

Aplikasi Buku Tamu dengan Laravel

Spesifikasi Project

1. Halaman landing
2. Halaman isi komentar dari pengunjung/tamu
3. Halaman melihat komentar dari para pengunjung/tamu
4. Koneksi database menggunakan MySQL
5. Membuat database bernama belajar-laravel
6. Terdapat sebuah tabel bernama buktamu (id, nama_pengunjung, komentar)

Prasyarat

1. Sudah memiliki PHP Development Stack (Laragon, XAMPP, HERD + DBNGIN)
2. PHP 8.2 / PHP 8.3
3. Database MySQL Server
4. Code Editor Visual Code
5. MySQL Connection Extension di Visual Code

Langkah-langkah

Membuat Database Baru belajar_laravel

Menggunakan perintah sql dari command mysql

```
CREATE DATABASE belajar_laravel CHARACTER SET utf8mb4 COLLATE  
utf8mb4_general_ci;
```

Atau kalian bisa menggunakan phpmyadmin dengan spesifikasi

Belajar_laravel

Charset utf8mb4

Collate utf8mb4_general_ci

Membuat Project Laravel bernama belajar_laravel

Dengan console

```
$ laravel new belajar-laravel  
$ cd belajar-laravel
```

Dengan composer

```
$ composer create-project laravel/laravel belajar-laravel  
$ cd belajar-laravel
```

Edit Environment Development Laravel

Memberitahu Laravel bahwa kita menggunakan database mysql, nama database, username, dan password di mysql

[project_laravel]/.env

```
DB_CONNECTION=mysql  
DB_HOST=127.0.0.1  
DB_PORT=3306  
DB_DATABASE=belajar_laravel  
DB_USERNAME=root  
DB_PASSWORD=rahasia
```

Migrasi Database

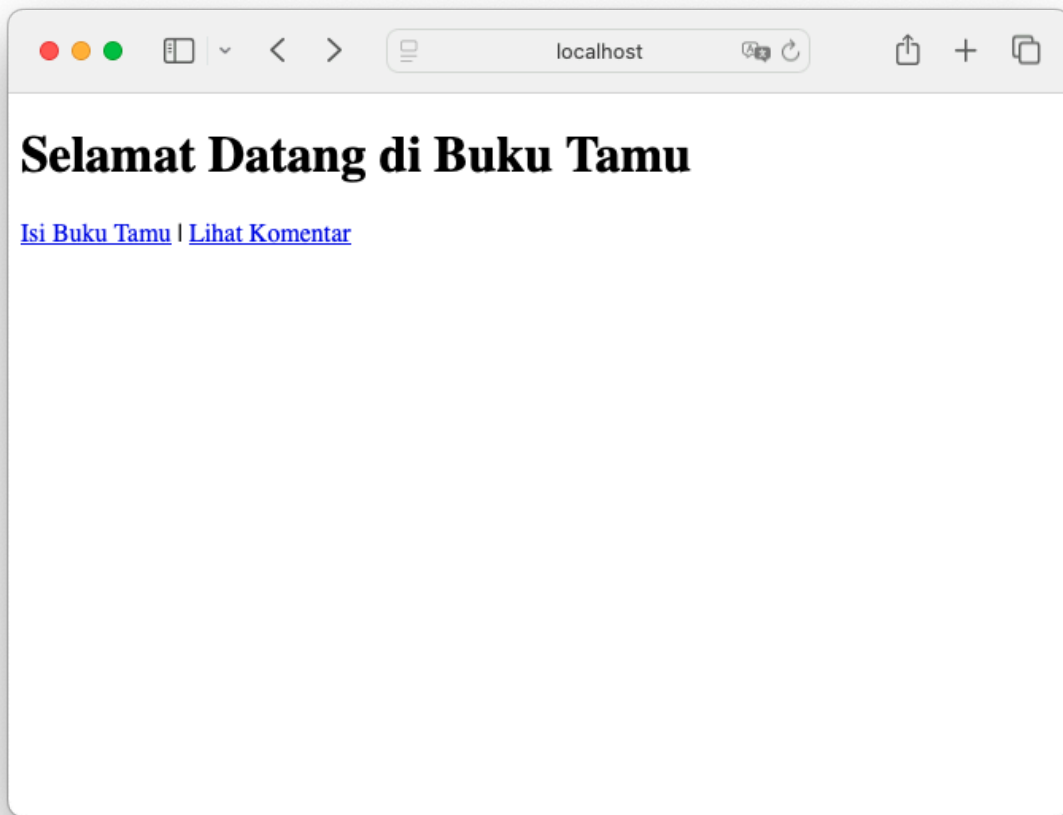
```
belajar-laravel$ php artisan migrate
```

