**SOFTWARE DESIGN DOCUMENT**

**Di susun 12 April 2025**

**Kelompok 1**

**Sukran Golit ( B02220100 )**

**Maimanatul Karimah ( B02220122 )**

**Jurusan Ilmu Komputer  
Universitas Muhammadiyah BIMA  
2025/2026**

**1. PENDAHULUAN**

**1.1 Tujuan**

Dokumen ini bertujuan untuk memberikan deskripsi yang terperinci mengenai desain arsitektural dan teknis dari sistem manajemen data mahasiswa berbasis web yang sedang dikembangkan. Tujuan dari dokumen ini antara lain:

1. Menyediakan panduan teknis untuk implementasi dan pengembangan sistem, termasuk arsitektur perangkat keras dan perangkat lunak, serta kebutuhan fungsional dan non-fungsional.
2. Memastikan keselarasan antara kebutuhan pengguna (admin, dosen, mahasiswa, dan staf akademik) dengan hasil akhir yang diharapkan, untuk mencapai efektivitas dalam pengelolaan data mahasiswa secara efisien dan aman.
3. Menjadi acuan bagi tim pengembang dan pemangku kepentingan dalam memastikan bahwa setiap komponen sistem dirancang dan dibangun sesuai dengan standar yang ditetapkan.
4. Menjamin keamanan dan integritas data mahasiswa, dengan pendekatan yang tepat dalam hal otorisasi, enkripsi, dan backup data.
5. Membantu integrasi sistem dengan aplikasi atau sistem lain yang ada di lingkungan kampus, melalui antarmuka API yang jelas dan terdokumentasi.

**1.2 Lingkup Masalah**

**Ruang Lingkup**

Ruang lingkup dari sistem manajemen data mahasiswa ini mencakup berbagai fungsi dan fitur utama yang dirancang untuk mendukung pengelolaan data secara efektif, aman, dan terintegrasi. Adapun ruang lingkup sistem meliputi:

1. Autentikasi dan Otorisasi Pengguna  
   Sistem menyediakan mekanisme login aman dengan validasi kredensial, serta pengaturan hak akses berbasis peran (role-based access control) untuk memastikan bahwa setiap pengguna hanya dapat mengakses data dan fitur sesuai kewenangannya (misalnya: admin, dosen, mahasiswa).
2. Manajemen Data Berbasis CRUD  
   Sistem mendukung operasi Create, Read, Update, dan Delete untuk berbagai entitas penting seperti data mahasiswa, pengguna, program studi, dan riwayat akademik.
3. Pembuatan Laporan dan Analitik  
   Sistem menyediakan fitur untuk menghasilkan laporan akademik dan statistik data mahasiswa, baik dalam bentuk tabel maupun visualisasi grafik. Laporan dapat diekspor dalam format PDF dan Excel.
4. Integrasi API dengan Sistem Eksternal  
   Sistem mendukung API RESTful yang memungkinkan integrasi dengan aplikasi atau sistem lain di lingkungan kampus, seperti sistem akademik, keuangan, atau aplikasi mobile.
5. Mekanisme Pencadangan Data (Backup)  
   Sistem dilengkapi dengan fitur backup data berkala untuk menjamin keberlangsungan layanan dan mencegah kehilangan data akibat gangguan teknis atau kesalahan operasional.

**1.3 Referensi**

* IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.
* Standar Keamanan Data: ISO/IEC 27001

Dalam pengelolaan data mahasiswa, keamanan informasi menjadi aspek yang sangat penting. Sistem ini dirancang dengan mengacu pada standar internasional ISO/IEC 27001, yang merupakan standar untuk Sistem Manajemen Keamanan Informasi (ISMS). Standar ini menetapkan kerangka kerja yang sistematis dalam mengelola informasi sensitif, agar tetap aman melalui penerapan kebijakan, prosedur, kontrol teknis, dan organisasi.

Beberapa prinsip utama yang diadopsi dari ISO/IEC 27001 meliputi:

1. Kerahasiaan: Data mahasiswa hanya dapat diakses oleh pihak yang berwenang.
2. Integritas: Data dijaga agar tetap akurat dan tidak mengalami perubahan tanpa izin.
3. Ketersediaan: Data tersedia saat dibutuhkan oleh pengguna yang berwenang.

