Software Requirements Specification (SRS)

Sistem Informasi Posyandu Digital

1. Pendahuluan

1.1 Tujuan

Dokumen ini menjabarkan kebutuhan sistem digital untuk **mencatat, mengelola, dan memantau data pasien anak (balita)** dan **riwayat imunisasi** di Posyandu, dengan pembagian peran super_admin dan petugas. Sistem ini bertujuan menggantikan pencatatan manual yang rawan kesalahan dan duplikasi.

1.2 Lingkup Sistem

Sistem ini akan:

- Menggantikan pencatatan fisik dengan sistem berbasis database terpusat.
- Mendukung dua modul utama:
  a. **Manajemen Data Pasien Anak**
  b. **Manajemen Imunisasi**
- Menyediakan 2 peran pengguna:
  a. super_admin
  b. petugas

1.3 Definisi Singkatan

- CRUD: Create, Read, Update, Delete
- NIK: Nomor Induk Kependudukan
- TTL: Tempat, Tanggal Lahir

2. Deskripsi Umum

2.1 Pengguna Sistem

Role	Tugas	Batasan Akses
super_admin	Mengelola akun petugas, data master	Full CRUD
Petugas	Mencatat dan mengupdate data pasien dan imunisasi	Hanya edit data yang dimasukkan sendiri
Orang Tua	Memberikan data ke petugas	Tidak ada akses sistem

2.2 Lingkungan Operasional

- Framework: Laravel 12
- Database: MySQL/MariaDB dengan enkripsi data sensitif
- Filament v3 (menggunakan fila-starter kit)
- Docker Dekstop

3. Role dan Hak Akses

3.1 Role: super_admin

No	Fitur	Akses	Batasan
1	Manajemen User	- CRUD akun petugas - Reset password petugas	Tidak bisa menghapus diri sendiri
2	Manajemen Pasien	- Lihat semua data pasien - CRUD data pasien	-
3	Manajemen Imunisasi	- Lihat semua data imunisasi - CRUD data imunisasi	-

3.2 Role: petugas

No	Fitur	Akses	Batasan
1	Manajemen Pasien	- Input pasien baru - Edit data pasien yang mereka input	Hanya bisa lihat/edit data sendiri

2	Manajemen Imunisasi	- Catat pemberian vaksin - Update status imunisasi (tertunda/selesai)	Tidak bisa hapus riwayat

4. Struktur Database dan Migrasi
Tabel Pasien

Field	Tipe Data	Deskripsi
id	INT (Primary Key)	ID pasien
nik	INT (Unique)	NIK pasien
tanggal_lahir	date	tanggal lahir pasien
jenis_kelamin	enum [‘L’, ‘P’]	Jenis kelamin untuk pasien
alamat	text	alamat pasien
nama_ortu	string	nama ortu pasien
foto	string (path direktori fotonya)	mengupload foto pasien secara opsional
petugas_id	int (Foreign Key)	petugas yang mencatat data pasien
created_at	timestamps	dibikin pada
updated_at	timestamps	diupdate pada

Tabel Imunisasi

Field	Tipe Data	Deskripsi
ID	INT (Primary Key)	ID Imunisasi
pasien_id	INT (Foreign Key)	Pasien yang mendapat imunisasi

jenis_vaksin	enum [‘BCG’, ‘Polio’, ‘DPT’, ‘Hepatitis B’, ‘Campak’]	Jenis vaksin
tanggal	date	tanggal imunisasi
status	enum [‘Tertunda’, ’Selesai’]	status imunisasinya
created_at	timestamps	dibikin pada
updated_at	timestamps	diupdate pada

5. Model & Relasi
Relasi User (pada App/Models/User.php)

```
public function pasien()
   {
      return $this->hasMany(Pasien::class, 'petugas_id');
   }
```

- **Relasi**: One-to-Many (1 petugas → banyak pasien)
- **Penjelasan**:
  - Satu user (dengan role `petugas`) dapat menangani **banyak data pasien**.
  - Kolom `petugas_id` di tabel `pasien` sebagai foreign key.
- **Contoh Use Case**:
  - Petugas A mendaftarkan 10 pasien → `petugas_id` di semua pasien tersebut bernilai ID Petugas A.

