

**MANAJEMEN PROYEK**  
**SOFTWARE REQUIREMENTS SPECIFICATION (SRS)**



Disusun Oleh:

Sifa Sahira	F55123059
Siti Hajar	F55123063
Asfita Saldarisya Nadjun	F55123072
Adelia Nur Sakinah	F55123074
Sri Aswanti	F55123075
Anggita Setiawati	F55123092
Rifaldi	F55123096

TI - A

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**  
**JURUSAN TEKNOLOGI INFORMASI**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS TADULAKO**

**2025**

# **BAB I**

## **PENDAHULUAN**

### **a. Latar Belakang**

Perkembangan teknologi informasi yang pesat telah membawa perubahan signifikan dalam berbagai bidang, termasuk dunia pendidikan tinggi. Perguruan tinggi dituntut untuk mampu beradaptasi dengan kemajuan teknologi agar kegiatan akademik dan administrasi dapat berjalan lebih efisien, transparan, serta mudah diakses oleh seluruh civitas akademika. Sistem informasi yang terintegrasi menjadi kebutuhan penting untuk menunjang proses akademik, terutama dalam pengelolaan data mahasiswa, dosen, dan mata kuliah yang jumlahnya terus bertambah setiap tahun.

Sebelum pemanfaatan sistem berbasis teknologi, berbagai aktivitas akademik di universitas sering kali dilakukan secara manual. Proses tersebut meliputi pengisian Kartu Rencana Studi (KRS), pencatatan nilai, penyusunan jadwal perkuliahan, serta rekapitulasi data akademik mahasiswa. Cara kerja manual ini menimbulkan sejumlah kendala, seperti keterlambatan dalam pengolahan data, kesalahan input, dan kesulitan dalam pencarian informasi. Selain itu, koordinasi antarbagian juga menjadi kurang efektif karena data tidak tersimpan dalam satu sistem terpusat.

Melalui penerapan sistem informasi akademik, Universitas berupaya memanfaatkan teknologi untuk mempermudah pengelolaan seluruh kegiatan akademik. Sistem ini dirancang agar dapat membantu mahasiswa, dosen, serta pihak administrasi dalam mengakses dan mengelola data akademik secara cepat dan efisien. Dengan penerapan sistem berbasis digital, pengelolaan data menjadi lebih teratur, proses administrasi lebih cepat, serta akurasi informasi dapat terjamin. Selain itu, sistem informasi akademik juga berperan penting dalam mendukung transparansi, akuntabilitas, serta peningkatan mutu layanan pendidikan di lingkungan Universitas.

Dengan adanya sistem informasi akademik, universitas dapat menciptakan tata kelola pendidikan yang lebih modern dan profesional. Sistem ini menjadi langkah strategis dalam meningkatkan kualitas pelayanan akademik dan administrasi agar sesuai dengan tuntutan zaman serta mendukung visi Universitas sebagai institusi pendidikan yang unggul dan berbasis teknologi informasi.

**b. Tujuan**

- 1) Menjelaskan kebutuhan sistem secara rinci agar pengembang memahami fungsionalitas yang harus tersedia dalam sistem.
- 2) Menyediakan acuan bagi tim pengembang dalam merancang dan membangun sistem sesuai kebutuhan pengguna di Universitas.
- 3) Menjamin agar sistem yang dikembangkan dapat berjalan dengan baik, stabil, serta memenuhi aspek fungsional dan non-fungsional.
- 4) Meningkatkan efisiensi, keakuratan, dan kecepatan dalam pengelolaan data akademik seperti KRS, KHS, jadwal perkuliahan, dan data mahasiswa.
- 5) Mendukung peningkatan kualitas pelayanan akademik berbasis teknologi informasi di Universitas.

## **BAB II**

### **LANDASAN TEORI**

Aplikasi SIAKAD dirancang untuk membantu perguruan tinggi dalam mengelola data akademik secara terpusat dan efisien. Sistem ini memungkinkan admin, dosen, dan mahasiswa untuk saling terhubung dalam satu platform digital.

