

Software Requirements Specification (SRS)

1. Pendahuluan

1.1 Tujuan Dokumen

Dokumen ini menjelaskan secara rinci kebutuhan perangkat lunak untuk sistem manajemen *scholarship* pada lingkungan akademik. Fokus utama sistem ini adalah pengelolaan data *student*, data *scholarship*, dan pencatatan relasi *student_scholarship* sebagai dokumentasi pemberian *scholarship* kepada *student*.

1.2 Lingkup Sistem

Sistem ini dirancang untuk memudahkan institusi akademik dalam:

- Mengelola data *student* (nama, email, nim)
- Mengelola daftar *scholarship* (name, description, provider, start_date, end_date, is_active)
- Mencatat *student_scholarship* beserta tanggal pemberian (*awarded_at*)
- Menampilkan histori pemberian *scholarship* kepada *student*
- Menyediakan pencarian dan pelacakan data yang efisien

2. Deskripsi Umum

2.1 Perspektif Sistem

Aplikasi ini berbasis web dan digunakan secara internal oleh pihak akademik (admin/operator akademik). Sistem dibangun menggunakan framework Laravel dengan arsitektur berbasis model relasional.

2.2 Fungsi Utama Sistem

- CRUD data *student*
- CRUD data *scholarship*
- Pencatatan relasi *student_scholarship*
- Menampilkan histori pemberian *scholarship* berdasarkan *student* maupun *scholarship*
- Validasi dan integritas data relasional

2.3 Pengguna Sistem

- Admin Akademik: Mengelola seluruh data dan hubungan antar entitas

3. Kebutuhan Fungsional

3.1 Modul Student

1. Admin dapat menambahkan, mengedit, dan menghapus data *student*.

2. Data yang dikelola:
 - name (string)
 - email (string, unik)
 - nim (string, unik)

3.2 Modul Scholarship

1. Admin dapat menambahkan, mengedit, dan menghapus data *scholarship*.
2. Data yang dikelola:
 - name (string)
 - description (text, nullable)
 - provider (string, nullable)
 - start_date (date, nullable)
 - end_date (date, nullable)
 - is_active (boolean, default: true)

3.3 Modul Student_Scholarship

1. Admin dapat mencatat hubungan antara *student* dan *scholarship*.
2. Data yang dikelola:
 - student_id (foreignId ke tabel students, cascade on delete)
 - scholarship_id (foreignId ke tabel scholarships, cascade on delete)
 - awarded_at (date, default: now())
3. Sistem menampilkan daftar *student* penerima suatu *scholarship* dan sebaliknya

4. Kebutuhan Non-Fungsional

4.1 Keamanan

- Hanya admin yang dapat mengakses sistem
- Validasi data dilakukan pada setiap input untuk mencegah duplikasi atau kesalahan data

4.2 Kinerja

- Sistem merespon permintaan dalam waktu maksimal 2 detik
- Sistem mampu mengelola data skala menengah hingga ribuan entri

4.3 Reliabilitas

- Data disimpan dalam sistem database relasional (MySQL/MariaDB)
- Sistem mendukung backup berkala

4.4 Ketersediaan

- Sistem dapat diakses setiap waktu oleh admin melalui jaringan local(https)/internal

5. Batasan Sistem

1. Tidak tersedia akses langsung oleh *student*
2. Tidak mendukung proses seleksi/pengajuan beasiswa otomatis
3. Tidak mendukung pengunggahan dokumen atau bukti fisik (*optional feature* di masa depan)

