Dokumen Desain Perangkat Lunak (Software Design Document - SDD)

Registrasi Mata kuliah-KRS

**1.Pendahuluan**

Dokumen desain software (SDD) adalah dokumen cetak biru yang digunakan dalam pengembangan perangkat lunak. Dokumen ini memberikan gambaran mendalam tentang bagaimana sistem akan dirancang untuk memenuhi kebutuhan pengguna dan tujuan proyek. Tujuan dari SDD untuk modul registrasi mata kuliah (KRS) adalah untuk mendefinisikan struktur, alur kerja, dan elemen yang diperlukan untuk membantu mahasiswa memilih mata kuliah secara efektif dan terorganisir.

**1.1 Tujuan**

Sehingga semua anggota tim, termasuk pengembang dan pemangku kepentingan, dapat memahami lebih baik ruang lingkup proyek, SDD memberikan gambaran yang jelas tentang struktur, fungsionalitas, dan keputusan desain sistem. Kedua, dokumen ini menetapkan standar pengkodean dan praktik terbaik yang harus diikuti oleh seluruh tim untuk memastikan bahwa semua komponen sistem KRS dibangun secara konsisten dan memudahkan pemeliharaan di masa depan.

SDD juga membantu berkomunikasi dengan berbagai pihak proyek, meningkatkan kerja sama dan mengurangi risiko kesalahpahaman tentang fungsi dan tujuan sistem. Tim pengembang dapat mengurangi risiko selama fase pengembangan dengan mengidentifikasi masalah yang mungkin terjadi dan membuat cara untuk mengatasinya dalam SDD. Selain itu, dokumen dapat dibagikan kepada klien dan pemangku kepentingan untuk meningkatkan transparansi dalam proses pengembangan, membantu mengelola ekspektasi, dan memberikan umpan balik yang bermanfaat.

Terakhir, SDD berfungsi sebagai referensi penting untuk perubahan di masa depan dalam siklus hidup perangkat lunak, memungkinkan tim untuk melacak versi perangkat lunak dan dokumentasi yang terkait dengan pengembangan KRS. Dengan demikian, SDD tidak hanya berfungsi sebagai panduan teknis tetapi juga sebagai alat untuk memastikan keberhasilan proyek melalui kerja sama yang baik dan pengurangan risiko.

**1.2 Ruang Lingkup**

Ruang lingkup Dokumen Desain Perangkat Lunak (SDD) untuk modul registrasi mata kuliah atau Kartu Rencana Studi (KRS) mencakup berbagai aspek yang akan diatur dan dijelaskan dalam pengembangan sistem ini. Berikut adalah rincian ruang lingkupnya:

1. Deskripsi Umum Sistem

Ruang lingkup ini mencakup penjelasan mengenai tujuan utama dari sistem registrasi mata kuliah, termasuk fungsionalitas yang akan disediakan untuk pengguna, seperti pendaftaran mata kuliah, pengelolaan KRS, dan pemantauan status pendaftaran.

2. Fungsionalitas Utama

SDD akan menjelaskan fungsionalitas utama yang harus ada dalam modul KRS, seperti:

- Pendaftaran mata kuliah oleh mahasiswa.

- Pengeditan dan pembatalan pendaftaran mata kuliah.

- Penjadwalan dan pengelolaan waktu perkuliahan.

- Penyimpanan dan pengambilan data KRS oleh mahasiswa dan dosen.

3. Antarmuka Pengguna

Ruang lingkup ini mencakup desain antarmuka pengguna (UI) yang akan digunakan oleh mahasiswa dan dosen, termasuk elemen-elemen visual, navigasi, dan interaksi pengguna dengan sistem.

4. Arsitektur Sistem

SDD akan menjelaskan arsitektur teknis dari sistem KRS, termasuk komponen perangkat lunak yang terlibat, database yang digunakan, serta integrasi dengan sistem lain jika diperlukan.

5. Keamanan dan Privasi Data

Dalam ruang lingkup ini, akan dibahas langkah-langkah keamanan yang akan diterapkan untuk melindungi data pribadi mahasiswa dan informasi akademik lainnya, termasuk autentikasi pengguna dan enkripsi data.

