ketikan php artisan migrate
21. Jika sudah berhasil, akan muncul halaman depan Laravel



22. Posisi route untuk seluruh web berada di folder routes/web.php
23. Posisi view yang menampung css, js, dan halaman html-php ada di folder resources. Di respources/views, menggunakan blade templating untuk mempermudah pengkodean website
24. Posisi controllers ada di app/Http/Controllers
25. Konfigurasi DB ada di .env
26. .Icon, gambar, dan yang bisa diakses dari luar ditempatkan di folder public
27. Ubah kode program di web.php

```
routes > web.php > ...
<?php

use Illuminate\Support\Facades\Route;

Route::get('/', function () {
    return view('welcome');
});
Route::get('/hai', function () {
    return 'Halo Dunia LARAVELLL';
});

```

28. Cek hasil running di web http://localhost/laravel/public/halo



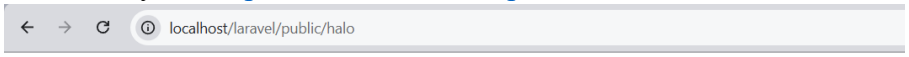
29. Buat halaman baru di folder resources/views
30. Ketikkan tanda seru ! Kemudian, klik untuk membuat kerangka file HTML
31. Tambahkan isi halaman halo.blade.php

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Halo</title>
</head>
<body>
    <h1>HALO SELAMAT DATANG DARI BLADE TEMPLATE</h1>
</body>
</html>
```

32. Arahkan route ke dari /halo ke halaman halo di views dengan

```php
<?php

use Illuminate\Support\Facades\Route;

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

33. Cek hasilnya di http://localhost/laravel/public/halo



HALO SELAMAT DATANG DARI BLADE TEMPLATE

34. Jika ingin membuat view di dalam folder, buat folder halo, di dalamnya, buat file halo

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Halo Halo</title>
</head>
<body>
    <h2>HALO HALO BANDUNG!</h2>
</body>
</html>
```

35. Update route di web.php

```php
<?php

use Illuminate\Support\Facades\Route;

Route::get('/', function () {
    return view('welcome');
});
Route::get('/halo2', function () {
    return view('halo.halo');
});
```

36. Hasil akses di web http://localhost/laravel/public/halo2

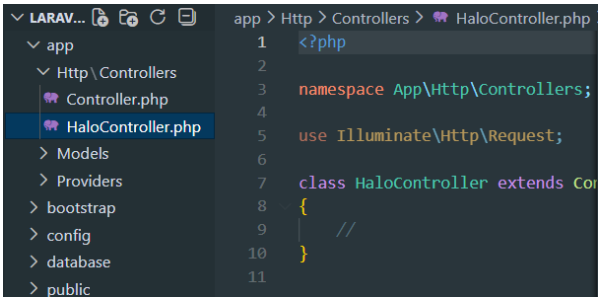
localhost/laravel/public/halo2

HALO HALO BANDUNG!

37. Buat HaloController dengan menerapkan command di terminal

```
PS C:\xampp\htdocs\laravel> php artisan make:controller HaloController

  INFO  Controller [C:\xampp\htdocs\laravel\app\Http\Controllers\HaloController.php] created successfully.
```

38. Hasil HaloController



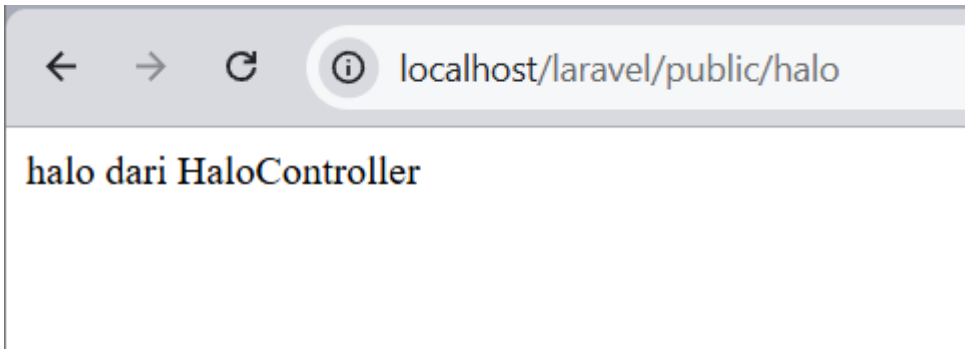
39. Isi dengan

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class HaloController extends Controller
{
    function index() {
        echo('halo dari HaloController');
    }
}
```

