

LAPORAN FINAL UML DIAGRAM

SISTEM INFORMASI AKADEMIK PERGURUAN TINGGI

Disusun Oleh:

Tim Pengembang Sistem Akademik

Tanggal:

Oktober 2025

DAFTAR ISI

1. [Pendahuluan](#)
 2. [Analisis Sistem](#)
 3. [Class Diagram](#)
 4. [Entity Relationship Diagram](#)
 5. [Penjelasan Entitas](#)
 6. [Relationship Antar Entitas](#)
 7. [Business Rules](#)
 8. [Kesimpulan](#)
-

1. PENDAHULUAN

1.1 Latar Belakang

Sistem Informasi Akademik Perguruan Tinggi adalah aplikasi web berbasis Django yang dirancang untuk mengelola aktivitas akademik di perguruan tinggi. Sistem ini mencakup manajemen data mahasiswa, dosen, mata kuliah, kelas, dan penilaian.

1.2 Tujuan Dokumen

Dokumen ini bertujuan untuk: - Menyajikan desain arsitektur database dalam bentuk UML diagram - Menjelaskan relationship antar entitas dalam sistem - Mendokumentasikan business rules yang diterapkan - Memberikan panduan untuk pengembangan dan maintenance sistem

1.3 Ruang Lingkup

Sistem mencakup: - **User Management:** Autentikasi dan otorisasi pengguna - **Academic Structure:** Fakultas, Program Studi - **People Management:** Data Mahasiswa dan Dosen - **Curriculum Management:** Mata Kuliah dan Kelas - **Assessment System:** Komponen Nilai, Penilaian, dan Nilai Akhir

2. ANALISIS SISTEM

2.1 Stakeholder

- **Administrator:** Mengelola seluruh data sistem
- **Staff Admin:** Mengelola data akademik terbatas
- **Dosen:** Mengelola kelas dan penilaian
- **Mahasiswa:** Mengakses informasi akademik pribadi

2.2 Functional Requirements

1. User Authentication & Authorization

2. Login/logout untuk berbagai role

3. Role-based access control

4. Data Management

5. CRUD operations untuk semua entitas

6. Data validation dan integrity

7. Academic Process

8. Enrollment mahasiswa ke kelas
9. Input dan kalkulasi nilai
10. Laporan akademik

2.3 Non-Functional Requirements

- **Security:** Data protection dan access control
 - **Performance:** Response time < 2 detik
 - **Scalability:** Support hingga 10,000 mahasiswa
 - **Usability:** User-friendly interface
-