Model & Relasi Pasien (pada App/Models/Pasien.php)

```
protected $table = 'pasien'; // karena Laravel default-nya 'pasiens'

   protected $fillable = [
      'nik', 'nama', 'tanggal_lahir', 'jenis_kelamin',
      'alamat', 'nama_ortu', 'foto', 'petugas_id'
   ];

   protected $casts = [
      'tanggal_lahir' => 'date',
   ];
```

```
public function petugas()
{
    return $this->belongsTo(User::class, 'petugas_id');
}

public function imunisasi()
{
    return $this->hasMany(Imunisasi::class);
}
```

- **Properti**:
  1. `$table`: Menyesuaikan nama tabel ke `pasien` (Laravel default plural: `pasiens`).
  2. `$fillable`: Field yang boleh diisi via mass assignment (misal: `Pasien::create($request->all())`).
  3. `$casts`: Konversi `tanggal_lahir` ke objek Carbon (memudahkan manipulasi tanggal).
- **Relasi**:
  1. **Many-to-One ke User**:
     - Banyak pasien bisa dimiliki oleh **1 petugas**.
     - Contoh: 50 pasien di Posyandu X ditangani oleh Petugas B.
  2. **One-to-Many ke Imunisasi**:
     - Satu pasien bisa memiliki **banyak riwayat imunisasi**.
     - Contoh: Pasien C punya 5 catatan imunisasi (BCG, Polio, dll).

Model dan Relasi Imunisasi (pada App/Models/Imunisasi.php)

```
protected $fillable = [
    'pasien_id', 'jenis_vaksin', 'tanggal', 'status'
];

protected $casts = [
    'tanggal' => 'date',
];

public function pasien()
{
    return $this->belongsTo(Pasien::class);
}
```

- **Properti**:
  - `$fillable`: Field yang bisa diisi massal.
  - `$casts`: Konversi `tanggal` ke Carbon (misal: `$imunisasi->tanggal->format('d F Y')`).
- **Relasi**:

- **Many-to-One ke Pasien**:
  - Setiap imunisasi **hanya dimiliki oleh 1 pasien**..
  - Contoh: Riwayat imunisasi Hepatitis B milik Pasien D.

6. Seeder & Data Dummy

6.1 Apakah Dibutuhkan Seeder?

Ya, sistem memerlukan data dummy (seeder) karena:

- **Untuk Testing**
  Memastikan semua fitur CRUD berjalan sesuai ekspektasi, termasuk validasi, filtering, dan role-based access.

6.2 Isi Data Dummy

No	Table/Model	Jumlah Data	Catatan
1	users	1 super_admin, 2 petugas	super_admin memiliki hak akses penuh
2	pasien	10 pasien anak	dibagi merata ke 2 petugas
3	imunisasi	5 pasien dari 10 memiliki 3–5 imunisasi	Jenis vaksin acak

6.3 Seeder Terkait

DatabaseSeeder.php (pada database/seeder/)

```
public function run()
{
    $this->call([
        RoleSeeder::class,
        UserSeeder::class,
        PasienSeeder::class,
        ImunisasiSeeder::class,
    ]);
```

}

UserSeeder.php (pada database/seeder/)

```
public function run(): void
    {
        $user = User::firstOrCreate(
            ['email' => 'admin@demo.com'],
            [
                'name' => 'Super Admin',
                'password' => bcrypt('password'),
                'role' => 'super_admin',
            ]
        );
        $user->assignRole('super_admin');

        $user = User::firstOrCreate(
            ['email' => 'petugas1@demo.com'],
            [
                'name' => 'Petugas 1',
                'password' => bcrypt('password'),
                'role' => 'petugas',
            ]
        );
        $user->assignRole('petugas');
        $user = User::firstOrCreate(
            ['email' => 'petugas2@demo.com'],
```

```
            [
                'name' => 'Petugas 2',
                'password' => bcrypt('password'),
                'role' => 'petugas',
            ]
        );
        $user->assignRole('petugas');
    }
```

