

LAPORAN PRAKTIKUM 6
PEMROGRAMAN BERBASIS WEB - A



Mahasiswa:

Ihsan Adi Putra (4522210085)

Dosen:

Adi Wahyu Pribadi, S.Si., M.Kom

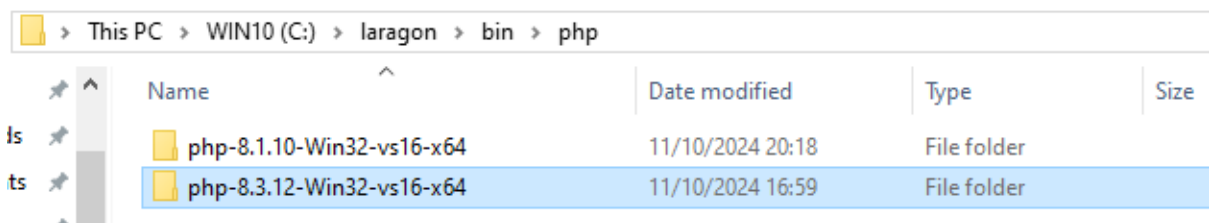
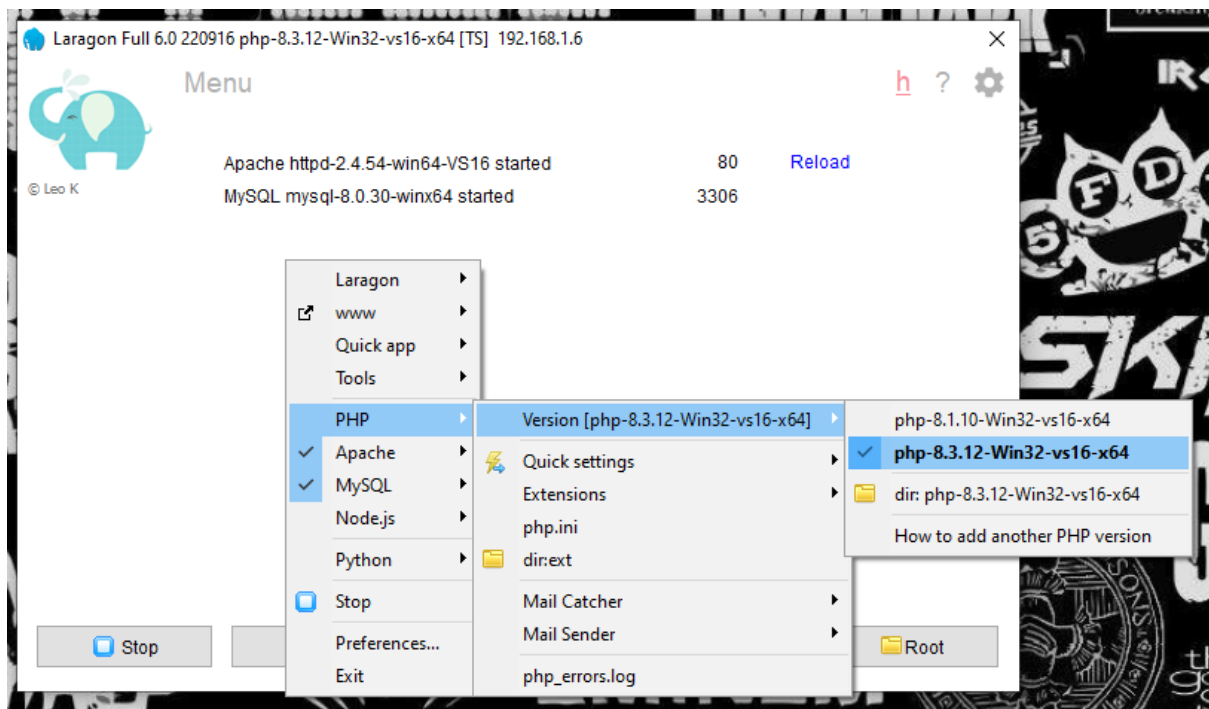
TEKNIK INFORMATIKA
UNIVERSITAS PANCASILA

2024

Aplikasi Buku Tamu dengan Laravel

Mengunduh Laragon

Versi PHP saya sebelumnya adalah 8.1, karna yang dibutuhkan PHP 8.2 atau 8.3, maka saya mengunduh PHP versi 8.3. Cara mengubahnya, copy file PHP 8.3 yang sudah di unduh, lalu buka laragon, klik menu/klik kanan di halaman putih mana saja, lalu pilih PHP>Version>dir. Lalu klik dir “php” atau cari folder laragon>bin>php, lalu paste dan ekstrak all, hapus file zipnya. Setelah itu, klik menu/klik kanan di halaman putih mana saja lagi, lalu pilih PHP>Version>pilih atau ceklis php 8.3.



