**Dokumen Desain Perangkat Lunak - Modul Perkuliahan dan Jadwal**

**1. Pendahuluan**

**1.1 Tujuan**

Dokumen ini bertujuan untuk memberikan deskripsi terperinci mengenai desain arsitektural dan teknis dari modul Perkuliahan dan Jadwal pada sistem informasi akademik berbasis web. Dokumen ini akan digunakan sebagai panduan dalam implementasi dan pengembangan sistem, memastikan keselarasan antara kebutuhan pengguna dan hasil akhir yang diharapkan.

**1.2 Ruang Lingkup**

Modul Perkuliahan dan Jadwal dirancang untuk mengelola data akademik secara efisien, mencakup fitur utama seperti:

* Pengelolaan kurikulum dan mata kuliah
* Penyusunan jadwal perkuliahan
* Pengaturan kelas dan ruang kuliah
* Pencarian jadwal dan pembuatan laporan
* Pengecekan konflik jadwal
* Integrasi dengan modul lain dalam sistem akademik

**1.3 Referensi**

* IEEE 1016-2009: Standard for Software Design Description
* Dokumentasi CodeIgniter 4 Framework
* Panduan Bootstrap 5 untuk desain antarmuka pengguna
* Peraturan akademik universitas

**2. Desain Arsitektur**

**2.1 Diagram Arsitektur Sistem**

Modul ini menggunakan arsitektur berbasis MVC (Model-View-Controller) dengan teknologi berikut:

* Backend: PHP CodeIgniter 4 Framework
* Frontend: HTML, CSS, Bootstrap 5
* Database: MySQL
* API: RESTful API untuk integrasi sistem

**2.2 Komponen Utama Sistem**

1. **Model**: Bertanggung jawab atas pengelolaan data dan interaksi dengan database (kurikulum, mata kuliah, jadwal, ruangan)
2. **View**: Menyediakan tampilan antarmuka pengguna yang responsif dan interaktif
3. **Controller**: Mengatur aliran data antara model dan view
4. **Database**: Struktur data yang disimpan dalam MySQL untuk memastikan konsistensi data
5. **API Layer**: Digunakan untuk komunikasi antar sistem dan mendukung integrasi dengan modul lain seperti Manajemen Dosen dan Mahasiswa
6. **Caching Layer**: Menggunakan Redis untuk mempercepat pengambilan data jadwal yang sering diakses

**3. Desain Modul dan Komponen**

**3.1 Modul Pengelolaan Kurikulum**

* **Fungsi**: Manajemen kurikulum, mata kuliah, prasyarat, dan bobot SKS
* **Input**: Data kurikulum (kode, nama, tahun), mata kuliah (kode, nama, SKS, deskripsi)
* **Output**: Data tersimpan dalam database, tampilan kurikulum
* **Teknologi**: ORM CodeIgniter untuk interaksi database
* **Validasi**: Validasi duplikasi kode mata kuliah, validasi prasyarat

**3.2 Modul Penyusunan Jadwal**

* **Fungsi**: Pembuatan jadwal perkuliahan, pengecekan konflik jadwal
* **Input**: Data mata kuliah, dosen, ruangan, waktu
* **Output**: Jadwal tersimpan, notifikasi konflik
* **Teknologi**: Algoritma pengecekan konflik jadwal, notifikasi email
* **Validasi**: Verifikasi ketersediaan ruangan dan dosen, kapasitas ruangan

**3.3 Modul Pengaturan Ruangan**

* **Fungsi**: Pengelolaan data ruangan, kapasitas, dan penggunaan
* **Input**: Data ruangan (kode, nama, lokasi, kapasitas)
* **Output**: Data tersimpan, tampilan penggunaan ruangan
* **Teknologi**: Dashboard visualisasi penggunaan ruangan, integrasi dengan modul jadwal
* **Validasi**: Pengecekan kapasitas terhadap jumlah peserta kelas

**3.4 Modul Laporan dan Pencarian**

* **Fungsi**: Pencarian jadwal, pembuatan laporan beban dosen dan penggunaan ruangan
* **Input**: Parameter pencarian (dosen, mata kuliah, ruangan, waktu)
* **Output**: Hasil pencarian, laporan PDF/Excel
* **Teknologi**: Library MPDF untuk PDF, PhpSpreadsheet untuk Excel, Chart.js untuk visualisasi data
* **Integrasi**: Google Calendar untuk sinkronisasi jadwal

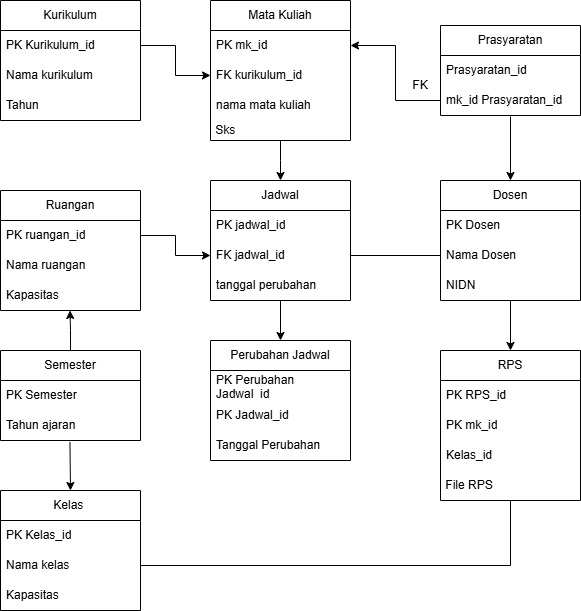
**4. Desain Basis Data**

**4.1 Model Data**

Struktur utama basis data mencakup tabel:

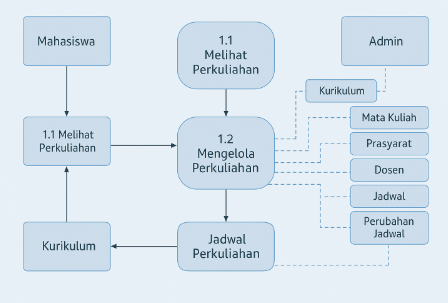
1. kurikulum (id, kode, nama, tahun\_aktif, status)
2. mata\_kuliah (id, kode\_mk, nama\_mk, sks, deskripsi, kurikulum\_id)
3. prasyarat (id, mk\_id, prasyarat\_mk\_id)
4. jadwal (id, mk\_id, dosen\_id, ruangan\_id, hari, jam\_mulai, jam\_selesai, semester\_id)
5. ruangan (id, kode, nama, lokasi, kapasitas)
6. semester (id, nama, tahun\_ajaran, status)
7. dosen (id, nip, nama, email)
8. kelas (id, kode, jadwal\_id, kapasitas)
9. rps (id, mk\_id, dokumen, tanggal\_upload)
10. perubahan\_jadwal (id, jadwal\_id, tanggal, alasan, status)

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



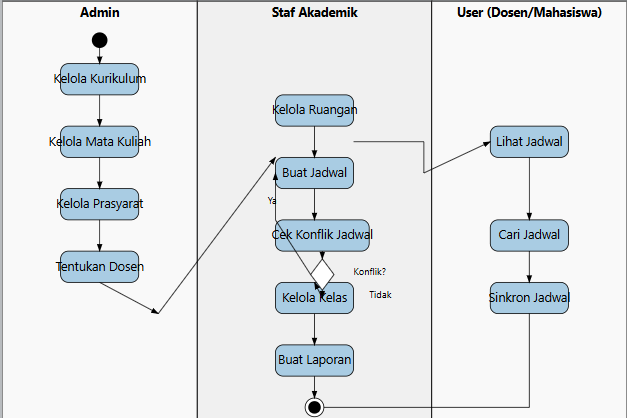
Gambar 1. Entity Relationship Diagram

**4.3 Diagram DFD**

****

Gambar 2. Data Flow diagram

**4.4 Diagram Aktifity**

****

Gambar 3. Activity Diagram

**4.5 Skema Indexing dan Optimasi**

* Indexing pada kolom pencarian utama seperti kode\_mk, nama\_mk untuk meningkatkan performa query
* Indexing pada kolom dosen\_id, ruangan\_id, hari untuk optimasi pencarian jadwal
* Partisi tabel jadwal berdasarkan semester untuk optimasi pengambilan data historis
* Backup otomatis setiap 24 jam untuk memastikan pemulihan data jika terjadi kegagalan

**5. Antarmuka Pengguna**

**5.1 Wireframe Desain**

* **Halaman Pengelolaan Kurikulum**: Form input dan tabel data kurikulum dengan fitur filtering
* **Halaman Pembuatan Jadwal**: Tampilan kalender dan timetable dengan drag-and-drop functionality
* **Dashboard Penggunaan Ruangan**: Visualisasi penggunaan ruangan dengan grafik dan diagram
* **Form Pencarian Jadwal**: Filter berdasarkan dosen, mata kuliah, ruangan dan waktu

**5.2 Desain Responsif**

* Menggunakan Bootstrap 5 untuk memastikan kompatibilitas dengan berbagai perangkat
* Layout fleksibel dengan grid system untuk tampilan yang lebih dinamis
* Optimasi untuk perangkat mobile dengan media queries dan responsive breakpoints

**6. Pertimbangan Keamanan**

* **Enkripsi Data**: Menggunakan AES-256 untuk menyimpan data sensitif
* **Kontrol Akses**: Implementasi Role-Based Access Control (RBAC) dengan peran yang terdefinisi (Admin Akademik, Staf, Dosen, Mahasiswa, Pimpinan)
* **Validasi Input**: Sanitasi input untuk mencegah serangan XSS dan SQL Injection
* **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 umum seperti 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
* Pengujian validasi prasyarat mata kuliah

**7.2 Pengujian Keamanan**

* Pengujian penetrasi untuk mengidentifikasi kerentanan sistem
* Uji beban untuk memastikan sistem dapat menangani lonjakan pengguna
* Simulasi serangan untuk menguji ketahanan sistem

**7.3 Uji Fungsionalitas**

* Verifikasi konflik jadwal berfungsi dengan baik
* Validasi pengaturan ruangan berdasarkan kapasitas
* Penggunaan Selenium untuk mengotomatiskan pengujian UI
* Pengujian integrasi dengan modul Manajemen Dosen dan Mahasiswa

**8. Performa dan Optimasi**

**8.1 Kebutuhan Performa**

* Waktu respons kurang dari 2 detik untuk operasi standar
* Mampu mendukung minimal 1000 pengguna bersamaan
* Pembuatan jadwal secara menyeluruh selesai dalam 5 menit

**8.2 Strategi Optimasi**

* Implementasi caching dengan Redis untuk data yang sering diakses
* Query optimization dengan melakukan indexing pada kolom relevan
* Penggunaan content delivery network (CDN) untuk aset statis
* Lazy loading untuk komponen UI yang tidak langsung terlihat

**9. Lampiran**

* Diagram kelas dan urutan proses penjadwalan
* Dokumentasi API untuk integrasi dengan modul lain
* Template laporan penggunaan ruangan dan beban dosen
* Prosedur pencadangan dan pemulihan data