**2. Desain Arsitektur**

**2.1 Diagram Arsitektur Sistem**

Sistem manajemen data mahasiswa dikembangkan menggunakan arsitektur MVC (Model-View-Controller), yang memisahkan logika aplikasi, tampilan, dan pengelolaan data untuk meningkatkan modularitas, kemudahan pemeliharaan, dan skalabilitas sistem.

Adapun komponen teknologi yang digunakan dalam arsitektur sistem ini meliputi:

* Backend  
  Menggunakan PHP dengan Yii2 Framework, yang menyediakan struktur berbasis MVC dan fitur keamanan bawaan seperti RBAC (Role-Based Access Control), validasi input, dan enkripsi.
* Frontend  
  Antarmuka pengguna dibangun menggunakan HTML, CSS, dan Bootstrap 5, untuk menghasilkan tampilan yang responsif dan mudah diakses melalui berbagai perangkat, termasuk desktop dan mobile.
* Database  
  Sistem menyimpan seluruh data mahasiswa, pengguna, riwayat akademik, serta log aktivitas dalam MySQL Database, yang handal untuk pengelolaan data relasional dalam skala menengah hingga besar.
* API (Application Programming Interface)  
  Sistem menyediakan RESTful API yang memungkinkan integrasi dengan sistem lain, seperti:
  + Sistem akademik kampus
  + Sistem keuangan
  + Aplikasi mobile mahasiswa
* Caching  
  Untuk meningkatkan performa dan mempercepat proses akses data, sistem menggunakan Redis sebagai mekanisme caching pada data yang sering diakses, seperti daftar mahasiswa aktif atau data program studi.

**2.2 Komponen Utama Sistem**

Sistem manajemen data mahasiswa dirancang dengan pendekatan arsitektur Model-View-Controller (MVC), serta dilengkapi dengan beberapa komponen tambahan untuk mendukung kinerja dan skalabilitas sistem. Berikut ini adalah komponen-komponen utama yang membentuk sistem secara keseluruhan:

1. Model  
   Bertanggung jawab untuk menangani logika bisnis dan pengelolaan data, termasuk interaksi langsung dengan database. Model memproses data mahasiswa, data pengguna, riwayat akademik, dan entitas lain yang relevan.
2. View  
   Merupakan bagian dari sistem yang menyajikan antarmuka pengguna (UI). Dibangun menggunakan HTML, CSS, dan Bootstrap 5, View menampilkan data secara interaktif dan responsif agar mudah digunakan oleh admin, mahasiswa, dosen, dan staf akademik.
3. Controller  
   Mengatur aliran data antara Model dan View, serta menangani permintaan dari pengguna. Controller bertugas menerima input dari antarmuka pengguna, memprosesnya melalui Model, dan kemudian menampilkan hasilnya melalui View.
4. Database  
   Seluruh informasi penting disimpan dalam MySQL, termasuk:
   * Data mahasiswa (NIM, nama, program studi, status)
   * Akun pengguna dan hak akses
   * Riwayat nilai dan kehadiran
   * Log sistem dan histori aktivitas  
     Struktur database dirancang untuk menjaga integritas dan konsistensi data.
5. API Layer  
   Sistem menyediakan lapisan API berbasis RESTful untuk mendukung:
   * Integrasi dengan sistem eksternal kampus (akademik, keuangan, aplikasi mobile)
   * Pertukaran data secara aman dan terstruktur menggunakan format JSON
6. Caching Layer  
   Menggunakan Redis untuk menyimpan data yang sering diakses, seperti daftar mahasiswa aktif atau data program studi. Hal ini bertujuan untuk mempercepat waktu respon sistem dan mengurangi beban query ke database.

**3. Desain Modul dan Komponen**

**3.1 Modul Autentikasi Pengguna**

* Fungsi: Login, logout, registrasi, dan pemulihan kata sandi.
* Input: Username, password, token autentikasi.
* Output: Token sesi pengguna, status autentikasi.
* Teknologi: OAuth2 untuk keamanan autentikasi, JWT untuk token berbasis sesi.
* Validasi: Captcha untuk mencegah serangan brute force.