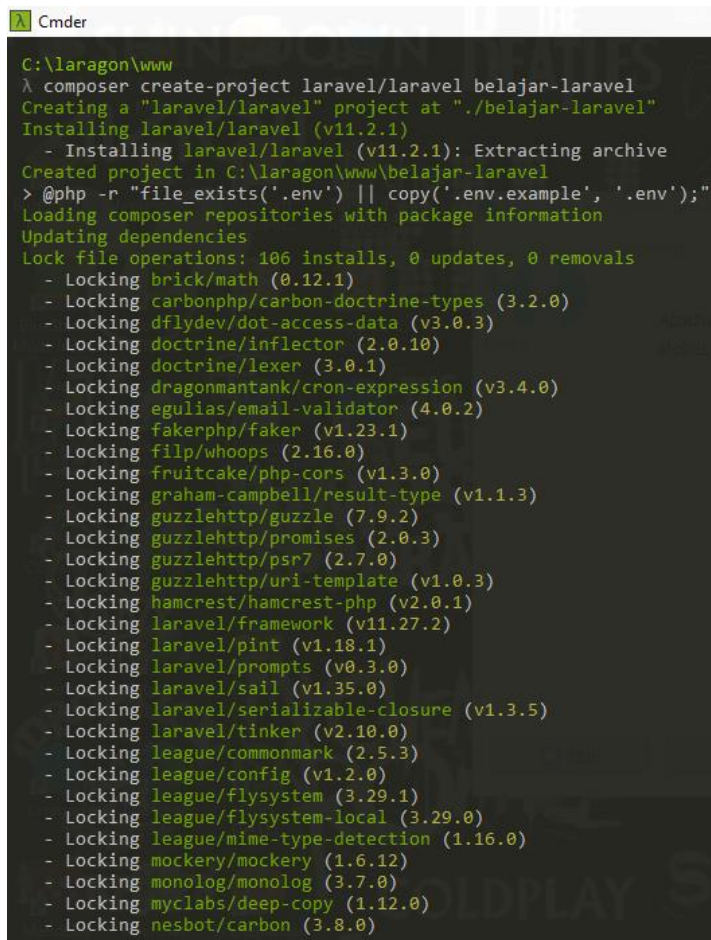
Membuat File belajar-laravel

Klik terminal pada Laragon, lalu ketik:

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

Tunggu. Jika sudah selesai, ketik:

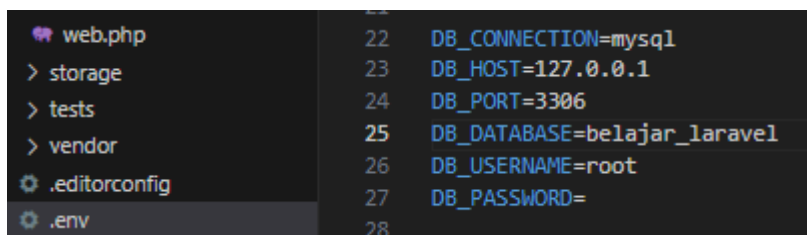
```
$ cd belajar-laravel
```



```
C:\laragon\www
λ composer create-project laravel/laravel belajar-laravel
Creating a "laravel/laravel" project at "../belajar-laravel"
Installing laravel/laravel (v11.2.1)
- Installing laravel/laravel (v11.2.1): Extracting archive
Created project in C:\laragon\www\belajar-laravel
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 106 installs, 0 updates, 0 removals
- Locking brick/math (0.12.1)
- Locking carbonphp/carbon-doctrine-types (3.2.0)
- Locking dflydev/dot-access-data (v3.0.3)
- Locking doctrine/inflector (2.0.10)
- Locking doctrine/lexer (3.0.1)
- Locking dragonmantank/cron-expression (v3.4.0)
- Locking egulias/email-validator (4.0.2)
- Locking fakerphp/faker (v1.23.1)
- Locking filp/whoops (2.16.0)
- Locking fruitcake/php-cors (v1.3.0)
- Locking graham-campbell/result-type (v1.1.3)
- Locking guzzlehttp/guzzle (7.9.2)
- Locking guzzlehttp/promises (2.0.3)
- Locking guzzlehttp/psr7 (2.7.0)
- Locking guzzlehttp/uri-template (v1.0.3)
- Locking hamcrest/hamcrest-php (v2.0.1)
- Locking laravel/framework (v11.27.2)
- Locking laravel/pint (v1.18.1)
- Locking laravel/prompts (v0.3.0)
- Locking laravel/sail (v1.35.0)
- Locking laravel/serializable-closure (v1.3.5)
- Locking laravel/tinker (v2.10.0)
- Locking league/commonmark (2.5.3)
- Locking league/config (v1.2.0)
- Locking league/flysystem (3.29.1)
- Locking league/flysystem-local (3.29.0)
- Locking league/mime-type-detection (1.16.0)
- Locking mockery/mockery (1.6.12)
- Locking monolog/monolog (3.7.0)
- Locking myclabs/deep-copy (1.12.0)
- Locking nesbot/carbon (3.8.0)
```

Buka visual studio code, lalu Open Folder, cari laragon>www>pilih folder “belajar-laravel”.

Edit Environment Development Laravel: Cari File .env, ubah menjadi seperti ini:



```
22 DB_CONNECTION=mysql
23 DB_HOST=127.0.0.1
24 DB_PORT=3306
25 DB_DATABASE=belajar_laravel
26 DB_USERNAME=root
27 DB_PASSWORD=
28
```

Buat tabel bukutamu dengan migration

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

Modifikasi metode up: Buka file migration yang baru dibuat di lokasi:

database/migrations/2024_10_11_144705_create_bukutamu_table.php. ubah menjadi seperti ini.

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

Buat Model BukuTanu dengan make:model

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

Buat Controller BukuTamu 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

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 Lihat Komentar (bukutamu/show.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>

```

Pada file app/Providers/AppServiceProvider.php, ubah menjadi seperti ini:

```

<?php

namespace App\Providers;

use Illuminate\Support\ServiceProvider;
use Illuminate\Support\Facades\Schema;
class AppServiceProvider extends ServiceProvider
{
    /**
     * Register any application services.
     *
     * @return void
     */
    public function register(): void
    {
        //
    }

    /**
     * Bootstrap any application services.
     *
     * @return void
     */
    public function boot()
    {