6. Batasan Sistem

SDD juga akan mencakup batasan-batasan sistem, seperti platform yang didukung (web atau mobile), keterbatasan dalam jumlah pengguna simultan, serta batasan terkait integrasi dengan sistem lain di institusi pendidikan.

**1.3 Referensi**

• IEEE 1016-2009: Standard for Software Design Description.

• Dokumentasi Yii2 Framework.

• Panduan Bootstrap 5 untuk desain antarmuka pengguna.

• Panduan keamanan OWASP untuk mengamankan aplikasi web.

**2. Desain Arsitektur**

**2.1 Diagram Arsitektur Sistem**

Untuk menggambarkan Diagram Arsitektur Sistem untuk modul registrasi mata kuliah (KRS), berikut adalah deskripsi yang mencakup komponen utama dan interaksi dalam sistem tersebut.

Komponen Utama dalam Arsitektur Sistem:

1. Pengguna

- Mahasiswa: Mengakses sistem untuk mendaftar, mengedit, dan mencetak KRS.

- Dosen: Memvalidasi KRS yang diajukan oleh mahasiswa.

- Admin (Akademik): Mengelola data pengguna, mata kuliah, dan proses registrasi.

2. Antarmuka Pengguna

- Halaman login untuk mahasiswa dan dosen.

- Dashboard untuk mahasiswa untuk mengelola KRS.

- Halaman admin untuk mengelola data akademik.

3. Server Aplikasi

- Menangani logika bisnis dan pemrosesan data.

- Berfungsi sebagai penghubung antara antarmuka pengguna dan database.

4. Database

- Menyimpan data pengguna, mata kuliah, dan KRS.

- Mengelola akses dan keamanan data.

Interaksi dalam Sistem:

- Mahasiswa melakukan login ke sistem melalui antarmuka pengguna.

- Setelah login, mahasiswa dapat memilih mata kuliah yang ingin diambil dan mengisi KRS.

- Data KRS yang diisi akan dikirim ke server aplikasi untuk diproses.

- Server aplikasi menyimpan data ke dalam database dan memberikan umpan balik kepada mahasiswa.

- Dosen dapat mengakses KRS yang diajukan oleh mahasiswa untuk melakukan validasi.

- Admin mengelola data pengguna dan mata kuliah melalui antarmuka admin yang terhubung dengan server aplikasi.

Berikut adalah representasi visual dari arsitektur sistem:

+------------------+

| Mahasiswa |

| |

| +------------+ |

| | Login | |

| +------------+ |

| |

| +------------+ |

| | KRS Form | |

| +------------+ |

+--------+---------+

|

| (Request)

v

+------------------+

| Server Aplikasi|

| |

| +--------------+ |

| | Logika Bisnis| |

| +--------------+ |

| |

+--------+---------+

|

| (Data Access)

v

+------------------+

| Database |

| |

| +--------------+ |

| | User Data | |

| +--------------+ |

| | Course Data | |

| +--------------+ |

| | KRS Data | |

| +--------------+ |

+------------------+

**2.2 komponen utama system**

Komponen utama dalam sistem registrasi mata kuliah (KRS) mencakup beberapa elemen penting yang saling terintegrasi untuk memastikan proses registrasi berjalan lancar dan efisien. Berikut adalah komponen utamanya:

1. Pengguna Sistem:

- Mahasiswa: Menggunakan sistem untuk memilih, mendaftarkan, atau mengedit mata kuliah yang akan diambil.

- Dosen: Memvalidasi KRS yang diajukan oleh mahasiswa sesuai dengan kurikulum.

- Admin Akademik: Mengelola data pengguna, mata kuliah, dan jadwal akademik.

2. Antarmuka Pengguna (User Interface):

- Menyediakan halaman login, menu registrasi mata kuliah, dan dashboard untuk mahasiswa, dosen, serta admin.

- Antarmuka ini harus ramah pengguna (user-friendly) dan mendukung akses melalui berbagai perangkat.

