**Dokumen Desain Perangkat Lunak (Software**

**Design Document - SDD)**

**1.Pendahuluan**

* 1. **Tujuan**

Dokumen ini bertujuan untuk memberikan deskripsi terperinci mengenai desain arsitektural dan teknis dari sistem *Website KRS Online* yang sedang dikembangkan. Dokumen ini akan digunakan sebagai panduan dalam implementasi dan pengembangan sistem, memastikan keselarasan antara kebutuhan fungsional dan non-fungsional yang telah ditetapkan dalam dokumen SRS (Software Requirement Specification) dengan hasil akhir yang diharapkan.

* 1. **Ruang Lingkup**

Sistem informasi ini dirancang untuk mengelola data pengguna secara efisien, mencakup fitur

utama seperti:

* **Autentikasi dan Otorisasi Pengguna** Sistem memungkinkan Mahasiswa, Dosen, dan Admin untuk login menggunakan kredensial yang sesuai, dengan pengaturan hak akses berbasis peran untuk memastikan keamanan data dan fungsi yang tepat.
* **Manajemen Data Pengguna dan KRS** Sistem menyediakan fitur untuk mengelola data mahasiswa, dosen, mata kuliah, dan jadwal kuliah, serta memungkinkan Mahasiswa untuk mengisi dan mengajukan KRS yang dapat disetujui atau direvisi oleh Dosen.
* **Integrasi API untuk Komunikasi dengan Sistem Eksternal** Sistem menyediakan API berbasis RESTful yang memungkinkan integrasi dengan Sistem Informasi Akademik (SIA) atau aplikasi eksternal lainnya untuk sinkronisasi data.
  1. **Referensi**
* IEEE 1016-2009: Standard for Software Design Description.
* Dokumentasi CodeIgniter 4 (CI4) Framework.
* Panduan Bootstrap 5 untuk desain antarmuka pengguna.
* Panduan keamanan OWASP untuk mengamankan aplikasi web.

**2. Desain Arsitektur**

**2.1 Diagram Arsitektur Sistem**

Sistem ini menggunakan arsitektur berbasis MVC (Model-View-Controller) dengan teknologi

berikut:

* Backend: PHP CodeIgniter 4 (CI4) Framework.
* Frontend: HTML, CSS, Bootstrap 5.
* Database: MySQL.
* API: RESTful API untuk integrasi sistem.
* Caching: Redis untuk meningkatkan performa sistem.

**2.2 Komponen Utama Sistem**

* **Model** Bertanggung jawab atas pengelolaan data dan interaksi dengan database. Model akan menangani operasi CRUD (Create, Read, Update, Delete) untuk entitas seperti data mahasiswa, dosen, mata kuliah, jadwal kuliah, dan KRS, serta memastikan konsistensi dan integritas data yang disimpan dalam sistem.
* **View** Menyediakan tampilan antarmuka pengguna yang responsif dan interaktif. View akan dirancang untuk memberikan pengalaman pengguna yang baik, dengan akses yang mudah ke berbagai fitur seperti pengisian KRS, persetujuan dosen, dan laporan data akademik. Tampilan ini akan dapat diakses melalui perangkat desktop dan mobile.
* **Controller** Mengatur aliran data antara Model dan View. Controller akan menerima permintaan dari pengguna, memprosesnya, dan mengirimkan hasilnya kembali ke tampilan yang sesuai. Ini juga mencakup pengaturan autentikasi pengguna dan hak akses berbasis peran (Role-Based Access Control).
* **Database (MySQL)** Merupakan struktur data yang digunakan untuk menyimpan informasi penting seperti data pengguna (mahasiswa, dosen), data mata kuliah, jadwal kuliah, dan status KRS. MySQL digunakan untuk memastikan konsistensi dan keandalan data, serta mendukung transaksi yang diperlukan dalam aplikasi.
* **API Layer** Digunakan untuk komunikasi antar sistem dan mendukung integrasi dengan sistem eksternal. API berbasis RESTful akan menyediakan endpoint untuk integrasi dengan Sistem Informasi Akademik (SIA) dan aplikasi lain yang digunakan untuk sinkronisasi data mahasiswa, dosen, dan mata kuliah.

**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**

* **Fungsi:** Modul ini mengelola proses CRUD (Create, Read, Update, Delete) untuk berbagai entitas penting dalam sistem KRS Online, seperti data mahasiswa, dosen, mata kuliah, jadwal kuliah, dan data pengisian KRS.
* **Input:** 
  + Data Mahasiswa: NIM, nama, program studi, angkatan, dsb.
  + Data Dosen: NIDN, nama, jabatan, mata kuliah yang diampu.
  + Data Mata Kuliah: Kode MK, nama MK, jumlah SKS, semester.
  + Data KRS: NIM mahasiswa, mata kuliah yang dipilih, status pengajuan.
  + Data Jadwal: Hari, jam, ruang kuliah, dosen pengampu.
* **Output:** 
  + Penyimpanan data ke dalam database MySQL.
  + Respons API untuk konsumsi internal dan eksternal (misalnya untuk menampilkan data KRS mahasiswa atau laporan mata kuliah yang diambil).
  + Notifikasi sistem untuk aksi tertentu (misalnya saat KRS berhasil diajukan atau disetujui).
* **Teknologi:** 
  + **CodeIgniter 4 ORM (Model CI4)** digunakan untuk mengelola interaksi antara sistem dan database secara terstruktur.
  + **Indexing Database (MySQL)** diterapkan pada kolom penting (seperti NIM, Kode Mata Kuliah) untuk mengoptimalkan performa pencarian dan kueri data yang kompleks.
  1. **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, username, nama, email, password, role, created\_at, updated\_at).