Buka Aplikasi Laravel dari Browser

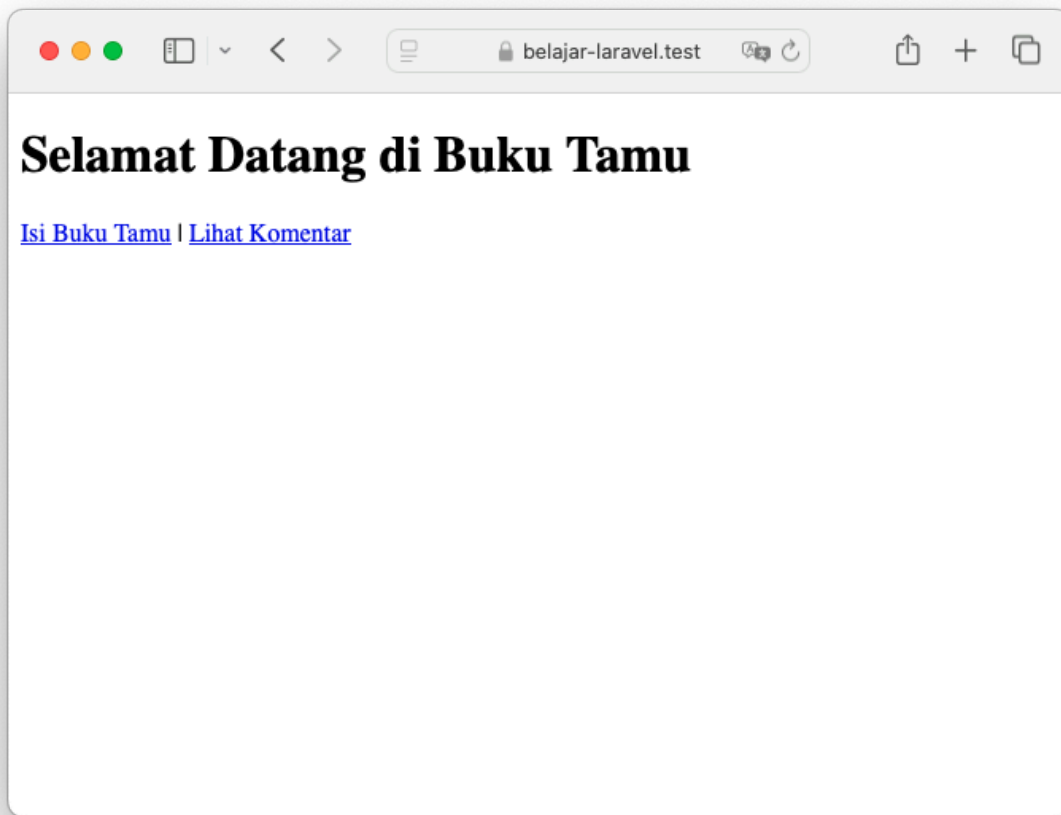
Langsung dari folder project dengan php web server

```
belajar-laravel$ php artisan serve  
  
INFO  Server running on [http://127.0.0.1:8000].  
  
Press Ctrl+C to stop the server
```

Buka browser <http://localhost:8000>



Atau jika menggunakan laragon setelah reload nginx atau herd bisa langsung akses ke browser <http://belajar-laravel.test>



Membuat tabel bukutamu

Buat bukutamu dengan migration

```
belajar-laravel$ php artisan make:migration create_bukutamu_table
```

Buka file migration yang baru dibuat di lokasi database/migrations/

Kemudian modifikasi metode up

```
public function up()
{
    Schema::create('bukutamu', function (Blueprint $table) {
        $table->id();
        $table->string('nama_pengunjung');
```

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

Jalankan migration

```
belajar-laravel$ php artisan migrate
```

Membuat Model BukuTamu

Buat Model dengan make:model

```
belajar-laravel$ php artisan make:model BukuTamu
```

Model ini digunakan untuk berinteraksi dengan tabel bukutamu di database. Tidak perlu modifikasi khusus jika mengikuti aturan penamaan laravel.