3. CLASS DIAGRAM

3.1 Mermaid Code - Class Diagram

```
classDiagram
    User --> Mahasiswa
    User --> Dosen
    Fakultas --> ProgramStudi
    Fakultas --> Dosen
    ProgramStudi --> Mahasiswa
    ProgramStudi --> MataKuliah
    MataKuliah --> Kelas
    Dosen --> Kelas
    Mahasiswa --> EnrollmentKelas
    Kelas --> EnrollmentKelas
    Kelas --> KomponenNilai
    EnrollmentKelas --> Penilaian
    KomponenNilai --> Penilaian
    EnrollmentKelas --> NilaiAkhir

class User{
    id
    username
    email
    first_name
    last_name
    is_staff
    is_superuser
}

class Fakultas{
    id
    nama
    kode
    dekan
}

class ProgramStudi{
    id
    nama
```

```
    kode
    jenjang
    fakultas_id
}

class Mahasiswa{
    id
    user_id
    nim
    nama
    email
    program_studi_id
    tahun_masuk
    semester_aktif
    status
    ipk
    total_sks
}

class Dosen{
    id
    user_id
    nip
    nidn
    nama
    email
    fakultas_id
    program_studi_id
    pendidikan_terakhir
    jabatan_akademik
    status
}

class MataKuliah{
    id
    kode
    nama
    sks
    semester
    program_studi_id
    jenis
    deskripsi
```

```
}

class Kelas{
    id
    mata_kuliah_id
    dosen_id
    nama_kelas
    tahun_akademik
    semester
    kapasitas
    ruangan
    hari
    jam_mulai
    jam_selesai
    is_active
}

class EnrollmentKelas{
    id
    mahasiswa_id
    kelas_id
    tanggal_daftar
    is_active
}

class KomponenNilai{
    id
    nama
    persentase
    kelas_id
}

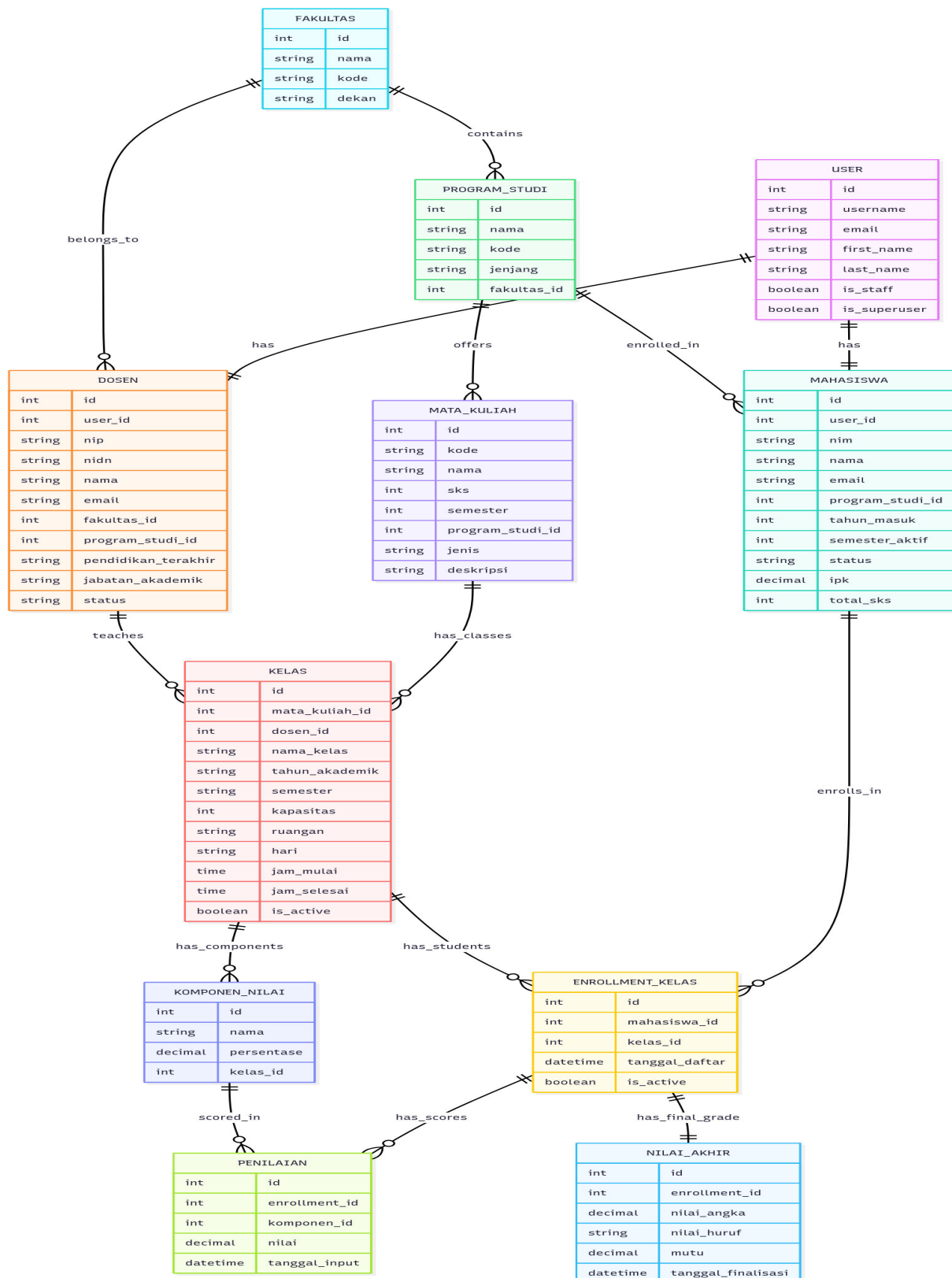
class Penilaian{
    id
    enrollment_id
    komponen_id
    nilai
    tanggal_input
}

class NilaiAkhir{
    id
```

```

    enrollment_id
    nilai_angka
    nilai_huruf
    mutu
    tanggal_finalisasi
  }

```



4. ENTITY RELATIONSHIP DIAGRAM

4.1 Mermaid Code - ERD

```

erDiagram
    USER ||--|| MAHASISWA : has
    USER ||--|| DOSEN : has
    FAKULTAS ||--o{ PROGRAM_STUDI : contains
    FAKULTAS ||--o{ DOSEN : belongs_to
    PROGRAM_STUDI ||--o{ MAHASISWA : enrolled_in
    PROGRAM_STUDI ||--o{ MATA_KULIAH : offers
    MATA_KULIAH ||--o{ KELAS : has_classes
    DOSEN ||--o{ KELAS : teaches
    MAHASISWA ||--o{ ENROLLMENT_KELAS : enrolls_in
    KELAS ||--o{ ENROLLMENT_KELAS : has_students
    KELAS ||--o{ KOMPONEN_NILAI : has_components
    ENROLLMENT_KELAS ||--o{ PENILAIAN : has_scores
    KOMPONEN_NILAI ||--o{ PENILAIAN : scored_in
    ENROLLMENT_KELAS ||--|| NILAI_AKHIR : has_final_grade

    USER {
        int id
        string username
        string email
        string first_name
        string last_name
        boolean is_staff
        boolean is_superuser
    }

    FAKULTAS {
        int id
        string nama
        string kode
        string dekan
    }

    PROGRAM_STUDI {
        int id
        string nama