3. Server Aplikasi:

- Berfungsi sebagai pusat pemrosesan logika bisnis, seperti validasi data KRS, pengelolaan jadwal mata kuliah, dan pembuatan laporan akademik.

- Menangani komunikasi antara antarmuka pengguna dan database.

4. Database:

- Menyimpan data penting seperti informasi mahasiswa, daftar mata kuliah, jadwal perkuliahan, serta data KRS.

- Berperan dalam pengelolaan data secara terstruktur untuk mendukung pencarian dan pelaporan.

5. Keamanan Sistem:

- Mencakup autentikasi pengguna (login dengan NIM dan password), enkripsi data sensitif, serta pengaturan hak akses untuk memastikan keamanan data akademik.

**3. Desain Modul dan komponen**

**3.1 Modul Autentikasi pengguna**

Modul autentifikasi pengguna dalam sistem registrasi mata kuliah (KRS) memiliki beberapa komponen dan fungsi penting yang memastikan keamanan dan aksesibilitas bagi pengguna. Berikut adalah penjelasan mengenai komponen utama dan proses autentifikasi dalam modul ini:

1. Input Pengguna:

- Username/NIM: Mahasiswa menggunakan Nomor Induk Mahasiswa (NIM) sebagai username untuk masuk ke sistem.

- Password: Mahasiswa harus memasukkan password yang telah ditentukan untuk mengakses akun mereka.

2. Validasi Autentifikasi:

- Sistem memverifikasi kombinasi username dan password yang dimasukkan oleh pengguna. Jika valid, pengguna akan diberikan akses ke dashboard KRS; jika tidak, sistem akan menampilkan pesan kesalahan.

3. Keamanan Tambahan:

- Kode Keamanan: Beberapa sistem menerapkan kode keamanan tambahan yang harus diisi oleh pengguna untuk meningkatkan keamanan saat login.

- Pengelolaan Password: Pengguna dapat mengubah password mereka melalui antarmuka yang disediakan, dan ada prosedur untuk reset password jika pengguna lupa.

4. Riwayat Login:

- Fitur ini mencatat aktivitas login pengguna, termasuk waktu dan perangkat yang digunakan, untuk meningkatkan keamanan akun dan mendeteksi akses yang tidak sah.

5. Antarmuka Pengguna:

- Halaman login yang dirancang dengan baik untuk memudahkan mahasiswa dalam memasukkan informasi autentifikasi mereka.

Proses Autentifikasi Pengguna:

1. Login:

- Pengguna mengakses halaman login dan memasukkan NIM serta password mereka.

- Setelah mengisi semua kolom yang diperlukan, pengguna menekan tombol "Login".

2. Validasi Sistem:

- Sistem memeriksa database untuk memastikan bahwa kombinasi username dan password cocok.

- Jika validasi berhasil, pengguna diarahkan ke dashboard KRS; jika gagal, pesan kesalahan ditampilkan.

3. Pengelolaan Akun:

- Setelah login, pengguna dapat mengelola akun mereka, termasuk mengubah password dan memperbarui informasi profil.

4. Logout:

- Pengguna dapat keluar dari sistem dengan menekan tombol logout untuk menjaga keamanan akun mereka setelah selesai menggunakan sistem.

**3.2 Modul manajemen data**

Modul manajemen data dalam sistem informasi akademik, seperti Kartu Rencana Studi (KRS), berfungsi untuk mengelola dan memproses berbagai jenis data yang berkaitan dengan kegiatan akademik. Berikut adalah komponen dan desain yang dapat diterapkan dalam modul manajemen data ini:

1. Komponen Utama Modul Manajemen Data

- Database Pengguna: Menyimpan informasi pengguna, termasuk data mahasiswa, dosen, dan staf akademik. Data yang disimpan meliputi:

- Identitas pribadi (nama, NIM/NIDN, email, dll.)

- Riwayat pendidikan dan status akademik.

- Pengelolaan Data Mata Kuliah: Mengelola informasi mengenai mata kuliah yang ditawarkan, termasuk:

- Kode mata kuliah, nama mata kuliah, deskripsi, jumlah SKS, dan prasyarat.

