

SOFTWARE REQUIREMENT SYSTEM (SRS)
SISTEM BEASISWA



Disusun Oleh

Nama : Ilham Sheva Renggafiarto
NIM : 20230801219
Prodi : Teknik Informatika

UNIVERSITAS ESA UNGGUL
2025

1. Pendahuluan

1.1 Tujuan

Dokumen Sistem Persyaratan Perangkat Lunak (SRS) ini merinci persyaratan fungsional dan non-fungsional untuk Sistem Beasiswa Universitas Esa Unggul. Tujuan utamanya adalah untuk menyediakan panduan yang jelas bagi tim pengembangan yang akan membangun sistem ini menggunakan Laravel dan Filament, memastikan bahwa produk akhir memenuhi kebutuhan pengguna dan pemangku kepentingan seperti yang digariskan dalam Dokumen Kebutuhan Bisnis (BRD).

1.2 Lingkup Produk

Sistem Beasiswa adalah platform digital yang akan mengelola seluruh proses pendaftaran beasiswa secara

online. Sistem ini akan memungkinkan mahasiswa untuk mendaftar dan mengunggah dokumen persyaratan, sementara staf akademik dapat memverifikasi dan menyeleksi pendaftar melalui panel administrasi. Sistem ini akan menyediakan alur kerja yang terstruktur, transparan, dan efisien, serta mengurangi proses manual yang rentan terhadap kesalahan.

1.3 Target Pengguna

Sistem ini memiliki tiga jenis peran utama dengan hak akses yang berbeda:

- **Peserta (Mahasiswa):** Pengguna akhir yang dapat mendaftar beasiswa dan mengunggah dokumen pendukung seperti KTM, KRS, dan KHS.
- **Staf Akademik:** Petugas yang memiliki wewenang untuk meninjau dan memverifikasi pendaftaran, lalu memberikan status persetujuan atau penolakan.
- **Super Admin:** Pengelola sistem dengan akses penuh untuk melihat seluruh data, mengelola akun pengguna, dan mengatur program beasiswa.

1.4 Definisi, Akronim, dan Singkatan

- **BRD:** Business Requirement Document (Dokumen Kebutuhan Bisnis)
- **SRS:** Software Requirement Specification (Spesifikasi Kebutuhan Perangkat Lunak)
- **KTM:** Kartu Tanda Mahasiswa
- **KRS:** Kartu Rencana Studi
- **KHS:** Kartu Hasil Studi
- **Laravel:** Kerangka kerja aplikasi web PHP
- **Filament:** Kerangka kerja untuk membangun panel admin Laravel

2. Deskripsi Umum

2.1 Perspektif Produk

Sistem Beasiswa ini akan menjadi platform

mandiri yang terpusat untuk semua aktivitas pendaftaran beasiswa. Ini akan diakses melalui

web browser dan tidak memiliki ketergantungan yang disebutkan secara eksternal dengan sistem lain, selain mungkin integrasi standar untuk pengiriman email atau penyimpanan *file*.

2.2 Fungsi Produk

Sistem akan menyediakan fungsi-fungsi utama berikut:

- **Autentikasi Pengguna:** Registrasi dan login untuk peserta, staf akademik, dan super admin.
- **Dasbor Peserta:** Menampilkan ringkasan dan navigasi untuk peserta, termasuk daftar beasiswa dan status pendaftaran.
- **Manajemen Pendaftaran Peserta:** Memungkinkan peserta untuk melamar beasiswa dan mengunggah dokumen.

- **Pelacakan Status Pendaftaran:** Peserta dapat melihat status pendaftaran yang diajukan.
- **Panel Staf Akademik:** Memungkinkan staf akademik untuk meninjau, memverifikasi, dan mengubah status pendaftaran beasiswa.
- **Panel Super Admin:** Mengelola program beasiswa (membuat, mengubah, menghapus) dan mengelola akun pengguna.

2.3 Karakteristik Pengguna

- **Peserta:** Mahasiswa yang akrab dengan penggunaan *web browser* dan proses pengunggahan dokumen.
- **Staf Akademik:** Petugas administrasi yang perlu meninjau dan mengelola data pendaftaran secara efisien. Mungkin memerlukan sedikit pelatihan awal untuk antarmuka Filament.
- **Super Admin:** Administrator sistem dengan pemahaman teknis yang baik, mampu mengelola data dan konfigurasi sistem.