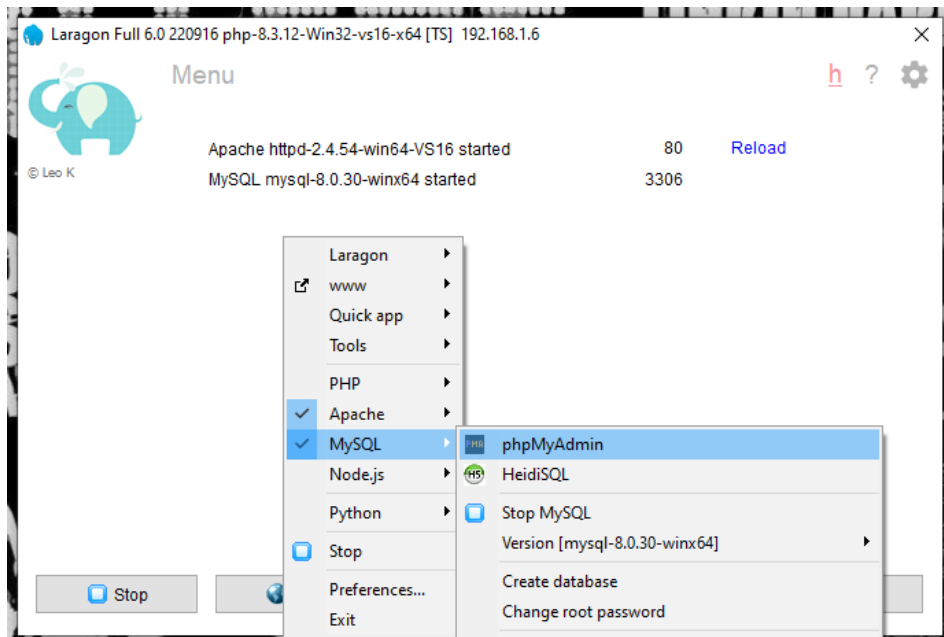
```



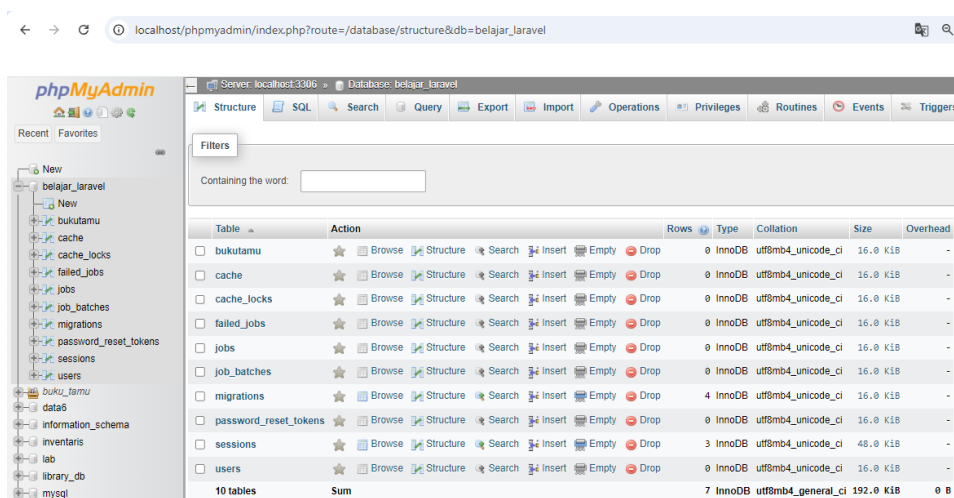
```
Schema::defaultStringLength(191);
}
}
```

Membuat Database Baru belajar_laravel

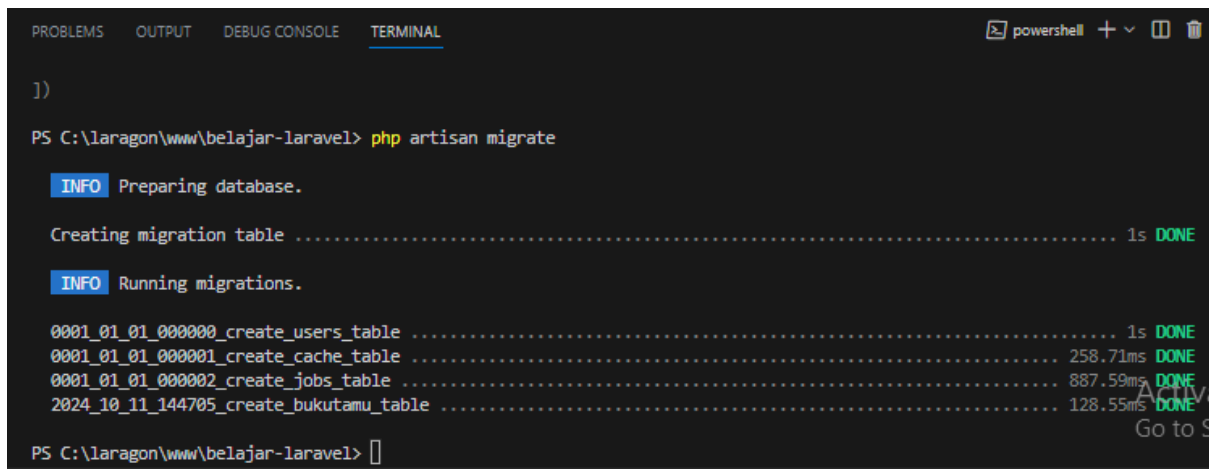
Klik menu/klik kanan di halaman putih mana saja, lalu pilih MySQL>phpMyAdmin



Klik New>SQL>ketik **CREATE DATABASE belajar_laravel CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;**

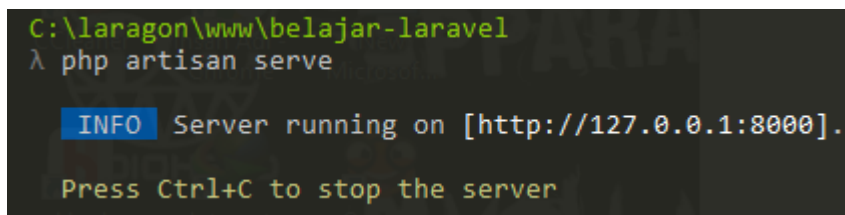


Pada terminal Laragon atau terminal Visual Studio Code, ketik: `belajar-laravel$ php artisan migrate`

A screenshot of a terminal window in Visual Studio Code. The terminal shows the command 'php artisan migrate' being executed. The output includes an 'INFO' message 'Preparing database.', followed by 'Creating migration table' which takes 1s and is marked 'DONE'. Another 'INFO' message 'Running migrations.' is shown, followed by a list of migrations: '0001_01_01_000000_create_users_table' (1s, DONE), '0001_01_01_000001_create_cache_table' (258.71ms, DONE), '0001_01_01_000002_create_jobs_table' (887.59ms, DONE), and '2024_10_11_144705_create_bukutamu_table' (128.55ms, DONE). The terminal ends with the prompt 'PS C:\laragon\www\belajar-laravel>'.

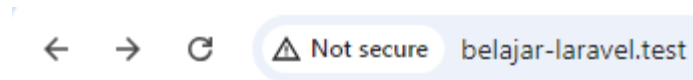
Buka Aplikasi Laravel dari Browser Langsung dari folder project dengan php web server.

Pada terminal Laragon, ketik: `belajar-laravel$ php artisan serve`

A screenshot of a terminal window showing the output of the 'php artisan serve' command. The prompt is 'C:\laragon\www\belajar-laravel' and the command entered is 'php artisan serve'. The output shows an 'INFO' message: 'Server running on [http://127.0.0.1:8000].'. Below this, it says 'Press Ctrl+C to stop the server'.