2. mahasiswa (id, nim, nama, prodi, angkatan, email, telepon, alamat).

3. dosen (id, nidn, nama, email, telepon, jabatan, prodi).

4. mata\_kuliah (id, kode\_mk, nama\_mk, sks, smester, prodi).

5. jadwal\_kuliah (id, mata\_kuliah\_id, dosen\_id, hari, jam, ruangan, semester, tahun\_ajaran).

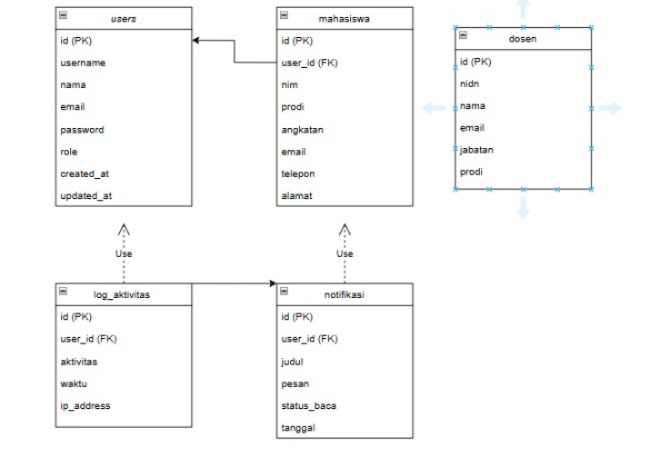
6. krs (id, mahasiswa\_id, mata\_kuliah\_id, semester, tahun\_ajaran, status, tanggal\_pengisian).

7. persetujuan\_krs (id, krs\_id, dosen\_id, status, catatan, tanggal\_persetujuan).

8. log\_aktivitas (id, user\_id, aktivitas, waktu, ip\_address).

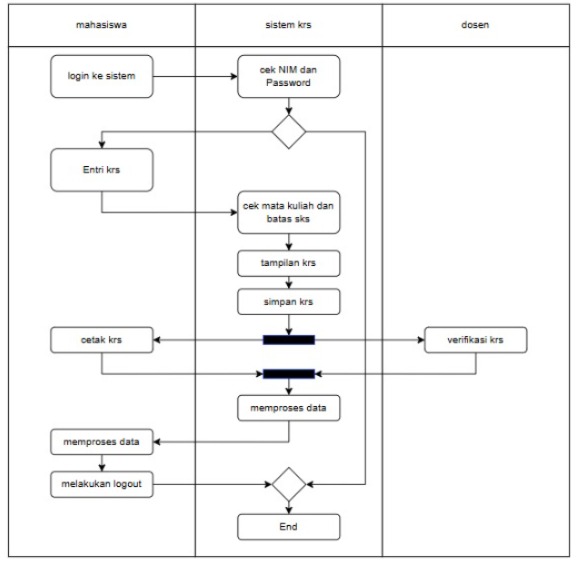
9. notifikasi (id, user\_id, judul, pesan, status\_baca, tanggal).

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



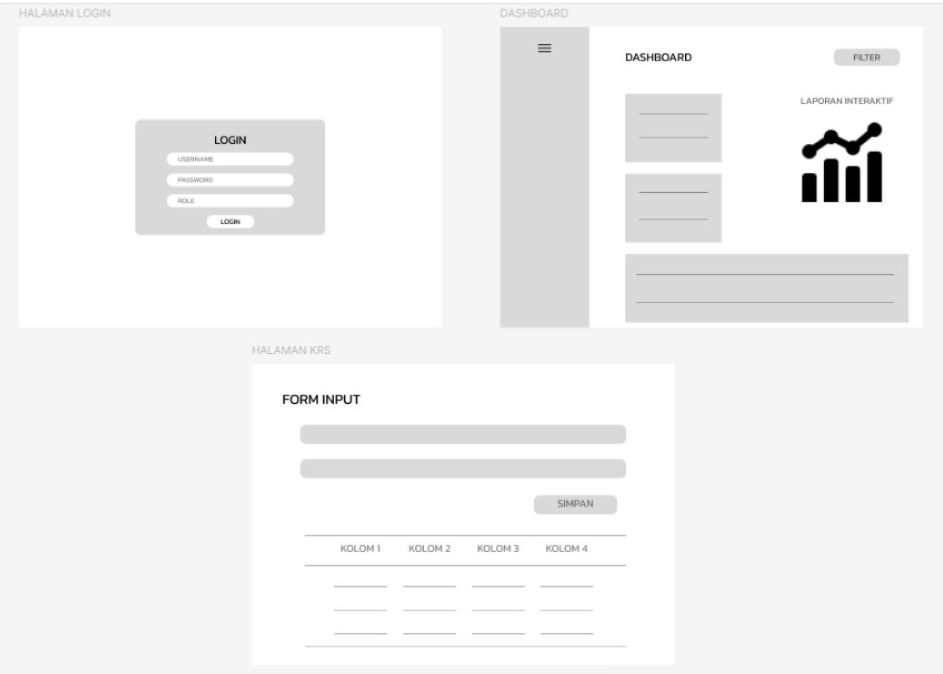


**4.3 Activity Diagram**



**5. Antarmuka Pengguna**

**5.1 Wireframe Desain**



**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.

**8. Lampiran**

* Diagram kelas dan urutan proses sistem.
* Dokumentasi API untuk integrasi eksternal.
* Laporan hasil pengujian dan rekomendasi perbaikan.