2.4 Batasan

- Sistem hanya akan mendukung proses pendaftaran beasiswa *online*.
- Verifikasi dokumen fisik atau wawancara tidak termasuk dalam lingkup sistem ini, kecuali jika data hasil verifikasi/wawancara tersebut diinput ke sistem oleh Staf Akademik.
- Sistem tidak akan menangani proses pencairan dana beasiswa.

3. Persyaratan Fungsional

Bagian ini menjelaskan secara rinci persyaratan fungsional sistem, dengan fokus pada implementasi menggunakan Laravel dan Filament.

3.1 Modul Autentikasi

- **FR.3.1.1 Registrasi Pengguna Baru:**
 - Sistem harus menyediakan formulir registrasi yang memungkinkan pengguna baru (peserta) untuk membuat akun.
 - Formulir harus meminta informasi seperti nama lengkap, alamat email, dan kata sandi.
 - Sistem harus memvalidasi alamat email unik.
 - Setelah registrasi, pengguna harus diarahkan untuk melakukan login.
 - **Implementasi Laravel/Filament:** Menggunakan fitur autentikasi bawaan Laravel (Laravel Breeze/Jetstream atau secara manual) untuk pendaftaran, dengan kemungkinan kustomisasi untuk peran Peserta.
- **FR.3.1.2 Login Pengguna:**
 - Sistem harus menyediakan formulir login untuk pengguna yang sudah terdaftar.
 - Pengguna harus dapat login menggunakan alamat email dan kata sandi yang terdaftar.
 - Sistem harus melakukan validasi kredensial login.
 - Setelah login berhasil, pengguna harus diarahkan ke dasbor yang sesuai dengan perannya.
 - **Implementasi Laravel/Filament:** Menggunakan fitur autentikasi bawaan Laravel. Panel admin Filament akan secara otomatis menangani login untuk Staf Akademik dan Super Admin ke *dashboard* mereka.
- **FR.3.1.3 Lupa Kata Sandi:**
 - Sistem harus menyediakan fitur untuk mengatur ulang kata sandi.

3.2 Modul Dasbor Peserta

- **FR.3.2.1 Tampilan Beranda Dasbor:**

- Dasbor peserta harus menampilkan ringkasan singkat atau pengumuman penting.
- **Implementasi Laravel/Filament:** Akan dibangun sebagai halaman kustom di luar panel Filament utama, atau sebagai *Livewire component* yang terpisah jika memungkinkan diintegrasikan.
- **FR.3.2.2 Daftar Beasiswa:**
 - Peserta harus dapat menelusuri dan melihat detail semua program beasiswa yang sedang dibuka.
 - Detail beasiswa harus mencakup nama beasiswa, deskripsi, persyaratan, dan periode pendaftaran.
 - Peserta harus dapat mengklik beasiswa untuk melihat detail lebih lanjut dan memulai proses pendaftaran.
 - **Implementasi Laravel/Filament:** Ini akan menjadi halaman *frontend* yang dibangun dengan Laravel, mungkin menggunakan data yang dikelola melalui Filament di sisi admin.
- **FR.3.2.3 Pendaftaran Beasiswa:**
 - Peserta harus dapat memilih beasiswa yang aktif dan mengisi formulir pendaftaran.
 - Sistem harus memungkinkan peserta untuk mengunggah dokumen persyaratan seperti KTM, KRS, dan KHS.
 - Sistem harus memvalidasi jenis dan ukuran *file* yang diunggah.
 - Setelah pengajuan, status pendaftaran harus diatur ke "Diproses".
 - **Implementasi Laravel/Filament:** Formulir pendaftaran dan pengunggahan *file* akan menjadi bagian dari *frontend* Laravel. *File* akan disimpan di *storage* yang aman.
- **FR.3.2.4 Pelacakan Status Pendaftaran:**
 - Peserta harus dapat melihat riwayat dan status terkini dari setiap pendaftaran yang pernah diajukan ("Diproses", "Disetujui", "Ditolak").
 - **Implementasi Laravel/Filament:** Halaman ini akan menjadi bagian dari *frontend* Laravel, menampilkan data pendaftaran yang terkait dengan pengguna yang *login*.

3.3 Modul Panel Akademik (Filament)

- **FR.3.3.1 Verifikasi Pendaftaran:**
 - Staf Akademik harus dapat melihat daftar semua pendaftar beasiswa.
 - Staf Akademik harus dapat meninjau detail pendaftaran dan dokumen yang diunggah oleh peserta.
 - Staf Akademik harus dapat mengubah status pendaftaran menjadi "Disetujui" atau "Ditolak".
 - Sistem harus mencatat riwayat perubahan status pendaftaran.
 - **Implementasi Laravel/Filament:** Akan diimplementasikan sebagai *Resource* Filament untuk pendaftaran beasiswa, dengan *custom actions* untuk mengubah status dan menampilkan dokumen.
- **FR.3.3.2 Filter dan Pencarian Pendaftar:**
 - Staf Akademik harus dapat memfilter dan mencari pendaftar berdasarkan berbagai kriteria (misalnya, status pendaftaran, nama beasiswa, nama peserta).
 - **Implementasi Laravel/Filament:** Fitur filter dan pencarian *built-in* Filament akan digunakan.

