

MANUAL BOOK
PEMBUATAN APLIKASI PENDAFTARAN SISWA BARU (PPDB)



Penulis :
Salsabila Nur Aisyah

SEKOLAH MENENGAH KEJURUAN
SMK INFORMATIKA UTAMA
REKAYASA PERANGKAT LUNAK

LANGKAH-LANGKAH PERANCANGAN

1. Pembuatan Proyek

Membuat proyek Laravel melalui command prompt dengan menggunakan versi 7.

```
C:\xampp\htdocs>composer create-project laravel/laravel Penerimaan_Siswa_Baru_PPDB "7.*"
```

Ini akan membuat proyek Laravel dengan nama "Penerimaan_Siswa_Baru_PPDB" menggunakan versi 7.

2. Konfigurasi database

Masuk ke dalam env lalu setting database.

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

3. Pembuatan Migrasi, Model, dan Controller

Setelah proyek dibuat, buka Visual Studio Code, lalu lanjutkan dengan membuat model, migrasi, dan kontroler menggunakan perintah:

```
php artisan make:model Pendaftar -m -r
```

Opsi -m digunakan untuk membuat migrasi, sedangkan -r untuk membuat kontroler.

4. Pengisian File Migrasi

Isi file migrasi dengan bidang yang telah ditentukan, serta tentukan jenis data yang sesuai. Dengan beberapa tipe data yang digunakan seperti string, integer, date dan text.

```

database > migrations > 2024_06_08_013720_create_pendaftars_table.php
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  class CreatePendaftarsTable extends Migration
8  {
9      /**
10     * Run the migrations.
11     *
12     * @return void
13     */
14     public function up()
15     {
16         Schema::create('pendaftars', function (Blueprint $table) {
17             $table->bigIncrements('Id_pendaftar');
18             $table->string('Nm_pendaftar');
19             $table->text('Alamat');
20             $table->string('Jenis_kelamin');
21             $table->integer('No_hp');
22             $table->string('Jurusan');
23             $table->date('Tgl_lahir');
24             $table->integer('NISN');
25             $table->timestamps();
26         });
27     }
28
29     /**
30     * Reverse the migrations.
31     *
32     * @return void
33     */
34     public function down()
35     {
36         Schema::dropIfExists('pendaftars');
37     }
38 }

```

5. Pengisian File Model

File model diisi dengan logika yang telah diisi di migrasi.

```

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Pendaftar extends Model
{
    protected $primaryKey = 'Id_pendaftar';

    protected $fillable = [
        'Nm_pendaftar',
        'Alamat',
        'Jenis_kelamin',
        'No_hp',
        'Jurusan',
        'Tgl_lahir',
        'NISN',
    ];
}

```

6. Pengisian Kontroler

File kontroler juga diisi, dibagian index, create, store, edit, update, dan delete.

- Isi controller public function index, dengan variable \$pendaftars yang menyimpan seluruh data index.

```

/**
 * @return \Illuminate\Http\Response
 */
public function index()
{
    $pendaftars = Pendaftar::all();
    return view('pendaftars.index', compact('pendaftars'));
}

```

- Isi controller public function create, yang akan menampilkan form create

```

/**
 * @return \Illuminate\Http\Response
 */
public function create()
{
    return view('pendaftars.create');
}

```

- Isi controller public function store, yang akan menambah data ketika di create.

```

*/
public function store(Request $request)
{
    // Validation rules
    $rules = [
        'Nm_pendaftar' => 'required|string|max:255',
        'Alamat' => 'required|string',
        'Jenis_kelamin' => 'required|string|max:10',
        'No_hp' => 'required|numeric',
        'Jurusan' => 'required|string|max:3',
        'Tgl_lahir' => 'required|date',
        'NISN' => 'required|string',
    ];

    $this->validate($request, $rules);

    Pendaftar::create($request->all());

    return redirect()->route('pendaftars.index')->with('success', 'Data created successfully');
}