Cakupan sistem mencakup fitur-fitur berikut :

- 1) Pengelolaan data pengguna (admin, dosen, mahasiswa)
- 2) Pengelolaan data akademik (mata kuliah, jadwal, nilai, absensi)
- 3) Tampilan laporan akademik dan rekap nilai per semester.
- 4) Pengumuman akademik dan informasi kegiatan kampus.

Sistem tidak mencakup :

- 1) Pengelolaan keuangan (seperti pembayaran SPP )
- 2) Fitur e-learning atau ujian online.
- 3) Sistem chat internal antar pengguna.

Istilah	Arti / Penjelasan
Siakad	Sistem Informasi Akademik
CRUD	Create, Read, Update, Delete - operasi dasar dalam pengelolaan data
Admin	Pengguna yang memiliki hak akses penuh untuk mengelola sistem
Dosen	Pengajar yang menginput nilai dan absensi mahasiswa
Mahasiswa	Pengguna yang melihat jadwal, nilai, dan data akademik pribadi
Dashboard	Halaman utama sistem yang menampilkan ringkasan informasi penting
Database	Tempat penyimpanan seluruh data yang digunakan oleh sistem
Login	Proses autentikasi untuk masuk kedalam sistem menggunakan akun terdaftar

Semester	Periode pembelajaran dalam satu tahun akademik
----------	--

## 1. Gambaran Umum Sistem

### a. Fungsi Utama Sistem

#### 1) Manajemen Data Mahasiswa

→ Sistem memungkinkan admin untuk menambah, mengubah, dan menghapus data mahasiswa (nama, NIM, program studi, status aktif, dsb.).

#### 2) Manajemen Data Dosen

→ Sistem menyediakan fitur pengelolaan data dosen, seperti nama, NIDN, jabatan akademik, mata kuliah yang diampu, dan jadwal mengajar.

#### 3) Manajemen Mata Kuliah dan KRS (Kartu Rencana Studi)

→ Mahasiswa dapat melakukan pengisian KRS secara online. Sistem menampilkan daftar mata kuliah yang tersedia setiap semester serta jadwal dan kapasitas kelas.

#### 4) Manajemen Nilai dan KHS (Kartu Hasil Studi)

→ Dosen dapat menginput nilai mahasiswa, dan mahasiswa dapat melihat hasil studi per semester dalam bentuk KHS.

#### 5) Pengelolaan Jadwal Kuliah

→ Admin dapat mengatur jadwal kuliah berdasarkan ruangan, waktu, dan dosen, sehingga tidak terjadi bentrokan jadwal.

#### 6) Cetak Dokumen Akademik

→ Sistem memungkinkan pengguna mencetak dokumen seperti KRS, KHS dan transkrip nilai.

#### 7) Autentikasi dan Manajemen Pengguna

→ Sistem memiliki fitur *login* untuk membedakan hak akses antara admin, dosen, dan mahasiswa.

### b. Karakteristik Pengguna

#### 1) Admin (Bagian Akademik / Operator Sistem)

- Peran: Mengelola seluruh data di sistem (mahasiswa, dosen, mata kuliah, jadwal, dan nilai).
- Kemampuan: Memiliki pengetahuan administrasi kampus dan dasar penggunaan komputer.

- Kebutuhan: Sistem yang mudah digunakan, cepat memproses data, dan mampu menampilkan laporan akademik dengan akurat.
- 2) Dosen
- Peran: Mengelola nilai mahasiswa, melihat daftar mahasiswa yang mengambil mata kuliah tertentu, serta mengatur jadwal dan presensi.
  - Kemampuan: Mampu mengoperasikan sistem berbasis web dan mengakses data akademik.
  - Kebutuhan: Fitur input nilai yang efisien, tampilan yang sederhana, dan laporan hasil belajar mahasiswa yang mudah diakses.
- 3) Mahasiswa
- Peran: Mengisi KRS, melihat jadwal kuliah, mengakses nilai (KHS), dan mencetak dokumen akademik.
  - Kemampuan: Mampu menggunakan aplikasi web dasar.
  - Kebutuhan: Akses cepat, tampilan yang mudah dipahami, serta keakuratan data akademik.