PasienSeeder.php (pada database/seeder/)

```
public function run(): void
    {
        $faker = Faker::create();

        $petugasIds = User::where('role', 'petugas')->pluck('id')->toArray();

        if (count($petugasIds) < 2) {
            echo "Minimal harus ada 2 petugas";
            return;
        }

        for ($i = 0; $i < 10; $i++) {
            // Pasien 0-4 ke petugasIds[0], pasien 5-9 ke petugasIds[1]
            $petugasId = $i < 5 ? $petugasIds[0] : $petugasIds[1];
```

```
        Pasien::create([
            'nik' => $faker->unique()->numerify('##########'),
            'nama' => $faker->name(),
            'tanggal_lahir' => $faker->date('Y-m-d', '2005-01-01'),
            'jenis_kelamin' => $faker->randomElement(['L', 'P']),
            'alamat' => $faker->address(),
            'nama_ortu' => $faker->name('male'),
            'foto' => null,
            'petugas_id' => $petugasId,
        ]);
    }
}
```

ImunisasiSeeder.php (pada database/seeder/)

```
public function run(): void
{
    $jenisVaksin = ['BCG', 'Polio', 'DPT', 'Hepatitis B', 'Campak'];

    Pasien::take(5)->get()->each(function ($pasien) use ($jenisVaksin) {
        foreach (array_slice($jenisVaksin, 0, rand(3, 5)) as $vaksin) {
            Imunisasi::create([
                'pasien_id' => $pasien->id,
                'jenis_vaksin' => $vaksin,
                'tanggal' => $pasien->tanggal_lahir->copy()->addMonths(rand(1, 12)),
                'status' => 'selesai',
```

```
            ]);
        }
    });
}
```

RoleSeeder.php

```
Role::firstOrCreate(['name' => 'super_admin', 'guard_name' => 'web']);
Role::firstOrCreate(['name' => 'petugas', 'guard_name' => 'web']);
```

6.4 Apakah Data Dummy Bisa diCRUD?

Ya. Semua data dummy yang dibuat menggunakan seeder:

- Bisa ditampilkan di panel Filament
- Bisa diubah/dihapus oleh pengguna yang sesuai role-nya
- Berlaku seperti data asli

7. Multi Panel

7.1 Apakah Perlu Menggunakan Multi Panel?

Penggunaan **multi panel** sangat disarankan dalam sistem ini karena terdapat perbedaan peran dan fitur antara super_admin dan petugas. Dengan pendekatan ini, sistem menjadi **lebih aman**, **modular**, dan **mudah digunakan** oleh masing-masing role.

7.2 Struktur Panel pada fila-starter

fila-starter sudah menyediakan dukungan multi-panel secara default. Berikut ini adalah dua panel yang digunakan:

Panel ID	Role Pengguna	Path URL	Guard	Resource
admin	super_admin	/admin	auth:web	PasienResource, ImunisasiResource
petugas	petugas	/petugas	auth:web	PasienResource, ImunisasiResource

7.3 Contoh Filament Resource per Panel
panel admin
php artisan make:filament-resource Pasien –generate --panel=admin
php artisan make:filament-resource Imunisasi –generate --panel=admin

Fitur:
- Admin bisa mengelola seluruh data.
- Lihat semua pasien dan semua imunisasi.
- CRUD akun petugas.

panel petugas
php artisan make:filament-resource Pasien --panel=petugas
php artisan make:filament-resource Imunisasi --panel=petugas

Fitur:

- Petugas hanya bisa melihat dan mengelola data yang mereka input sendiri.

7.4 Filter Data Berdasarkan petugas_id
Di PasienResource pada panel petugas
```
public static function getEloquentQuery(): Builder
{
   return parent::getEloquentQuery()
       ->where('petugas_id', auth()->id());
}
```

Di ImunisasiResource pada panel petugas
```
public static function getEloquentQuery(): Builder
{
   return parent::getEloquentQuery()
       ->whereHas('pasien', function ($query) {
          $query->where('petugas_id', auth()->id());
       });
```

}

8. Out Of Scope

- Integrasi dengan sistem eksternal (seperti BPJS, rumah sakit, EMR)
- Fitur konsultasi dokter atau telemedicine
- Akses sistem langsung untuk orang tua