3.4 Modul Panel Super Admin (Filament)

- **FR.3.4.1 Manajemen Program Beasiswa:**

- Super Admin harus dapat membuat program beasiswa baru, termasuk nama, deskripsi, persyaratan, dan periode pendaftaran.
- Super Admin harus dapat mengubah detail program beasiswa yang sudah ada.
- Super Admin harus dapat menghapus program beasiswa.
- **Implementasi Laravel/Filament:** Akan diimplementasikan sebagai *Resource* Filament untuk program beasiswa, memungkinkan operasi CRUD (Create, Read, Update, Delete).
- **FR.3.4.2 Manajemen Pengguna:**
 - Super Admin harus dapat melihat daftar semua akun pengguna (peserta, staf akademik, super admin).
 - Super Admin harus dapat membuat akun baru untuk staf akademik dan super admin.
 - Super Admin harus dapat mengedit informasi akun pengguna (misalnya, nama, email, peran).
 - Super Admin harus dapat menonaktifkan atau menghapus akun pengguna.
 - **Implementasi Laravel/Filament:** Akan diimplementasikan sebagai *Resource* Filament untuk pengguna, dengan pembatasan hak akses yang sesuai.
- **FR.3.4.3 Melihat Seluruh Data:**
 - Super Admin harus memiliki akses penuh untuk melihat seluruh data dalam sistem (pendaftar, beasiswa, pengguna).
 - **Implementasi Laravel/Filament:** Filament akan secara otomatis menyediakan akses ke semua *Resource* bagi pengguna dengan peran Super Admin.

3.5 Modul Pelaporan (Implisit dari BRD)

- **FR.3.5.1 Laporan Rekapitulasi Data:**
 - Sistem harus dapat menghasilkan laporan rekapitulasi data beasiswa untuk pimpinan institusi.
 - Laporan ini dapat mencakup jumlah pendaftar per beasiswa, status pendaftaran, dll.
 - **Implementasi Laravel/Filament:** Ini dapat diimplementasikan sebagai halaman *custom* di Filament yang menghasilkan laporan atau mengekspor data dalam format yang dapat digunakan (CSV/Excel).

4. Persyaratan Non-Fungsional

Bagian ini menjelaskan persyaratan yang berkaitan dengan kualitas sistem.

4.1 Keamanan

- **NFR.4.1.1 Kontrol Akses Berbasis Peran (RBAC):**
 - Akses ke fitur dan data harus dibatasi sesuai dengan peran pengguna (Peserta, Akademik, Super Admin).
 - Pengguna hanya boleh mengakses fitur dan data yang relevan dengan peran mereka.
 - **Implementasi Laravel/Filament:** Laravel menyediakan *middleware* dan *gates/policies* untuk RBAC. Filament memiliki integrasi yang kuat dengan otorisasi Laravel untuk mengelola hak akses ke *resources* dan halaman.
- **NFR.4.1.2 Keamanan Data:**
 - Dokumen pribadi yang diunggah oleh peserta harus disimpan dengan aman.
 - Dokumen tidak boleh dapat diakses oleh pihak yang tidak berwenang.
 - Sistem harus melindungi dari serangan umum seperti injeksi SQL dan *cross-site scripting* (XSS).

- **Implementasi Laravel/Filament:** Menggunakan *storage disk* yang aman (misalnya, *private disk* di Laravel) untuk dokumen yang diunggah. Laravel secara *default* melindungi dari banyak kerentanan keamanan web.
- **NFR.4.1.3 Enkripsi Kata Sandi:**
 - Kata sandi pengguna harus disimpan dalam format terenkripsi (dihash) di *database*.
 - **Implementasi Laravel/Filament:** Laravel secara *default* menggunakan *hashing* kata sandi.