**3.2 Modul Manajemen Data**

Modul Manajemen Data merupakan salah satu komponen inti dalam sistem manajemen data mahasiswa yang bertanggung jawab terhadap pengelolaan berbagai entitas data yang digunakan dalam sistem. Modul ini mendukung berbagai operasi untuk memastikan data tetap akurat, mutakhir, dan mudah diakses.

Fungsi

Modul ini menyediakan fitur **CRUD (Create, Read, Update, Delete)** untuk berbagai entitas dalam sistem, seperti:

* Pengguna (admin, dosen, mahasiswa)
* Data mahasiswa (identitas, program studi, status akademik)
* Transaksi akademik (KRS, KHS, nilai)
* Dokumen dan arsip pendukung

**Input**

Data yang dimasukkan ke dalam modul ini meliputi:

* Nama lengkap pengguna
* Email dan username
* Peran pengguna (admin, dosen, mahasiswa)
* Informasi mahasiswa seperti NIM, program studi, angkatan, dan status

**Output**

* Data yang telah disimpan akan tersimpan secara permanen dalam database MySQL
* Data juga tersedia melalui API RESTful sebagai respons terhadap permintaan sistem eksternal atau aplikasi pengguna

**Teknologi Pendukung**

* ORM (Object-Relational Mapping) Yii2 digunakan untuk memudahkan interaksi antara model dan database, mempercepat pengembangan, dan menjaga keamanan query.
* Mekanisme indexing diterapkan pada kolom penting dalam tabel database (seperti NIM, nama, program studi) untuk mengoptimalkan pencarian dan filter data, terutama saat menangani data dalam jumlah besar.

**3.2 Modul Manajemen Data**

Modul Manajemen Data merupakan salah satu komponen inti dalam sistem manajemen data mahasiswa yang bertanggung jawab terhadap pengelolaan berbagai entitas data yang digunakan dalam sistem. Modul ini dirancang untuk mendukung berbagai operasi penting guna memastikan bahwa data tetap akurat, mutakhir, aman, dan mudah diakses oleh pengguna yang berwenang.

**Fungsi**

Modul ini menyediakan fitur **CRUD** (*Create, Read, Update, Delete*) untuk entitas-entitas utama dalam sistem, antara lain:

* Pengguna: mencakup admin, dosen, mahasiswa, dan staf akademik.
* Data Mahasiswa: seperti identitas pribadi, program studi, angkatan, dan status akademik.
* Transaksi Akademik: termasuk pengisian KRS, pencatatan nilai KHS, dan rekam jejak studi.
* Dokumen dan Arsip Pendukung: seperti surat keterangan, transkrip, dan dokumen administratif lainnya.

**Input**

Data yang dimasukkan ke dalam sistem melalui modul ini meliputi:

* Nama lengkap pengguna
* Email dan username
* Peran pengguna (admin, dosen, mahasiswa)
* Informasi mahasiswa: NIM, program studi, angkatan, status akademik
* Data transaksi: mata kuliah yang diambil, nilai, semester, dan file dokumen (jika ada)

**Output**

* Data yang telah diproses akan tersimpan secara permanen dalam database MySQL.
* Data juga tersedia dalam bentuk respons API RESTful, yang memungkinkan integrasi dengan sistem eksternal (seperti sistem akademik pusat atau aplikasi mobile).

**Teknologi Pendukung**

* ORM Yii2  
  Menggunakan Object-Relational Mapping (ORM) dari Yii2 untuk menyederhanakan interaksi antara model aplikasi dan database, serta menjaga keamanan dan efisiensi dalam penulisan query SQL.
* Mekanisme Indexing  
  Diterapkan pada kolom-kolom penting (seperti NIM, nama mahasiswa, dan program studi) guna meningkatkan performa pencarian data dan penyaringan informasi, terutama saat menangani data dalam jumlah besar.

**3.3 Modul Laporan dan Analitik**

* Fungsi: Pembuatan laporan berbasis data dengan visualisasi grafik.
* Output: File PDF atau Excel, grafik statistik.
* Teknologi: Library MPDF untuk PDF, PhpSpreadsheet untuk Excel, Chart.js untuk visualisasi data.