Membuat Controller BukuTamu

Buat Controller dengan make:controller

```
belajar-laravel$ php artisan make:controller BukuTamuController
```

Buka file app/Http/Controllers/BukuTamuController.php dan tambahkan metode berikut:

```
<?php  
  
namespace App\Http\Controllers;  
  
use Illuminate\Http\Request;  
// Menggunakan Model BukuTamu atau menggunakan tabel bukutamu  
use App\Models\BukuTamu;  
  
class BukuTamuController extends Controller  
{  
    // Menampilkan halaman landing page  
    public function index()  
    {
```

```

        return view('welcome');
    }

    // Menampilkan formulir buku tamu
    public function create()
    {
        return view('bukutamu.create');
    }

    // Menyimpan data buku tamu
    public function store(Request $request)
    {
        // Validasi input
        $request->validate([
            'nama_pengunjung' => 'required|max:255',
            'komentar' => 'required',
        ]);

        // Simpan data ke database
        BukuTamu::create([
            'nama_pengunjung' => $request->nama_pengunjung,
            'komentar' => $request->komentar,
        ]);

        // Redirect ke halaman melihat komentar
        return redirect()->route('bukutamu.show');
    }

    // Menampilkan komentar dari para tamu
    public function show()
    {
        $data = BukuTamu::latest()->get();
        return view('bukutamu.show', compact('data'));
    }
}

```

Mengatur Mass Assignment pada Model

Properti **\$fillable** pada model digunakan untuk melindungi **mass assignment**, yaitu mekanisme di mana Anda dapat mengisi beberapa atribut dari model secara sekaligus menggunakan metode seperti **create()**, **update()**, atau **fill()**. Dengan **\$fillable**, kita menentukan atribut-atribut mana yang boleh diisi melalui mass assignment.

Buka file app/Models/BukuTamu.php dan tambahkan property \$fillable

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class BukuTamu extends Model
{
    use HasFactory;

    protected $table = 'bukutamu';

    protected $fillable = [
        'nama_pengunjung',
        'komentar',
    ];
}
```

Menyiapkan Routing

Buka file routes/web.php dan tambahkan route berikut:

```
<?php

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

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

Route::get('/', [BukuTamuController::class, 'index'])->name('home');
Route::get('/bukutamu', [BukuTamuController::class,
    'create'])->name('bukutamu.create');
Route::post('/bukutamu', [BukuTamuController::class,
    'store'])->name('bukutamu.store');
Route::get('/bukutamu/komentar', [BukuTamuController::class,
    'show'])->name('bukutamu.show');
```

Membuat View/Tampilan Web dengan Blade

Halaman landing page (welcome.blade.php)

```
<!DOCTYPE html>
<html>
<head>
    <title>Selamat Datang</title>
</head>
<body>
    <h1>Selamat Datang di Buku Tamu</h1>
    <a href="{{ route('bukutamu.create') }}">Isi Buku Tamu</a> |
    <a href="{{ route('bukutamu.show') }}">Lihat Komentar</a>
</body>
</html>
```

Halaman isi Buku Tamu (bukutamu/create.blade.php)

```
<!DOCTYPE html>
<html>
<head>
    <title>Isi Buku Tamu</title>
</head>
<body>
    <h1>Isi Buku Tamu</h1>

    @if ($errors->any())
        <div style="color:red;">
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif

    <form action="{{ route('bukutamu.store') }}" method="POST">
        @csrf
        <label for="nama_pengunjung">Nama:</label><br>
        <input type="text" id="nama_pengunjung" name="nama_pengunjung"
value="{{ old('nama_pengunjung') }}"><br><br>
        <label for="komentar">Komentar:</label><br>
```



```

        <textarea id="komentar" name="komentar">{{ old('komentar')
    }}</textarea><br><br>
        <button type="submit">Kirim</button>
    </form>

    <br>
    <a href="{{ route('home') }}">Kembali ke Halaman Utama</a>
</body>
</html>

```

Halaman Lihat Komentar (bukutamu/show.blade.php)

```

<!DOCTYPE html>
<html>
<head>
    <title>Komentar Buku Tamu</title>
</head>
<body>
    <h1>Komentar dari Para Tamu</h1>

    @if ($data->isEmpty())
        <p>Belum ada komentar.</p>
    @else
        @foreach ($data as $item)
            <div style="border:1px solid #ccc; padding:10px;
margin-bottom:10px;">
                <strong>{{ $item->nama_pengunjung }}</strong> berkata:<br>
                <p>{{ $item->komentar }}</p>
                <small>Waktu: {{ $item->created_at }}</small>
            </div>
        @endforeach
    @endif

    <br>
    <a href="{{ route('home') }}">Kembali ke Halaman Utama</a>
</body>
</html>

```

Jalankan Aplikasi di Browser Kalian!

Buat Laporan

1. Tampilkan screenshot dari aplikasi yang dikerjakan
2. Berikan kesimpulan dan saran untuk pengembangan lebih lanjut
3. Link github aplikasi kalian