**c. Batasan Sistem**

Dalam perencanaan pengembangan aplikasi SIAKAD (Sistem Informasi Akademik) untuk tingkat universitas, terdapat sejumlah batasan yang perlu diperhatikan agar pengembangan sistem berjalan terarah dan realistik. Dari sisi teknis, aplikasi ini dibangun menggunakan framework Laravel sebagai platform utama pengembangan backend dan frontend dasar, dengan dukungan CSS/Tailwind untuk membangun tampilan antarmuka pengguna (user interface). Karena bersifat berbasis web, aplikasi hanya dapat dijalankan melalui browser modern seperti Google Chrome, Mozilla Firefox, atau Microsoft Edge.

Sistem ini memerlukan koneksi internet yang stabil untuk dapat diakses secara optimal, mengingat seluruh proses pengolahan data dilakukan secara daring (online) melalui server Laravel yang terhubung dengan database MySQL. Aplikasi belum mendukung mode offline maupun instalasi dalam bentuk desktop atau mobile app, sehingga akses dan fungsinya sepenuhnya bergantung pada jaringan dan server yang aktif. Selain itu, kapasitas penyimpanan data dan performa sistem juga bergantung pada spesifikasi server yang digunakan.

Dari sisi non-teknis, pengembangan aplikasi SIAKAD ini hanya difokuskan pada pengelolaan data akademik universitas, yang mencakup data mahasiswa, dosen, mata kuliah, nilai, jadwal, dan laporan akademik. Sistem

belum mencakup fitur pengelolaan keuangan seperti pembayaran SPP, penggajian dosen, maupun sistem e-learning dan ujian daring. Akses terhadap sistem juga dibatasi berdasarkan hak akses pengguna, di mana admin memiliki hak penuh untuk mengelola data, sementara dosen dan mahasiswa hanya dapat mengakses informasi sesuai perannya masing-masing. Selain itu, sistem belum terintegrasi dengan platform eksternal seperti SIAK Nasional, Learning Management System (LMS), atau API pihak ketiga.

**d. Asumsi dan Dependensi (4)**

Dalam proses perancangan dan pengembangan aplikasi SIAKAD untuk universitas, terdapat sejumlah asumsi dan dependensi yang dijadikan dasar agar sistem dapat beroperasi secara optimal dan sesuai tujuan yang diharapkan. Asumsi menggambarkan kondisi yang dianggap benar pada saat sistem dirancang, sedangkan dependensi menggambarkan komponen atau faktor eksternal yang menjadi penopang berjalannya sistem.

**1) Asumsi Sistem**

Aplikasi SIAKAD ini dibangun dengan beberapa asumsi dasar berikut:

- a) Seluruh pengguna sistem, baik admin, dosen, maupun mahasiswa, memiliki perangkat (komputer, laptop, atau smartphone) yang dapat mengakses internet.
- b) Pengguna menggunakan browser modern yang kompatibel dengan teknologi web.
- c) Setiap pengguna memiliki akun login resmi yang diberikan oleh pihak universitas untuk menjaga keamanan dan otentikasi sistem.
- d) Pihak universitas telah memiliki data akademik yang valid dan terstruktur, sehingga dapat diimpor ke dalam sistem dengan mudah.
- e) Server aplikasi Laravel beroperasi dengan ketersediaan tinggi (uptime baik) agar dapat diakses kapan saja tanpa gangguan.
- f) Pengguna memiliki pemahaman dasar dalam mengoperasikan aplikasi berbasis web, seperti login, input data, dan melihat laporan nilai.