```



```
    string kode
    string jenjang
    int fakultas_id
}

MAHASISWA {
    int id
    int user_id
    string nim
    string nama
    string email
    int program_studi_id
    int tahun_masuk
    int semester_aktif
    string status
    decimal ipk
    int total_sks
}

DOSEN {
    int id
    int user_id
    string nip
    string nidn
    string nama
    string email
    int fakultas_id
    int program_studi_id
    string pendidikan_terakhir
    string jabatan_akademik
    string status
}

MATA_KULIAH {
    int id
    string kode
    string nama
    int sks
    int semester
    int program_studi_id
    string jenis
    string deskripsi
```

```
}

KELAS {
    int id
    int mata_kuliah_id
    int dosen_id
    string nama_kelas
    string tahun_akademik
    string semester
    int kapasitas
    string ruangan
    string hari
    time jam_mulai
    time jam_selesai
    boolean is_active
}

ENROLLMENT_KELAS {
    int id
    int mahasiswa_id
    int kelas_id
    datetime tanggal_daftar
    boolean is_active
}

KOMPONEN_NILAI {
    int id
    string nama
    decimal persentase
    int kelas_id
}

PENILAIAN {
    int id
    int enrollment_id
    int komponen_id
    decimal nilai
    datetime tanggal_input
}

NILAI_AKHIR {
    int id
```

```
int enrollment_id
decimal nilai_angka
string nilai_huruf
decimal mutu
datetime tanggal_finalisasi
}
```

5. PENJELASAN ENTITAS

5.1 Core Entities

User

- **Purpose:** Django built-in model untuk autentikasi
- **Key Fields:** username, email, is_staff, is_superuser
- **Role:** Base authentication untuk semua user types

Fakultas

- **Purpose:** Organizational unit tingkat tertinggi
- **Key Fields:** nama, kode, dekan
- **Business Logic:** Satu fakultas memiliki banyak program studi

ProgramStudi

- **Purpose:** Program akademik dalam fakultas
- **Key Fields:** nama, kode, jenjang (D3/S1/S2/S3)
- **Business Logic:** Mengelompokkan mata kuliah dan mahasiswa

5.2 People Entities

Mahasiswa

- **Purpose:** Data mahasiswa dan progress akademik
- **Key Fields:** nim, nama, ipk, total_sks, semester_aktif
- **Business Logic:** Terkait dengan User untuk login access

- **Calculated Fields:** semester_saat_ini berdasarkan tahun_masuk

Dosen

- **Purpose:** Data dosen dan karir akademik
- **Key Fields:** nip, nidn, pendidikan_terakhir, jabatan_akademik
- **Business Logic:** Bisa mengajar di multiple program studi

5.3 Curriculum Entities

MataKuliah

- **Purpose:** Definisi mata kuliah dalam kurikulum
- **Key Fields:** kode, nama, sks, semester, jenis
- **Business Logic:** Memiliki prasyarat (self-referential)

Kelas

- **Purpose:** Instance mata kuliah untuk semester tertentu
- **Key Fields:** nama_kelas, tahun_akademik, kapasitas
- **Business Logic:** Kombinasi mata kuliah + dosen + jadwal

EnrollmentKelas

- **Purpose:** Junction table mahasiswa-kelas
- **Key Fields:** mahasiswa_id, kelas_id, is_active
- **Business Logic:** Many-to-many relationship resolver

5.4 Assessment Entities

KomponenNilai

- **Purpose:** Komponen penilaian per kelas
- **Key Fields:** nama, persentase
- **Examples:** UTS (30%), UAS (40%), Tugas (30%)

Penilaian

- **Purpose:** Nilai mahasiswa per komponen
- **Key Fields:** enrollment_id, komponen_id, nilai
- **Business Logic:** Unique constraint per enrollment-komponen

NilaiAkhir

- **Purpose:** Nilai final dengan konversi huruf dan mutu
 - **Key Fields:** nilai_angka, nilai_huruf, mutu
 - **Business Logic:** Auto-calculate dari penilaian komponen
-

6. RELATIONSHIP ANTAR ENTITAS

6.1 One-to-One Relationships

1. **User** ↔ **Mahasiswa**: Setiap mahasiswa memiliki satu user account
2. **User** ↔ **Dosen**: Setiap dosen memiliki satu user account
3. **EnrollmentKelas** ↔ **NilaiAkhir**: Setiap enrollment memiliki satu nilai akhir