40. Ubah web.php menjadi,

```php
<?php

use Illuminate\Support\Facades\Route;

Route::get('/', function () {
    return view('welcome');
});
Route::get('/halo', [App\Http\Controllers\HaloController::class, 'index']);
```

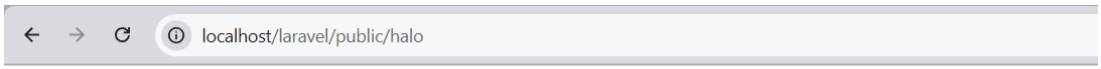
Hasil di http://localhost/laravel/public/halo



41. Jika HaloController mengarahkan ke resouces/views file halo.blade.php

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class HaloController extends Controller
{
    function index() {
        return view('halo');
    }
}
```

Hasilnya:



HALO SELAMAT DATANG DARI BLADE TEMPLATE

42. Tambahkan variable $nama untuk dimuat ke halaman halo

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class HaloController extends Controller
{
    function index() {
        $nama = 'Universitas Pembangunan Jaya';
        return view('halo', ['nama' => $nama]);
    }
}
```

43. Ubah views halo, dengan menambahkan $nama, menjadi

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Halo</title>
</head>
<body>
    <h1>HALO SELAMAT DATANG {{ $nama }}, DARI BLADE TEMPLATE</h1>
</body>
</html>
```

44. Hasil view dengan variable

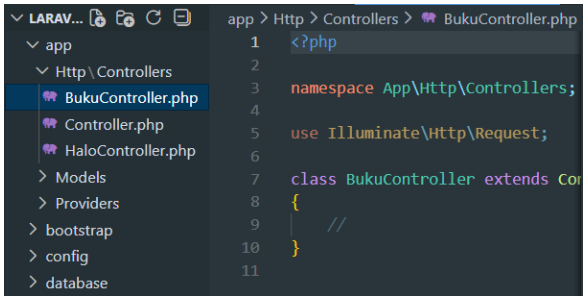
45. Cara lain memasukkan variabel $nama dengan compact di HaloController

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class HaloController extends Controller
{
    function index() {
        $nama = 'Universitas Pembangunan Jaya';
        // return view('halo', ['nama' => $nama]);
        return view('halo', compact('nama'));
    }
}
```

46. Buat BukuController untuk mengatur data buku dengan command php artisan make:controller BukuController

```
PS C:\xampp\htdocs\laravel> php artisan make:controller BukuController

  INFO  Controller [C:\xampp\htdocs\laravel\app\Http\Controllers\BukuController.php] created successfully.
```

Hasilnya:



47. Tambahkan kode program ke BukuController

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class BukuController extends Controller
{
    public function index()
    {
        return view('buku.index');
    }

    public function create()
    {
        //
    }

    public function store(Request $request)
    {
        //
    }

    public function show($id)
    {
        //
    }
    public function edit($id)
    {
        //
    }