**2) Dependensi Sistem**

Selain asumsi, sistem ini juga memiliki beberapa ketergantungan utama (dependensi) sebagai berikut:

- a) Sistem bergantung pada framework Laravel sebagai kerangka utama pengembangan aplikasi untuk mengatur logika bisnis, autentikasi pengguna, dan integrasi data.
- b) Aplikasi memerlukan database MySQL sebagai tempat penyimpanan seluruh data akademik, termasuk data mahasiswa, dosen, mata kuliah, nilai, dan jadwal.
- c) Sistem membutuhkan server web untuk menjalankan aplikasi Laravel dengan baik.
- d) Aplikasi bergantung pada koneksi internet agar pengguna dapat mengakses sistem secara real-time dan melakukan sinkronisasi data antar pengguna.
- e) Browser modern menjadi komponen penting untuk menampilkan antarmuka aplikasi berbasis HTML.
- f) Kinerja sistem juga bergantung pada konfigurasi dan pemeliharaan server yang dilakukan oleh tim teknis universitas untuk memastikan keandalan dan keamanan data.

## 2. Kebutuhan Fungsional

- a. Autentikasi Pengguna (Login & Logout).

Sistem menyediakan fitur autentikasi untuk memastikan hanya pengguna terdaftar (admin, dosen, dan mahasiswa) yang dapat mengakses halaman tertentu. Tiap pengguna akan mendapatkan hak akses yang berbeda sesuai perannya. Setelah login, sistem menampilkan dashboard sesuai role; admin mengelola data akademik, dosen mengelola nilai dan KRS, mahasiswa mengakses data studi pribadi.

- 1) Aktor:

Admin, Dosen dan Mahasiswa.

- 2) Input:

Memasukkan Username/NIM/NIP dan Password.

- 3) Proses:

Pengguna membuka halaman login dan mengisi data kredensial, sistem memverifikasi data dengan database. Jika data valid sistem membuat sesi login dan menentukan hak akses pengguna. Sistem akan menampilkan halaman utama sesuai role dan saat logout sistem menghapus sesi aktif.

4) Output:

Dashboard pengguna sesuai peran (Admin/Dosen/Mahasiswa) dan tersedia notifikasi kesalahan bila login gagal.

Dengan demikian akan menjamin keamanan dan privasi data akademik dengan kontrol akses yang terpisah antar peran pengguna.

b. Manajemen Data Mahasiswa.

Admin dapat menambahkan, memperbarui, dan menghapus data mahasiswa. Data ini menjadi pondasi utama dalam seluruh modul akademik seperti KRS, KHS, dan penjadwalan.

1) Aktor:

Admin.

2) Input:

NIM, Nama, Program studi, Angkatan, Kontak, Alamat dan Status Akademik.

3) Proses:

Admin membuka menu data mahasiswa lalu dapat menambah data baru, mengedit data lama, atau menghapus mahasiswa yang sudah tidak aktif. Sistem melakukan validasi agar NIM tidak ganda dan semua field wajib terisi, perubahan tersimpan di database dan otomatis terhubung dengan modul lain seperti KRS dan nilai.

4) Output:

Tabel daftar mahasiswa yang telah diperbarui dan pesan konfirmasi setiap kali data berhasil disimpan.

Data mahasiswa tersimpan rapi dan menjadi basis utama seluruh proses akademik lainnya.

c. Manajemen Data Dosen.

Admin dapat mengelola seluruh data dosen yang aktif di kampus, termasuk identitas dasar, program studi pengampu, dan jabatan akademik. Data ini digunakan dalam penjadwalan kuliah dan input nilai.

1) Aktor:

Admin.

2) Input:

NIP, Nama Dosen, Program Studi dan Jabatan Akademik.

3) Proses:

Admin akan melakukan CRUD data dosen dan sistem langsung menyimpan data ke database.

4) Output:

Daftar dosen diperbarui di sistem.