- Jadwal kuliah dan pengaturan kelas.

- Pengelolaan Kartu Rencana Studi (KRS): Memfasilitasi mahasiswa dalam mengisi dan mengelola KRS mereka. Fitur yang disediakan termasuk:

- Formulir pengisian KRS online.

- Validasi prasyarat mata kuliah.

- Penyimpanan dan pengunduhan KRS yang telah disetujui.

- Pengelolaan Data Penilaian: Menyimpan dan mengelola data penilaian mahasiswa, termasuk:

- Nilai akhir dari setiap mata kuliah.

- Laporan hasil studi (KHS) dan transkrip.

2. Proses Pengelolaan Data

- Input Data: Fitur untuk menambah atau memperbarui data mahasiswa, dosen, dan mata kuliah melalui antarmuka pengguna yang ramah.

- Validasi Data: Memastikan bahwa semua data yang dimasukkan memenuhi kriteria yang ditetapkan (misalnya, validasi format email, keunikan NIM/NIDN).

- Sinkronisasi Data: Proses untuk menyinkronkan data dengan sistem eksternal atau database lain jika diperlukan.

3. Keamanan Data

- Otentikasi Pengguna: Mengimplementasikan sistem autentikasi untuk memastikan bahwa hanya pengguna terotorisasi yang dapat mengakses dan mengelola data.

- Enkripsi Data: Menggunakan enkripsi untuk melindungi data sensitif saat disimpan di database maupun saat dikirim melalui jaringan.

**3.3 Modul laporan dan analitik**

Modul laporan dan analitik dalam sistem informasi akademik berfungsi untuk mengelola, menganalisis, dan menyajikan data akademik dalam bentuk laporan yang berguna untuk pengambilan keputusan. Berikut adalah penjelasan mengenai desain modul laporan dan analitik ini:

2. Komponen Utama Modul

- Pengelolaan Data Sumber: Mengumpulkan data dari berbagai modul lain dalam sistem informasi akademik, seperti data mahasiswa, dosen, mata kuliah, dan nilai.

- Generator Laporan: Fitur yang memungkinkan pengguna untuk membuat laporan berdasarkan kriteria tertentu. Laporan yang dapat dihasilkan antara lain:

- Laporan pendaftaran mahasiswa per semester.

- Laporan kehadiran kelas.

- Laporan hasil studi (KHS) mahasiswa.

- Statistik kinerja mata kuliah.

- Dashboard Analitik: Antarmuka visual yang menyajikan data analitik dalam bentuk grafik dan tabel. Dashboard ini dapat menampilkan:

- Tren pendaftaran mahasiswa dari waktu ke waktu.

- Rata-rata nilai per mata kuliah atau program studi.

- Fitur Filter dan Pencarian: Memungkinkan pengguna untuk memfilter data berdasarkan parameter tertentu (misalnya, tahun ajaran, program studi, atau jenis kelamin) untuk analisis yang lebih terfokus.

3. Arsitektur Sistem

- Client-Server Architecture: Menggunakan arsitektur klien-server untuk memisahkan antarmuka pengguna dari logika bisnis dan penyimpanan data.

- Database: Menyimpan semua data yang diperlukan untuk laporan dan analitik dalam basis data yang terstruktur dengan baik.

**3.4 Modul integrasi API**

Desain modul integrasi API untuk registrasi mata kuliah (KRS) bertujuan untuk memfasilitasi pertukaran data antara sistem KRS dan aplikasi atau sistem lain yang relevan. Berikut adalah penjelasan mengenai desain modul ini.

1. Komponen Utama Modul Integrasi API

- API Gateway: Poin masuk utama untuk semua permintaan API. Ini mengelola dan mengarahkan permintaan ke layanan yang sesuai serta menerapkan kebijakan keamanan.

- Web Service RESTful: Menggunakan arsitektur RESTful untuk memfasilitasi komunikasi antara sistem. Web service ini dapat menangani berbagai jenis permintaan (GET, POST, PUT, DELETE) untuk mengelola data KRS.