    public function update(Request $request, $id)
    {
        //
    }

    public function destroy($id)
    {
        //
    }
}
```

48. Buat folder buku di view dan file index.blade.php
49. Isi halaman view buku/index.blade.php dengan tabel data buku dan form tambah buku

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Data Buku</title>
    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
    <!-- Font Awesome untuk ikon -->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">
    <style>
        .action-buttons {
        white-space: nowrap;
        }
        .action-buttons a {
        color: #000;
        text-decoration:none;
        }
        .table-responsive {
        margin-top: 20px;
        }
        h1 {
        color: #333;
        margin-bottom: 30px;
    }
    </style>
</head>
<body>
     <div class="container mt-5">
        <h1 class="text-center">Daftar Buku</h1>

        <!-- Tabel Data Buku -->
        <div class="table-responsive">
            <table class="table table-striped table-hover">
                <thead class="table-dark">
                    <tr>
                        <th>No</th>
                        <th>Judul Buku</th>
                        <th>Pengarang</th>
                        <th>Tahun Terbit</th>
                        <th>Aksi</th>
                    </tr>
                </thead>
                <tbody>
                    <!-- Data Buku 1 -->
                    <tr>
                        <td>1</td>
                        <td>Pemrograman JavaScript Modern</td>
                        <td>Joko Santoso</td>
                        <td>2022</td>
                        <td class="action-buttons">
                            <button class="btn btn-sm btn-warning">
                                <i class="fas fa-edit"></i> Edit
                        </button>
                        <button class="btn btn-sm btn-danger">
                            <i class="fas fa-trash-alt"></i> Hapus
                        </button>
                    </td>
                </tr>

                <!-- Data Buku 2 -->
                <tr>
                    <td>2</td>
                    <td>Belajar Laravel untuk Pemula</td>
                    <td>Budi Santoso</td>
                    <td>2021</td>
                    <td class="action-buttons">
                        <button class="btn btn-sm btn-warning">
                            <i class="fas fa-edit"></i> Edit
                    </button>
                    <button class="btn btn-sm btn-danger">
                        <i class="fas fa-trash-alt"></i> Hapus
                    </button>
                </td>
            </tr>

        </tbody>
    </table>
</div>

    <!-- Form Tambah Data (opsional) -->
<div class="mt-4 p-3 bg-light rounded">
    <h3>Tambah Buku Baru</h3>
    <form action="" method="post">
        <div class="mb-3">
            <label for="judul" class="form-label">Judul Buku</label>
            <input type="text" class="form-control" id="judul" name="judul" required>
        </div>
        <div class="mb-3">
            <label for="pengarang" class="form-label">Pengarang</label>
            <input type="text" class="form-control" id="pengarang" name="pengarang" required>
        </div>
        <div class="mb-3">
            <label for="tahun_terbit" class="form-label">Tahun Terbit</label>
            <input type="number" class="form-control" id="tahun_terbit" name="tahun_terbit" min="1900" max="2099" required>
        </div>
        <button type="submit" class="btn btn-primary">Simpan</button>
    </form>
</div>

</form>
</div>
</div>

<!-- Bootstrap JS Bundle with Popper -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

50. Lengkapi BukuController dengan fungsi index menampilkan view buku.index

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class BukuController extends Controller
{
    public function index()
    {
        return view('buku.index');
    }
```

51. Tambahkan route di web.php untuk route ke BukuController index

```php
<?php

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

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

Route::get('/buku', [BukuController::class, 'index']);
```

52. Hasil running menampilkan daftar buku dengan tampilan tabel

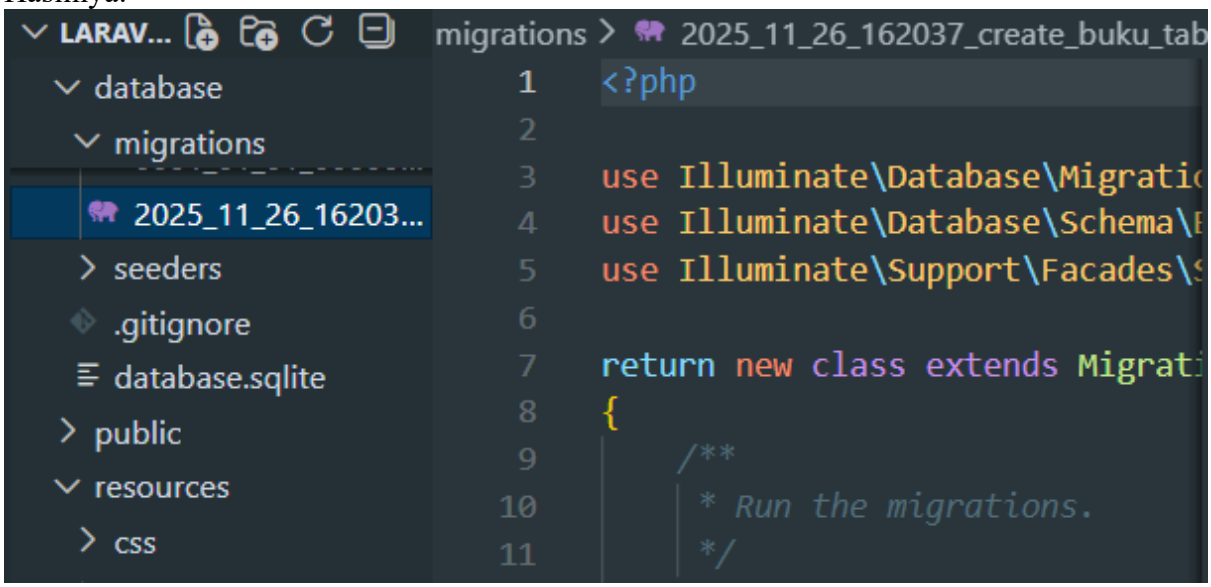
53. Set basis data di .env untuk MySQL dan konfigurasi database

```
.env
23  DB_CONNECTION=mysql
24  DB_HOST=127.0.0.1
25  DB_PORT=3310
26  DB_DATABASE=laravel
27  DB_USERNAME=root
28  DB_PASSWORD=
```