Dengan demikian informasi dosen dapat digunakan dalam penjadwalan dan juga penilaian.

d. Manajemen Data Mata Kuliah.

Admin dapat menambahkan, memperbarui, dan menghapus mata kuliah.

1) Aktor:

Admin.

2) Input:

Kode Mata Kuliah, Nama Mata Kuliah, SKS dan Dosen Pengampu.

3) Proses:

Admin akan mengisi atau memperbarui data mata kuliah lalu sistem menyimpan perubahan dan sistem akan menampilkan daftar mata kuliah.

4) Output:

Ditampilkan daftar mata kuliah yang aktif.

Dengan demikian data mata kuliah akan menjadi dasar untuk penjadwalan dan juga KRS.

e. Pengisian KRS (Kartu Rencana Studi).

Mahasiswa dapat memprogram mata kuliah yang akan diambil setiap semester.

1) Aktor:

Mahasiswa.

2) Input:

Daftar mata kuliah tersedia dan pilihan mata kuliah.

3) Proses:

Sistem akan menampilkan daftar mata kuliah yang tersedia sesuai dengan prodi dan semester lalu mahasiswa dapat memilih mata kuliah yang diinginkan setelah itu sistem menyimpan pilihan ke database.

4) Output:

Daftar KRS mahasiswa per semester.

Dengan demikian mahasiswa dapat mengatur rencana studi dengan mandiri dan terstruktur..

f. Persetujuan KRS oleh Dosen Wali.

Dosen wali memverifikasi dan menyetujui daftar mata kuliah yang diambil mahasiswa.

1) Aktor:

Dosen.

2) Input:

Daftar KRS mahasiswa bimbingannya.

3) Proses:

Dosen meninjau daftar KRS lalu memilih setuju atau tolak disertai catatan alasan bila perlu dan sistem akan menyimpan status persetujuan.

4) Output:

Status KRS mahasiswa (disetujui/ditolak)..

Dengan demikian KRS mahasiswa tervalidasi agar sesuai dengan aturan akademik.

g. Manajemen Jadwal Kuliah.

Admin akan membuat dan mengelola jadwal perkuliahan berdasarkan mata kuliah, dosen, ruang dan waktu.

1) Aktor:

Admin.

2) Input:

Kode mata kuliah, Dosen pengampu dan Ruang serta waktu kuliah.

3) Proses:

Admin akan memasukkan detail jadwal kuliah lalu sistem memeriksa potensi bentrok waktu/ruangan jika sesuai maka sistem akan menyimpan dan menampilkan jadwal yang valid.

4) Output:

Jadwal kuliah per program studi.

Dengan demikian jadwal perkuliahan tersusun rapi dan tidak saling tumpang tindih.

h. Pengisian Absensi dan Pengelolaan Nilai Mahasiswa.

Dosen dapat memasukkan absensi mahasiswa dan nilai mahasiswa untuk setiap mata kuliah yang diampunya. Nilai terdiri dari komponen kehadiran, tugas/quiz, UTS dan UAS yang dikalkulasi menjadi nilai akhir.

1) Aktor:

Dosen dan Mahasiswa.

2) Input:

NIM mahasiswa dan komponen nilai (Kehadiran, Tugas/Quiz, UTS & UAS).

3) Proses:

Dosen membuka daftar mahasiswa dalam mata kuliah tertentu lalu mengisi status kehadiran (Hadir, Izin, Sakit, Alfa) dengan menandai kehadiran mahasiswa dan nilai sesuai komponen yang ditentukan. Sistem menghitung nilai akhir (misal: Kehadiran 70%, Tugas 5%, UTS 10% dan UAS 15%) selanjutnya data nilai disimpan dan ditampilkan dalam KHS mahasiswa.

4) Output:

Rekap kehadiran dan nilai mahasiswa per mata kuliah bisa berupa persentase per semester.