**3.4 Modul Integrasi API**

* Fungsi: Komunikasi dengan sistem eksternal seperti sistem pembayaran dan layanan pihak ketiga.
* Input: Data permintaan dalam format JSON.
* Output: Respons API dari layanan eksternal.
* Teknologi: Implementasi RESTful API dengan autentikasi berbasis API key.

**4. Desain Basis Data**

**4.1 Model Data**

Struktur utama basis data mencakup tabel:

1. users (id, nama, email, password, role, created\_at, updated\_at).

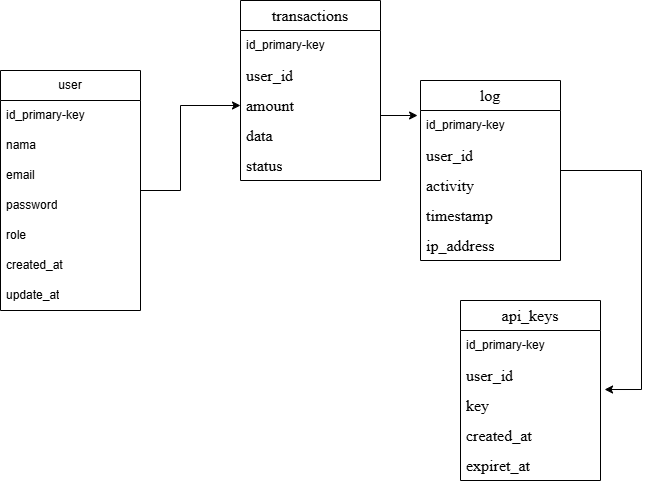
2. transactions (id, user\_id, amount, date, status).

3. logs (id, user\_id, activity, timestamp, ip\_address).

4. api\_keys (id, user\_id, key, created\_at, expired\_at).

**4.2 Diagram Entity-Relationship (ERD)**

(Diagram ERD dapat disertakan sebagai gambar yang menunjukkan relasi antar tabel. Manajemen Data Mahasiswa)



**4.3 Skema Indexing dan Optimasi**

* Indexing pada kolom pencarian utama untuk meningkatkan performa query.
* Partisi tabel transaksi berdasarkan periode untuk optimasi pengambilan data.
* Backup otomatis setiap 24 jam untuk memastikan pemulihan data jika terjadi kegagalan.

**5. Antarmuka Pengguna**

**5.1 Wireframe Desain**

* Halaman Login: Form input username dan password dengan opsi autentikasi dua faktor. • Dashboard: Menampilkan ringkasan data dan laporan interaktif dengan filter. • Halaman CRUD: Form input dan tabel data dengan fitur sorting dan pagination.

**5.2 Desain Responsif**

* Menggunakan Bootstrap 5 untuk memastikan kompatibilitas dengan berbagai perangkat.
* Layout fleksibel dengan grid system untuk tampilan yang lebih dinamis.
* Penggunaan AJAX untuk mempercepat interaksi pengguna tanpa perlu reload halaman.

**6. Pertimbangan Keamanan**

* Enkripsi Data: Menggunakan AES-256 untuk menyimpan data sensitif.
* Kontrol Akses: Implementasi Role-Based Access Control (RBAC) untuk membatasi hak akses pengguna.
* Proteksi API: Penggunaan API key dan token autentikasi untuk mencegah akses tidak sah.
* Audit Log: Penyimpanan semua aktivitas pengguna untuk keamanan dan kepatuhan.
* Firewall Aplikasi Web (WAF): Mencegah serangan SQL Injection dan XSS.

**7. Pengujian dan Validasi**

**7.1 Pengujian Unit**

* Pengujian tiap modul sistem dengan PHPUnit.
* Simulasi skenario input valid dan tidak valid.

**7.2 Pengujian Keamanan**

* Pengujian penetrasi untuk mengidentifikasi kerentanan sistem.
* Uji beban untuk memastikan sistem dapat menangani lonjakan pengguna.

**7.3 Uji Fungsionalitas**

* Pengujian fitur utama sistem berdasarkan spesifikasi kebutuhan.
* Penggunaan Selenium untuk mengotomatiskan pengujian UI.