54. Buat migration untuk tabel buku dengan php artisan make:migration create_buku_table

```
PS C:\xampp\htdocs\laravel> php artisan make:migration create_buku_table

  INFO  Migration [C:\xampp\htdocs\laravel\database\migrations\2025_11_26_162037_create_buku_table.php] created successfully.
```

Hasilnya:



55. Isi migration create_buku_table dengan kolom id, judul, pengarang, tahun_terbit, timestamps

```php
public function up(): void
    {
        Schema::create('buku', function (Blueprint $table) {
            $table->id();
            $table->string('judul', '255');
            $table->string('pengarang', '100');
            $table->string('tahun_terbit');
            $table->timestamps();
        });
    }
```

56. Ketikan command berikut, agar tabel buku dibuat di database

```
PS C:\xampp\htdocs\laravel> php artisan make:migration c

PS C:\xampp\htdocs\laravel> php artisan make:migration cPS C:\xampp\htdocs\la
ion create_buku_table
php artisan migrate

  WARN  The database 'laravel' does not exist on the 'mysql' connection.

  Would you like to create it? (yes/no) [yes]
  ❯ yes

  INFO  Preparing database.

  Creating migration table  21.13ms DONE
  0001_01_01_000001_create_cache_table  14.97ms DONE
  0001_01_01_000002_create_jobs_table  59.96ms DONE
  2025_11_26_162037_create_buku_table  7.59ms DONE
```

Hasilnya:

Server: 127.0.0.1:3310 » Database: laravel » Tabel: buku

Jelajahi | Struktur | SQL | Cari | Tambahkan

Struktur tabel | Tampilan hubungan

	#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai
☐	1	**id**	bigint(20)		UNSIGNED	Tidak
☐	2	**judul**	varchar(255)	utf8mb4_unicode_ci		Tidak
☐	3	**pengarang**	varchar(100)	utf8mb4_unicode_ci		Tidak
☐	4	**tahun_terbit**	varchar(255)	utf8mb4_unicode_ci		Tidak
☐	5	**created_at**	timestamp			Ya
☐	6	**updated_at**	timestamp			Ya

57. Buat Seeder BukuSeeder untuk mengisi data di table buku dengan command

```
database
  migrations
    0001_01_01_00000...
    2025_11_26_16203...
  seeders
    BukuSeeder.php
    DatabaseSeeder.php
  .gitignore
  database.sqlite
public
```

```php
<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB;

class BukuSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
```

58. Tambahkan library Facades\DB, isi BukuSeeder sesuai isi tabel

```php
<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB;

class BukuSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        $buku = [
            [
                'judul' => 'Laskar Pelangi',
                'pengarang' => 'Andrea Hirata',
                'tahun_terbit' => 2005,
                'created_at' => now(),
                'updated_at' => now(),
            ],
            [
                'judul' => 'Bumi Manusia',
                'pengarang' => 'Pramoedya Ananta Toer',
                'tahun_terbit' => 1980,
                'created_at' => now(),
                'updated_at' => now(),
            ],
            [
                'judul' => 'Pemrograman JavaScript',
                'pengarang' => 'Zainal Arifin',
                'tahun_terbit' => 2024,
                'created_at' => now(),
                'updated_at' => now(),
            ],
        ];

        DB::table('buku')->insert($buku);
    }
}
```