Proses absensi dan juga penilaian lebih cepat, akurat dan transparan bagi dosen dan mahasiswa, data nilai terekam secara otomatis dan dapat diakses mahasiswa kapan saja.

i. Akses dan Cetak KHS (Kartu Hasil Studi).

Mahasiswa dapat melihat dan mencetak nilai yang diperoleh tiap semester. Sistem menampilkan IP per semester dan IP kumulatif.

1) Aktor:

Mahasiswa.

2) Input:

Pilihan semester.

3) Proses:

Mahasiswa memilih semester tertentu lalu sistem menampilkan semua mata kuliah dan nilai yang diperoleh selanjutnya mahasiswa dapat mencetak hasil dalam format PDF.

4) Output:

Tampilan KHS dan file cetak (PDF).

Dengan demikian mahasiswa dapat memantau perkembangan akademik dengan jelas.

j. Laporan Akademik dan Monitoring.

Admin dapat menghasilkan laporan akademik yang berisi rekap data mahasiswa, dosen, mata kuliah, nilai, dan status keaktifan per semester.

1) Aktor:

- Admin.
- 2) Input:
    - Parameter laporan (semester, prodi, tahun akademik).
  - 3) Proses:
    - Sistem mengambil data dari seluruh modul kemudian diolah menjadi laporan ringkas dan grafik. Admin dapat mencetak laporan dalam format PDF atau Excel.
  - 4) Output:
    - Laporan akademik visual dan file ekspor.
    - Mempermudah dalam pemantauan kegiatan akademik secara menyeluruh dan akuntabel.