Pada Browser, ketik: <http://belajar-laravel.test>

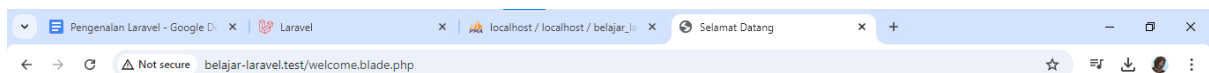
Setelah tampilan seperti ini, klik: [welcome.blade.php](#)



Index of /

- [.editorconfig](#)
- [.env](#)
- [.env.example](#)
- [.gitattributes](#)
- [.gitignore](#)
- [README.md](#)
- [app/](#)
- [artisan](#)
- [bootstrap/](#)
- [bukutamu/](#)
- [composer.json](#)
- [composer.lock](#)
- [config/](#)
- [database/](#)
- [package.json](#)
- [phpunit.xml](#)
- [public/](#)
- [resources/](#)
- [routes/](#)
- [storage/](#)
- [tests/](#)
- [vendor/](#)
- [vite.config.js](#)
- [welcome.blade.php](#)

Hasil Tampilan Akhir



Selamat Datang di Buku Tamu

[Isi Buku Tamu](#) | [Lihat Komentar](#)

Kesimpulan:

Aplikasi ini telah berhasil menampilkan cara penggunaan MVC di Laravel, pemanfaatan fitur migration untuk pembuatan tabel, validasi form untuk memastikan data yang diinputkan valid, serta penggunaan routing untuk mengarahkan ke berbagai halaman yang dibutuhkan.

Saran:

1. Validasi dan Keamanan Lebih Lanjut
2. Paginasi pada Halaman Komentar
3. Pengelolaan Data Lebih Lanjut (CRUD)
4. Desain UI/UX yang Lebih Baik
5. Autentikasi dan Admin Panel
6. Fitur Pencarian dan Filter Komentar
7. Internasionalisasi

Dengan pengembangan lebih lanjut ini, aplikasi Buku Tamu Laravel akan semakin kaya fitur, lebih aman, dan memberikan pengalaman yang lebih baik bagi pengguna maupun admin.

Link Github: <https://github.com/ihsap/TugasPBW.git>