- Pengelola Autentikasi: Memastikan bahwa hanya pengguna terotorisasi yang dapat mengakses API, menggunakan metode seperti token JWT (JSON Web Token) atau OAuth.

- Data Mapper: Bertanggung jawab untuk memetakan data antara format yang berbeda (misalnya, dari JSON ke format database) agar dapat diproses dengan benar oleh sistem tujuan.

- Pengelola Kesalahan: Menangani kesalahan yang mungkin terjadi selama proses integrasi, termasuk validasi input dan penanganan kesalahan jaringan.

2. Arsitektur Sistem

- Client-Server Architecture: Menggunakan arsitektur klien-server di mana klien (misalnya, aplikasi mobile atau web) mengirimkan permintaan ke server API.

- Database Terintegrasi: Menyimpan semua data yang diperlukan untuk proses integrasi dalam basis data yang terstruktur dengan baik.

3. Proses Integrasi Data

1. Pengiriman Permintaan: Klien mengirimkan permintaan ke API gateway menggunakan metode HTTP yang sesuai.

2. Autentikasi dan Otorisasi: API gateway memverifikasi kredensial pengguna sebelum meneruskan permintaan ke layanan backend.

3. Pengolahan Permintaan: Permintaan diteruskan ke web service yang sesuai untuk diproses.

4. Pengembalian Respon: Setelah pemrosesan selesai, data dikembalikan ke klien dalam format yang telah ditentukan (misalnya, JSON).

**Keamanan dan Manajemen API**

- Rate Limiting: Mengatur jumlah permintaan yang dapat dilakukan pengguna dalam periode waktu tertentu untuk mencegah penyalahgunaan.

- Logging dan Monitoring: Mencatat semua aktivitas API untuk analisis lebih lanjut dan pemecahan masalah jika terjadi kesalahan.

**4. DESAIN BASIS DATA**

**4.1 Model Data**

Desain Basis Data untuk Registrasi Mata Kuliah (KRS)

2. Entitas Utama dan Atribut

Berikut adalah entitas utama yang perlu ada dalam basis data KRS beserta atributnya:

|  |  |
| --- | --- |
| Entitas | Atribut |
| Mahasiswa | ID\_Mahasiswa(PK),Nama,NIM,Email,Jurusan,Tanggal\_Lahir |
| Mata kuliah | Kode\_Mata\_Kuliah(PK),Nama,Mata,Kuliah,SKS,Deskripsi,Prasyarat |
| Jadwal Ruangan | ID\_Jadwal(PK),Kode\_Mata\_Kuliah(FK),Hari,Jam\_Mulai,Jam\_Selesai, |
| Pendaftaran KRS semester | ID\_Pendaftaran(PK),ID\_Mahasiswa (FK),Kode\_Mata\_Kuliah(FK), |
| Dosen | ID\_Dosen(PK),Nama,Dosen,Email \_Dosen |
| Kelas | ID\_Kelas (PK),Kode\_Mata\_Kuliah(FK),ID\_Dosen(FK),Ruangan |
|  |  |

**4.2 diagram entity relationship (ERD)**

Berikut adalah desain ERD untuk sistem registrasi mata kuliah (KRS). Diagram ini mencakup entitas utama, atribut, dan hubungan antar entitas.

Entitas dan Atribut

1. Mahasiswa

- ID\_Mahasiswa (Primary Key)

- Nama

- NIM

- Email

- Jurusan

- Tanggal\_Lahir

2. Mata Kuliah

- Kode\_Mata\_Kuliah (Primary Key)

- Nama\_Mata\_Kuliah

- SKS

- Deskripsi

- Prasyarat

3. Jadwal

- ID\_Jadwal (Primary Key)

- Kode\_Mata\_Kuliah (Foreign Key)

- Hari

- Jam\_Mulai

- Jam\_Selesai

- Ruangan

4. Pendaftaran KRS

- ID\_Pendaftaran (Primary Key)

- ID\_Mahasiswa (Foreign Key)

- Kode\_Mata\_Kuliah (Foreign Key)

- Semester

5. Dosen

- ID\_Dosen (Primary Key)

- Nama\_Dosen

- Email\_Dosen

6. Kelas

- ID\_Kelas (Primary Key)

- Kode\_Mata\_Kuliah (Foreign Key)

- ID\_Dosen (Foreign Key)

- Ruangan

Relasi Antar Entitas

-Mahasiswa ↔ Pendaftaran KRS: Relasi one-to-many karena satu mahasiswa dapat mendaftar ke banyak mata kuliah.