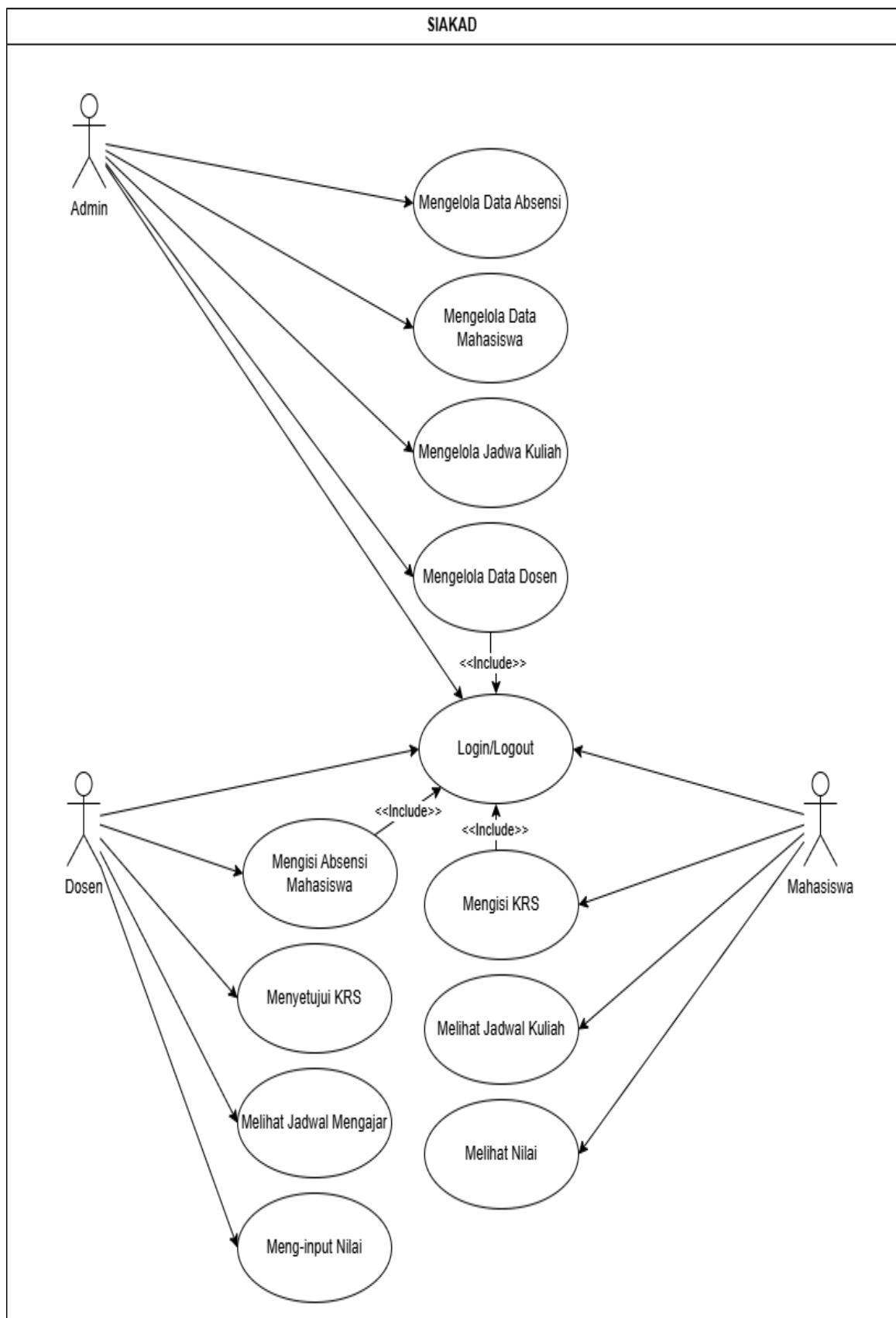
### **3. Kebutuhan Non Fungsional**

- a. Kinerja (Performance)
  - 1) Website harus dapat diakses dengan cepat saat koneksi internet berjalan normal.
  - 2) Database harus mampu menampung banyak data mahasiswa, dosen, dan nilai tanpa memperlambat proses pencarian data.
  - 3) Proses login, pengisian KRS, dan berbagai penginputan harus berlangsung dengan cepat dan efisien.
- b. Keamanan (Security)
  - 1) Setiap pengguna (admin, mahasiswa, dosen) wajib login menggunakan username, dan password yang sulit.
  - 2) Password pengguna disimpan dalam bentuk terenkripsi untuk mencegah kebocoran data.
  - 3) Sistem harus memiliki mekanisme hak akses (role-based access control) agar setiap pengguna hanya bisa mengakses sesuai perannya.
  - 4) Setiap data dilindungi dari perubahan oleh pihak yang tidak berwenang.
  - 5) setiap aktivitas penting semisal penghapusan data akan selalu dicatat dalam log aktivitas.
- c. Kemudahan Pengguna
  - 1) Antarmuka (UI) sistem harus mudah dipahami dan digunakan oleh pengguna tanpa perlu pelatihan khusus.
  - 2) Setiap halaman memiliki navigasi yang tepat, jelas dan konsisten.

- 3) Form input seperti login, pengisian krs, dan input nilai harus memiliki pesan kesalahan (error message) yang informatif jika terjadi kesalahan pengisian form.
  - 4) Warna, ikon, dan juga tulisan dirancang agar ramah pengguna dan tidak membingungkan.
- d. Profitabilitas
- 1) Sistem dapat diakses dari berbagai perangkat seperti laptop, tablet, atau hp.
  - 2) Sistem dapat dijalankan di beberapa OS seperti windows, dan IOS tanpa perlu konfigurasi tambahan.
- e. Skalabilitas
- 1) Sistem dirancang agar dapat dikembangkan lebih lanjut jika jumlah pengguna meningkat di masa depan.
  - 2) Database dapat diperluas tanpa perlu mengubah struktur sistem utama.
- f. Pemeliharaan (Maintainability)
- 1) Struktur kode program dibuat terorganisir dan mudah dipahami untuk memudahkan pengembangan di kemudian hari.

#### 4. Diagram

##### a. Use Case Diagram



## b. ERD