6. Antarmuka Pengguna (UI)

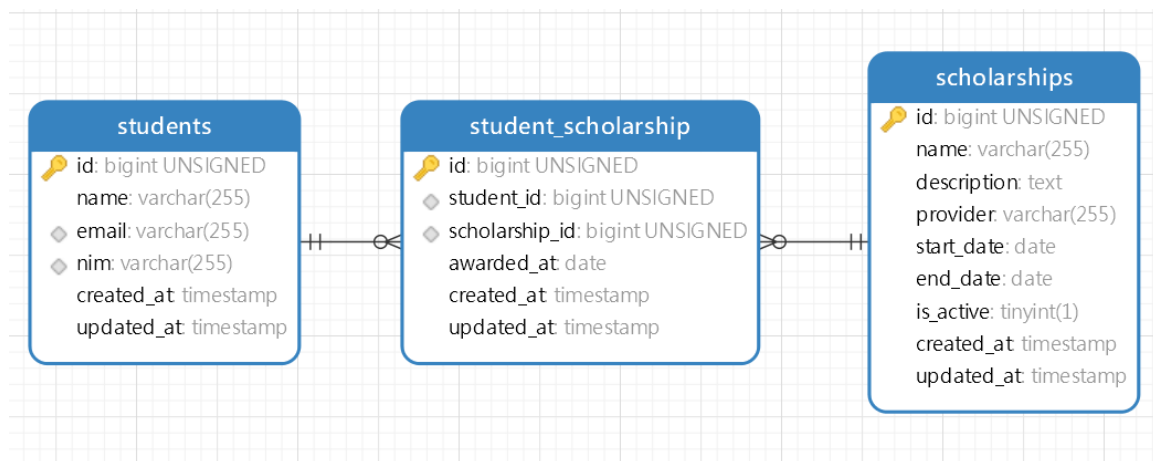
- Berbasis admin panel (menggunakan Filamen)
- Navigasi:
 - Students
 - Scholarships
 - Student_Scholarship
- Input form mengikuti skema kolom tabel dari masing-masing entitas

7. Diagram Kelas & Relasi

Sistem terdiri dari 3 entitas utama:

- students
- scholarships
- student_scholarship

Relasi:

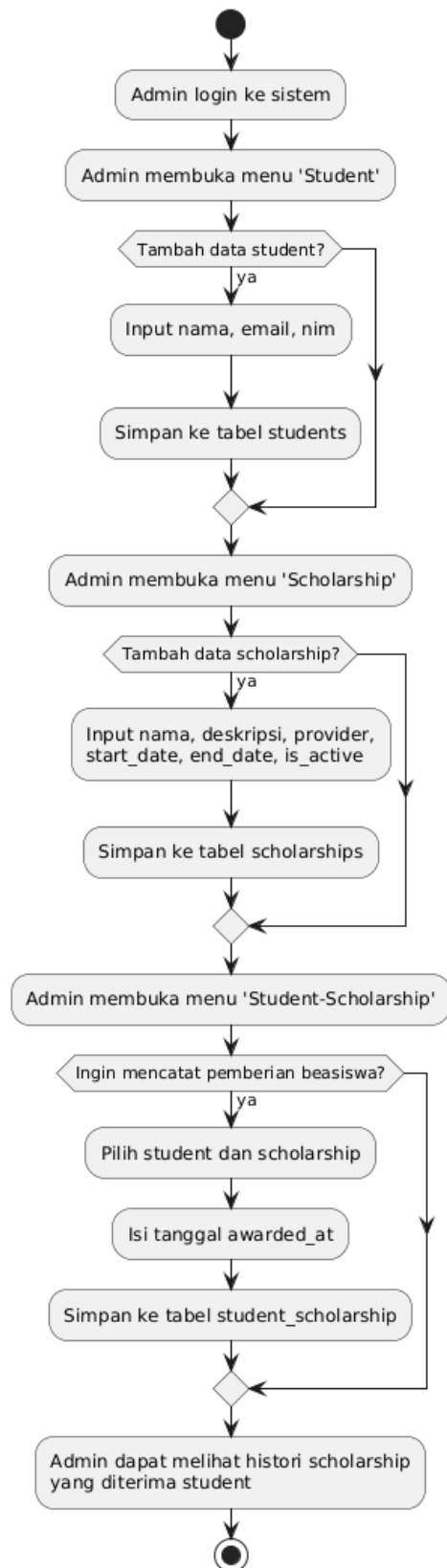


8. Catatan Tambahan

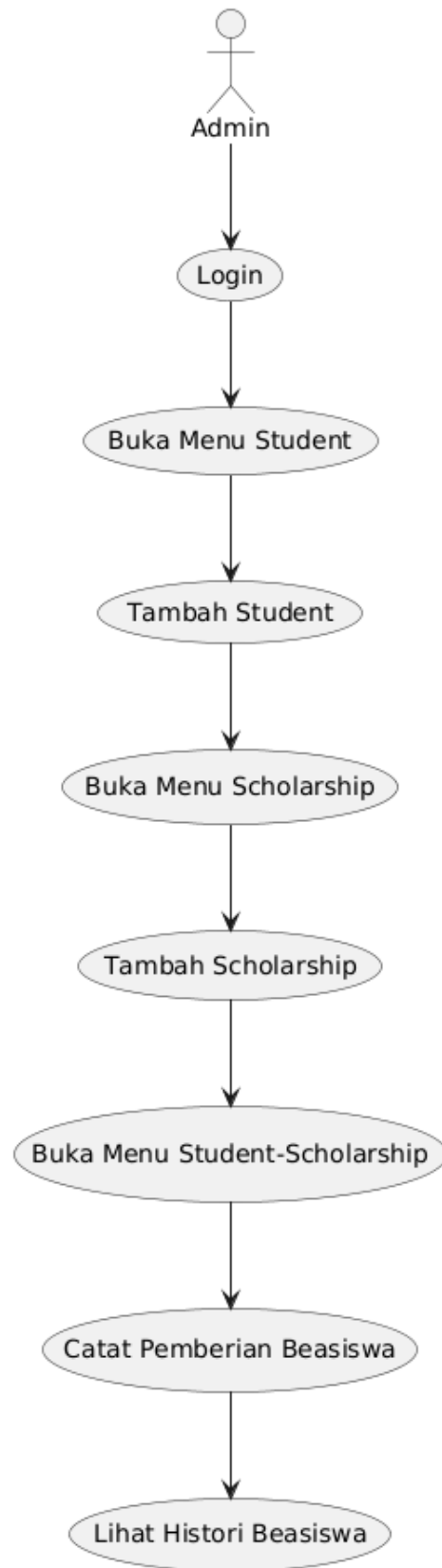
Sistem dapat dikembangkan lebih lanjut dengan fitur-fitur tambahan seperti:

- Akses portal mandiri oleh student
- Pengajuan scholarship secara daring
- Integrasi notifikasi untuk deadline atau status beasiswa

9. Flowchart



10. UseCase



10. API

```
Route::prefix('scholarships')->middleware('apikey')->group(function () {
    Route::get('/', [ScholarshipController::class, 'index']);
    Route::post('/', [ScholarshipController::class, 'store']);
    Route::get('{id}', [ScholarshipController::class, 'show']);
    Route::put('{id}', [ScholarshipController::class, 'update']);
    Route::delete('{id}', [ScholarshipController::class, 'destroy']);
});

Route::prefix('students')->middleware('apikey')->group(function () {
    Route::get('/', [StudentController::class, 'index']);
    Route::post('/', [StudentController::class, 'store']);
    Route::get('{id}', [StudentController::class, 'show']);
    Route::put('{id}', [StudentController::class, 'update']);
    Route::delete('{id}', [StudentController::class, 'destroy']);
});

Route::prefix('student-scholarships')->middleware('apikey')->group(function () {
    Route::get('/', [StudentScholarshipController::class, 'index']);
    Route::post('/', [StudentScholarshipController::class, 'store']);
    Route::delete('{id}', [StudentScholarshipController::class, 'destroy']);
});
```

Gambar yang ditampilkan menunjukkan konfigurasi routing API pada framework Laravel, di mana terdapat tiga grup utama route yang masing-masing bertanggung jawab mengelola data beasiswa (scholarships), data mahasiswa (students), dan relasi antara mahasiswa dan beasiswa (student-scholarships). Ketiga grup route ini menggunakan prefix URL sesuai entitasnya dan dilindungi oleh middleware bernama `apikey`, yang berfungsi sebagai lapisan keamanan agar hanya request yang memiliki API Key yang sah yang dapat mengakses API tersebut.

Pada grup route `scholarships`, sistem menyediakan endpoint RESTful standar: GET / untuk mengambil seluruh data beasiswa, POST / untuk menambahkan data baru, GET /{id} untuk melihat detail beasiswa tertentu, PUT /{id} untuk memperbarui data, dan DELETE /{id} untuk menghapus data. Hal yang sama juga diterapkan pada grup route `students`, dengan endpoint yang menangani operasi serupa namun untuk data mahasiswa.

Sementara itu, grup route `student-scholarships` berfokus pada pencatatan relasi antara mahasiswa dan beasiswa. Endpoint yang tersedia adalah GET / untuk melihat semua relasi yang ada, POST / untuk mencatat relasi baru (misalnya saat seorang mahasiswa mendapatkan beasiswa tertentu), dan DELETE /{id} untuk menghapus relasi tersebut.

Secara keseluruhan, konfigurasi routing ini mencerminkan struktur API yang rapi, RESTful, dan aman, serta mencerminkan fungsionalitas sistem manajemen scholarship berbasis Laravel.