- Mata Kuliah ↔ Pendaftaran KRS: Relasi one-to-many karena satu mata kuliah dapat diambil oleh banyak mahasiswa.

- Mata Kuliah ↔ Jadwal: Relasi one-to-many karena satu mata kuliah dapat memiliki beberapa jadwal.

- Dosen ↔ Kelas: Relasi one-to-many karena satu dosen dapat mengajar beberapa kelas.

Simbol dalam ERD

- Entitas: Direpresentasikan dengan bentuk persegi panjang.

- Atribut: Direpresentasikan dengan bentuk oval.

- Relasi: Direpresentasikan dengan bentuk belah ketupat (diamond).

- Garis Penghubung: Menunjukkan hubungan antara entitas dan atribut, serta relasi antar entitas.

Contoh Diagram ERD Konseptual:

Mahasiswa -------------------- Mata\_Kuliah

| |

| |

v v

Atribut Jadwal -------------------- Dosen

Penjelasan Diagram:

1. Mahasiswa memiliki atribut seperti NIM, nama, email, dll.

2. Mahasiswa mendaftar ke mata kuliah melalui tabel Pendaftaran KRS yang menyimpan hubungan antara mahasiswa dan mata kuliah.

3. Mata kuliah memiliki jadwal yang dikelola dalam tabel Jadwal.

4. Dosen mengajar kelas tertentu yang terkait dengan mata kuliah.

**4.3 skema indexing dan optimasi**

1. Skema Indexing

Indexing adalah teknik yang digunakan untuk mempercepat proses pencarian data dalam basis data. Berikut adalah beberapa jenis indeks yang dapat diterapkan dalam sistem KRS:

- Indeks Primer: Digunakan pada kolom yang menjadi primary key, seperti `ID\_Mahasiswa`, `Kode\_Mata\_Kuliah`, dan `ID\_Pendaftaran`. Indeks ini memastikan bahwa setiap nilai dalam kolom tersebut unik dan mempercepat pencarian berdasarkan kunci utama.

- Indeks Sekunder: Dapat diterapkan pada kolom yang sering digunakan dalam kondisi pencarian (WHERE clause), seperti `NIM`, `Nama\_Mahasiswa`, dan `Kode\_Mata\_Kuliah`. Indeks ini membantu mempercepat query yang melibatkan pencarian berdasarkan kolom-kolom tersebut.

- Indeks Gabungan: Untuk query yang sering menggunakan beberapa kolom dalam kondisi pencarian, seperti kombinasi `NIM` dan `Semester`, indeks gabungan dapat dibuat untuk meningkatkan performa.

- Indeks Full-Text: Jika sistem memungkinkan pencarian berdasarkan kata kunci dalam deskripsi mata kuliah atau nama mahasiswa, indeks full-text dapat digunakan untuk meningkatkan kecepatan pencarian teks.

2. Teknik Optimasi

- Normalisasi Basis Data: Melakukan normalisasi untuk mengurangi redundansi data dan memastikan integritas data. Dengan mengelompokkan data ke dalam tabel-tabel yang relevan, sistem akan lebih efisien dalam penyimpanan dan pengambilan data.

- Penggunaan Query yang Efisien: Mengoptimalkan query SQL dengan menggunakan JOIN yang tepat, menghindari penggunaan subquery yang tidak perlu, serta memanfaatkan fungsi agregat dengan bijak.

- Caching Hasil Query: Mengimplementasikan caching untuk menyimpan hasil query yang sering diakses. Ini mengurangi beban pada database dengan menghindari eksekusi query yang sama berulang kali.