6.2 One-to-Many Relationships

1. **Fakultas** → **ProgramStudi**: Satu fakultas memiliki banyak program studi
2. **ProgramStudi** → **Mahasiswa**: Satu prodi memiliki banyak mahasiswa
3. **ProgramStudi** → **MataKuliah**: Satu prodi menawarkan banyak mata kuliah
4. **MataKuliah** → **Kelas**: Satu mata kuliah memiliki banyak kelas
5. **Dosen** → **Kelas**: Satu dosen mengajar banyak kelas
6. **Kelas** → **KomponenNilai**: Satu kelas memiliki banyak komponen nilai

6.3 Many-to-Many Relationships

1. **Mahasiswa** ↔ **Kelas**: Via EnrollmentKelas
2. **MataKuliah** ↔ **MataKuliah**: Self-referential untuk prasyarat

6.4 Complex Relationships

1. **Assessment Chain**:

2. Kelas → KomponenNilai → Penilaian ← EnrollmentKelas

3. **Academic Hierarchy:**

4. Fakultas → ProgramStudi → MataKuliah → Kelas

7. BUSINESS RULES

7.1 Data Integrity Rules

1. **Unique Constraints:**

2. NIM mahasiswa harus unique
3. NIP dan NIDN dosen harus unique
4. Kode fakultas, program studi, mata kuliah harus unique
5. Email user harus unique

6. **Referential Integrity:**

7. Mahasiswa harus terdaftar di program studi yang valid
8. Kelas harus memiliki mata kuliah dan dosen yang valid
9. Penilaian hanya bisa dibuat untuk enrollment yang aktif

7.2 Academic Business Rules

1. **Enrollment Rules:**

2. Mahasiswa hanya bisa enroll ke kelas yang sesuai dengan program studinya
3. Kapasitas kelas tidak boleh dilampaui
4. Mahasiswa tidak bisa enroll ke mata kuliah yang sama di semester yang sama

5. **Grade Calculation Rules:**

6. Nilai akhir dihitung dari weighted average komponen nilai
7. Konversi nilai huruf otomatis berdasarkan nilai angka:
 - A: 85-100, B+: 80-84, B: 75-79, C+: 70-74
 - C: 65-69, D+: 60-64, D: 55-59, E: 0-54

8. $\text{Mutu} = \text{Bobot Nilai} \times \text{SKS}$

9. **IPK Calculation:**

10. $\text{IPK} = \Sigma(\text{Mutu}) / \Sigma(\text{SKS})$

11. Hanya mata kuliah dengan status lulus yang dihitung

7.3 Security Rules

1. **Access Control:**

2. Mahasiswa hanya bisa akses data pribadi
3. Dosen bisa akses data kelas yang diajar
4. Staff admin bisa akses data sesuai kewenangan
5. Super admin memiliki akses penuh

6. **Data Privacy:**

7. Data pribadi mahasiswa/dosen hanya bisa diakses oleh yang berwenang
 8. Audit trail untuk perubahan data sensitif
-

8. KESIMPULAN

8.1 Ringkasan Desain

Sistem Informasi Akademik telah dirancang dengan arsitektur database yang solid menggunakan 11 entitas utama dengan relationship yang well-defined. Desain ini mendukung:

1. **Scalability:** Struktur modular memungkinkan penambahan fitur
2. **Maintainability:** Clear separation of concerns antar entitas
3. **Data Integrity:** Comprehensive constraints dan business rules
4. **Security:** Role-based access control yang robust

8.2 Keunggulan Desain

1. **Normalized Database:** Mengurangi redundansi data

2. **Flexible Grading:** Support multiple komponen penilaian
3. **Academic Hierarchy:** Struktur fakultas-prodi yang clear
4. **Audit Capability:** Tracking perubahan data dengan timestamp

8.3 Rekomendasi Pengembangan

1. **Phase 1:** Implementasi core entities (User, Fakultas, ProgramStudi, Mahasiswa, Dosen)
2. **Phase 2:** Curriculum management (MataKuliah, Kelas, EnrollmentKelas)
3. **Phase 3:** Assessment system (KomponenNilai, Penilaian, NilaiAkhir)
4. **Phase 4:** Advanced features (prasyarat, laporan, analytics)

8.4 Maintenance Guidelines

1. **Regular Backup:** Schedule database backup harian
2. **Performance Monitoring:** Monitor query performance dan indexing
3. **Data Validation:** Regular data integrity checks
4. **Security Updates:** Regular security patch dan access review

Dokumen ini merupakan panduan teknis untuk pengembangan dan maintenance Sistem Informasi Akademik Perguruan Tinggi.

Versi: 1.0

Tanggal: Oktober 2025

Status: Final