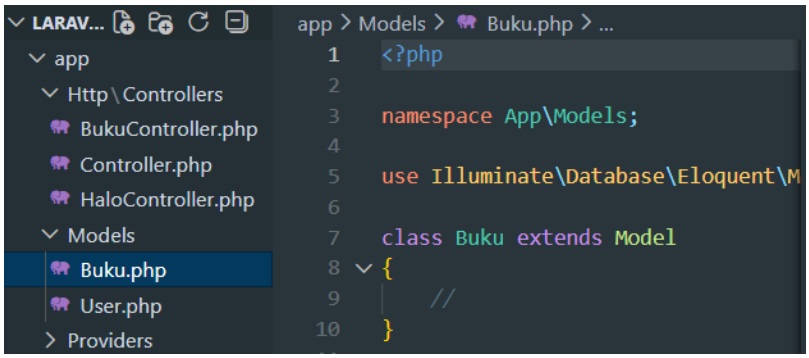
59. Ketikan command untuk memasukan data di BukuSeeder ke Table buku

```
PS C:\xampp\htdocs\laravel> php artisan db:seed --class=BukuSeeder
```

id	judul	pengarang	tahun_terbit	created_at	updated_at
1	Laskar Pelangi	Andrea Hirata	2005	2025-11-26 16:34:36	2025-11-26 16:34:36
2	Bumi Manusia	Pramoedya Ananta Toer	1980	2025-11-26 16:34:36	2025-11-26 16:34:36
3	Pemrograman JavaScript	Zainal Arifin	2024	2025-11-26 16:34:36	2025-11-26 16:34:36

60. Buat model Buku dengan php artisan make:model Buku

```
  INFO  Model [C:\xampp\htdocs\laravel\app\Models\Buku
.php] created successfully.

PS C:\xampp\htdocs\laravel>
```



61. Isi model Buku dengan protected $table dan $fillable fields

```php
class Buku extends Model
{
    protected $table = 'buku';

    protected $fillable = [
        'judul',
        'pengarang',
        'tahun_terbit',
    ];
}
```

62. Tambahkan fungsi index di BukuController untuk mengambil semua data buku dan tampilkan di view

```php
<?php

namespace App\Http\Controllers;

use App\Models\Buku;
use Illuminate\Http\Request;

class BukuController extends Controller
{
    public function index()
    {
        $buku = Buku::all();
        return view('buku.index', compact('buku'));
    }
```

63. Ubah view buku/index.blade.php menggunakan foreach untuk menampilkan buku dari database

```php
@foreach ($buku as $item)
                <tr>
                    <td>{{ $loop->iteration }}</td>
                    <td>{{ $item->judul }}</td>
                    <td>{{ $item->pengarang }}</td>
                    <td>{{ $item->tahun_terbit }}</td>
                    <td class="action-buttons">
                        <button class="btn btn-sm btn-warning">
                            <i class="fas fa-edit"></i> Edit
                        </button>
                        <button class="btn btn-sm btn-danger">
                            <i class="fas fa-trash-alt"></i> Hapus
                        </button>
                    </td>
                </tr>
            @endforeach
```

Hasilnya:

64. Modifikasi route di web.php buat route index dan route store buku

```
Route::get('/buku',  [BukuController::class, 'index'])->name('buku.index');
Route::post('/buku', [BukuController::class, 'store'])->name('buku.store');
```

65. Tambahkan alamat form action di view buku/index.blade.php untuk menyimpan data

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

66. Tambahkan fungsi store di BukuController untuk validasi input dan simpan data buku

```
public function store(Request $request)
    {
                $validated = $request->validate([
            'judul'        => 'required|string|min:3',
            'pengarang'    => 'required|string|min:3',
            'tahun_terbit' => 'required|integer',
        ]);

        Buku::create($validated);

        return redirect()
            ->route('buku.index')
            ->with('success', 'Buku berhasil ditambahkan.');
    }
```