- Partitioning Tabel: Jika tabel KRS menjadi sangat besar, partitioning dapat digunakan untuk membagi tabel menjadi bagian-bagian lebih kecil berdasarkan kriteria tertentu (misalnya, semester atau tahun ajaran) untuk meningkatkan performa akses data.

- Monitoring dan Tuning Kinerja: Secara rutin memonitor kinerja database dan melakukan tuning pada indeks, query, serta konfigurasi server database agar tetap optimal seiring dengan pertumbuhan data.

3. Implementasi Indeks

Berikut adalah contoh implementasi indeks pada tabel KRS menggunakan SQL:

-- Membuat indeks primer pada tabel Mahasiswa

CREATE TABLE Mahasiswa (

ID\_Mahasiswa INT PRIMARY KEY,

NIM VARCHAR(20) UNIQUE,

Nama VARCHAR(100),

Email VARCHAR(100),

Jurusan VARCHAR(50),

Tanggal\_Lahir DATE

);

-- Membuat indeks sekunder pada kolom Nama

CREATE INDEX idx\_nama\_mahasiswa ON Mahasiswa(Nama);

-- Membuat tabel KRS dengan indeks

CREATE TABLE KRS (

ID\_Pendaftaran INT PRIMARY KEY,

ID\_Mahasiswa INT,

Kode\_Mata\_Kuliah VARCHAR(10),

Semester INT,

FOREIGN KEY (ID\_Mahasiswa) REFERENCES Mahasiswa(ID\_Mahasiswa)

);

-- Membuat indeks gabungan pada tabel KRS

CREATE INDEX idx\_krs\_mahasiswa\_semester ON KRS(ID\_Mahasiswa, Semester);

**5. ANTARMUKA PENGGUNA**

**5.1 Wireframe desain**

1. Halaman Login

- Deskripsi: Halaman pertama yang muncul saat pengguna membuka aplikasi. Pengguna, baik mahasiswa maupun dosen, harus memasukkan NIM dan password untuk mengakses sistem.

- Fitur:

- Kolom input untuk NIM dan password.

- Tombol "Login" untuk mengirimkan data.

- Opsi "Lupa Password" untuk pemulihan akun.

2. Halaman Ringkasan (Summary)

- Deskripsi: Setelah login, pengguna diarahkan ke halaman ringkasan yang menampilkan informasi penting.

- Fitur:

- Pesan selamat datang dengan nama mahasiswa.

- Tampilkan semester yang akan ditempuh.

- Rekapitulasi hasil studi dari semester sebelumnya, termasuk jumlah SKS yang telah diselesaikan dan mata kuliah yang direkomendasikan.

3. Halaman Pilih Mata Kuliah

- Deskripsi: Halaman ini menampilkan daftar mata kuliah yang dapat dipilih sesuai dengan semester yang akan ditempuh.

- Fitur:

- Daftar mata kuliah yang relevan, dengan penandaan mata kuliah yang sudah diambil atau lulus.

- Fitur pencarian untuk memudahkan menemukan mata kuliah tertentu.

- Checkbox atau tombol untuk memilih mata kuliah.

4. Halaman Persetujuan (Approval Request)

- Deskripsi: Menampilkan daftar mata kuliah yang telah dipilih oleh mahasiswa untuk diverifikasi oleh dosen pembimbing akademik.

- Fitur:

- Daftar mata kuliah yang dipilih dengan opsi untuk mengedit atau membatalkan pilihan.

- Tombol "Kirim Permintaan Persetujuan" untuk mengirimkan daftar ke dosen.

5. Halaman Verifikasi Dosen

- Deskripsi: Halaman khusus untuk dosen pembimbing akademik untuk memverifikasi pilihan mata kuliah mahasiswa.

- Fitur:

- Daftar mata kuliah yang diajukan oleh mahasiswa dengan status verifikasi.

- Opsi untuk menyetujui atau menolak pilihan mata kuliah.

- Catatan atau komentar dari dosen jika diperlukan.

6. Halaman Hasil KRS (BRS)

- Deskripsi: Menampilkan hasil akhir dari pendaftaran KRS setelah diverifikasi oleh dosen.

- Fitur:

- Daftar mata kuliah yang telah disetujui beserta jadwalnya.