```

- Isi controller public function edit, yang akan menampilkan tampilan edit.

```

*/
public function edit(Pendaftar $pendaftar)
{
    return view('pendaftars.edit', compact('pendaftar'));
}
/**

```

- Isi controller public function update, yang akan mengubah tampilan sebelumnya di index.

```

public function update(Request $request, Pendaftar $pendaftar)
{
    $rules = [
        'Nm_pendaftar' => 'required|string|max:255',
        'Alamat' => 'required|string',
        'Jenis_kelamin' => 'required|string|max:10',
        'No_hp' => 'required|numeric',
        'Jurusan' => 'required|string|max:3',
        'Tgl_lahir' => 'required|date',
        'NISN' => 'required|string',
    ];

    $this->validate($request, $rules);

    $pendaftar->update($request->all());

    return redirect()->route('pendaftars.index')->with('success', 'Data created successfully');
}

```

- Isi controller public function delete, yang akan menghapus data-data.

```

* @return \Illuminate\Http\Response
*/
public function destroy(Pendaftar $pendaftar)
{
    $pendaftar->delete();
    return redirect()->route('pendaftars.index')->with('success', 'Data deleted successfully');
}

```

7. Mengisi Route

Route didefinisikan di file route/web.php untuk mengarahkan permintaan HTTP ke fungsi yang sesuai dalam kontroler.

```
6 Route::get('/', function () {
7     return view('welcome');
8 });
9
10 Route::resource('pendaftars', 'PendaftarController');
```

8. Membuat View

Membuat folder dengan nama "pendaftars", lalu membuat 3 file yaitu, Index.blade.php, Create.blade.php, dan Edit.blade.php.

9. Mengisi Index

Setelah index terisi dengan sintaks HTML, tampilkan tampilan yang diinginkan. Tampilannya :



Index ini dibuat untuk melihat antarmuka pengguna/tampilan.

10. Mengisi Create

Setelah create terisi dengan sintaks HTML, tampilkan tampilan yang diinginkan. Tampilannya :

The screenshot shows a web browser window with the URL `127.0.0.1:8000/pendaftaran/create`. The page title is "Registrasi". The form contains the following fields:

- Nama pendaftar: [Empty text input]
- Alamat: [Empty text input]
- Jenis kelamin: Radio buttons for "Perempuan" (selected) and "Laki-laki".
- No_hp: [Empty text input]
- Institusi: Dropdown menu with "RPL" selected.
- Tgl_lahir: Date picker showing "11/06/2024".
- NISN: [Empty text input]
- Submit: Blue button.

Merupakan sebuah formulir, yang diisi oleh pengguna.

11. Mengisi Edit

Setelah edit terisi dengan sintaks HTML, tampilkan tampilan yang diinginkan.

The screenshot shows a web browser window with the URL `127.0.0.1:8000/pendaftaran/4/edit`. The page title is "EDIT". The form contains the following fields:

- Nama pendaftar: "salsabila"
- Alamat: "pordek"
- Jenis kelamin: Radio buttons for "Perempuan" (selected) and "Laki-laki".
- No_hp: "8829822"
- Institusi: Dropdown menu with "RPL" selected.
- Tgl_lahir: Date picker showing "11/06/2024".
- NISN: "888888"
- Submit: Blue button.

Edit ini berfungsi jika pengguna mengalami kesalahan dalam penulisan.

12. Penghapusan Data

Delete diperlukan untuk menghapus data yang tidak dibutuhkan. Setelah dihapus, data akan hilang atau terhapus dari database.

Dan penghapusan ini jika terjadi double data atau data yang sudah tidak terpakai.

Ini adalah langkah-langkah dalam pembuatan aplikasi pendaftaran siswa baru menggunakan Laravel. Dengan mengikuti langkah-langkah ini, aplikasi akan dibangun dengan rapih dan terstruktur.