67. Tambahkan route edit buku di web.php

```
Route::get('/buku/{id}/update',  [BukuController::class, 'store'])->name('buku.update');
```

68. Tambahkan link edit di bawah button Edit di buku\index

```
<button class ="btn btn-sm btn-warning">
    <a href="{{ route('buku.edit', ['id' => $item->id]) }}">
    <i class="fas fa-edit"></i> Edit
    </a>
</button>
```

69. Tambahkan fungsi edit di BukuController menampilkan buku dengan id tertentu

```
public function edit($id)
    {
        $buku       = Buku::all();
        $bukuDetail = Buku::findOrFail($id);

        return view('buku.index', compact('buku', 'bukuDetail'));
    }
```

70. Edit view di buku\index sehingga bisa memuat ‘edit’

```
<!-- Form Tambah/Edit Data -->
<div class="mt-4 p-3 bg-light rounded">
    <h3>{{ isset($bukuDetail) ? 'Edit Buku' : 'Tambah Buku Baru' }}</h3>
```

71. Tambahkan route di web.php untuk mengupdate data

```
Route::get('/buku/{id}/update',  [BukuController::class, 'store'])->name('buku.update');
```

72. Tambahkan di view buku\index sehingga bisa mengarahkan ke buku.update

```
<!-- Form yang dapat digunakan untuk tambah dan edit -->
<form action="{{ isset($bukuDetail) ? route('buku.update', ['id' => $bukuDetail->id]) : route('buku.store') }}" method="post">
    @csrf
    @if(isset($bukuDetail))
        @method('PUT')
    @endif
```

73. Tambahkan fungsi update di BukuController untuk update data buku

```
public function update(Request $request, $id)
{
     $validated = $request->validate([
        'judul'        => 'required|string|min:3',
        'pengarang'    => 'required|string|min:3',
        'tahun_terbit' => 'required|integer',
    ]);
    Buku::where('id', $id)->update($validated);
    return redirect()->route('buku.index')->with('success', 'Buku berhasil diubah.');
}
```

74. Tambahkan input value lama di form edit view buku

```
<div class="mb-3">
    <label for="judul" class="form-label">Judul Buku</label>
    <input type="text" class="form-control" id="judul" name="judul"
           value="{{ old('judul', $bukuDetail->judul ?? '') }}" required>
</div>

<div class="mb-3">
    <label for="pengarang" class="form-label">Pengarang</label>
    <input type="text" class="form-control" id="pengarang" name="pengarang"
           value="{{ old('pengarang', $bukuDetail->pengarang ?? '') }}" required>
</div>

<div class="mb-3">
    <label for="tahun_terbit" class="form-label">Tahun Terbit</label>
    <input type="number" class="form-control" id="tahun_terbit" name="tahun_terbit"
           min="1900" max="2099" value="{{ old('tahun_terbit', $bukuDetail->tahun_terbit ?? '') }}" required>
</div>
```

75. Tambahkan route delete buku di web.php

```
Route::delete('/buku/{id}/delete', [BukuController::class, 'destroy'])->name('buku.destroy');
```

76. Tambahkan tombol delete di view buku/index

```
<form action="{{ route('buku.destroy', ['id' => $item->id]) }}" method="POST" style="display: inline;">
   @csrf
   @method('DELETE')
   <button type="submit" class="btn btn-sm btn-danger" onclick="return confirm('Apakah Anda yakin ingin menghapus buku ini?')">
      <i class="fas fa-trash-alt"></i> Hapus
   </button>
</form>
```

77. Tambahkan fungsi destroy di BukuController untuk menghapus buku

```
public function destroy($id)
    {
        $bukuDetail = Buku::findOrFail($id);
        $bukuDetail->delete();

        return redirect()->route('buku.index')
                         ->with('success', 'Buku berhasil dihapus.');
    }
```

78. Cek hasil hapus data buku

79. Cek hasil tambah data buku

80. Cek hasil Edit data buku