4.2 Kinerja

- **NFR.4.2.1 Waktu Muat Halaman:**
 - Sistem harus responsif dengan waktu muat halaman tidak lebih dari 3 detik pada koneksi internet yang wajar.
 - **Implementasi Laravel/Filament:** Optimalisasi *query database*, *caching*, dan kompresi aset akan diterapkan. Filament sendiri dirancang untuk kinerja yang baik.
- **NFR.4.2.2 Skalabilitas:**
 - Sistem harus dapat menangani peningkatan jumlah pengguna dan data di masa mendatang.
 - **Implementasi Laravel/Filament:** Arsitektur Laravel dan Filament mendukung skalabilitas melalui penggunaan *database* yang efisien, *queueing*, dan *load balancing*.

4.3 Usabilitas

- **NFR.4.3.1 Antarmuka Intuitif:**
 - Desain antarmuka harus bersih, modern, dan mudah digunakan oleh semua kalangan pengguna tanpa memerlukan pelatihan khusus.
 - **Implementasi Laravel/Filament:** Filament secara inheren menyediakan antarmuka admin yang bersih dan modern. Desain *frontend* (untuk peserta) akan dibuat dengan UX/UI yang berpusat pada pengguna.
- **NFR.4.3.2 Aksesibilitas:**
 - Sistem harus dapat diakses dari berbagai *browser web* modern dan perangkat (desktop, tablet, *mobile*).
 - **Implementasi Laravel/Filament:** Filament secara *responsive* dan mendukung berbagai ukuran layar.

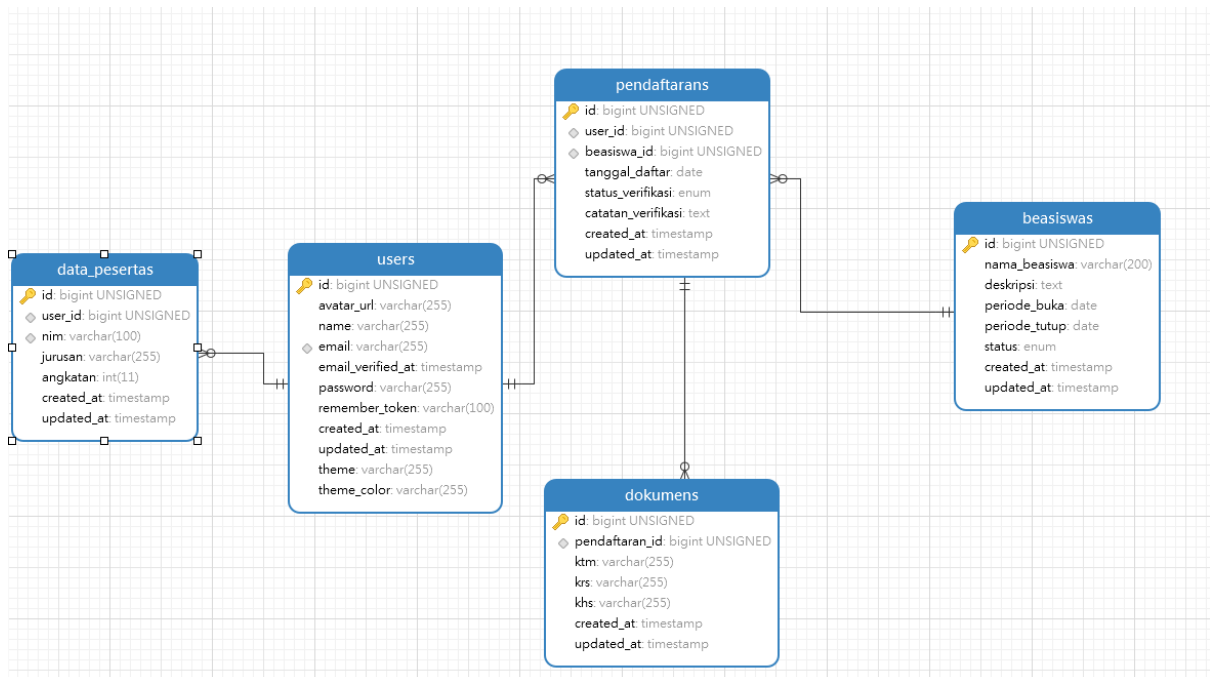
4.4 Ketersediaan

- **NFR.4.4.1 Ketersediaan Sistem:**
 - Sistem harus tersedia 24/7, kecuali untuk jendela pemeliharaan terjadwal.

4.5 Pemeliharaan

- **NFR.4.5.1 Kemudahan Pemeliharaan:**
 - Kode harus ditulis dengan standar Laravel dan praktik terbaik Filament untuk memudahkan pemeliharaan dan pengembangan di masa mendatang.
 - Dokumentasi teknis internal harus disediakan.

5. Entity Relationship Diagram (ERD)



6. Use Case Diagram