- Opsi untuk mencetak atau menyimpan KRS dalam format PDF.

Pertimbangan Desain Antarmuka

- User Experience (UX): Antarmuka harus intuitif dan mudah digunakan, dengan navigasi yang jelas agar pengguna dapat dengan cepat menemukan informasi dan fungsi yang dibutuhkan.

- Responsivitas: Antarmuka harus dapat diakses melalui berbagai perangkat, termasuk desktop dan mobile, agar pengguna dapat melakukan registrasi kapan saja dan di mana saja.

- Aksesibilitas: Memastikan bahwa antarmuka dapat diakses oleh semua pengguna, termasuk mereka yang memiliki kebutuhan khusus.

**5.1 Wireframe design**

Wireframe Design untuk Halaman Login, Dashboard, dan Halaman CRUD

Berikut adalah desain wireframe untuk tiga komponen utama dalam sistem registrasi mata kuliah (KRS): halaman login, dashboard, dan halaman CRUD (Create, Read, Update, Delete).

1. Halaman Login

Deskripsi: Halaman login adalah titik akses pertama bagi pengguna untuk masuk ke dalam sistem.

Elemen Utama:

- Logo Aplikasi: Menampilkan logo di bagian atas.

- Kolom Input:

- Username/NIM

- Password

- Tombol:

- Login

- Lupa Password (link)

- Pesan Kesalahan: Tempat untuk menampilkan pesan jika login gagal.

Wireframe:

+-----------------------------------+

| [Logo] |

| |

| Username: [\_\_\_\_\_\_\_\_\_\_\_\_\_\_] |

| |

| Password: [\_\_\_\_\_\_\_\_\_\_\_\_\_\_] |

| |

| [Login] |

| |

| [Lupa Password?] |

| |

+-----------------------------------+

```

2. Dashboard

Elemen Utama:

- Menu Navigasi: Tautan ke berbagai bagian sistem (KRS, Jadwal, Profil, Logout).

- Ringkasan KRS: Menampilkan status KRS mahasiswa.

- Notifikasi: Pemberitahuan tentang kegiatan atau pengumuman terbaru.

- Statistik Akademik: Grafik atau tabel yang menunjukkan kinerja akademik.

Wireframe:

+--------------------------------------------------+

| [Logo] Dashboard |

|--------------------------------------------------|

| [KRS] [Jadwal] [Profil] [Logout] |

|--------------------------------------------------|

| Ringkasan KRS |

| - Mata Kuliah Terdaftar: 4 |

| - SKS Total: 12 |

|--------------------------------------------------|

| Notifikasi |

| - Pengumuman: KRS dibuka hingga 20 April |

|--------------------------------------------------|

| Statistik Akademik |

| [Grafik Kinerja] |

+--------------------------------------------------+

```

3. Halaman CRUD

Deskripsi: Halaman ini memungkinkan pengguna untuk melakukan operasi Create, Read, Update, dan Delete pada data mata kuliah.

Elemen Utama:

- Form Input Mata Kuliah:

- Kode Mata Kuliah

- Nama Mata Kuliah

- SKS

- Deskripsi

- Tombol Aksi:

- Simpan (untuk Create/Update)

- Hapus (untuk Delete)

- Daftar Mata Kuliah: Tabel yang menampilkan semua mata kuliah dengan opsi Edit dan Hapus di setiap baris.

Wireframe:

+-----------------------------------------------+

| Manajemen Mata Kuliah |

|-----------------------------------------------|

| Kode: [\_\_\_\_\_\_\_\_\_\_\_\_\_\_] |

| Nama: [\_\_\_\_\_\_\_\_\_\_\_\_\_\_] |

| SKS: [\_\_\_\_\_] |

| Deskripsi: [\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_]|

| |

| [Simpan] [Hapus] |

|-----------------------------------------------|

| Daftar Mata Kuliah |

|------------------------------------------------|

| Kode | Nama | SKS | Aksi |

|----------|------------------|-----|-----------|

| MK001 | Pemrograman Dasar | 3 | [Edit] [Hapus]|

+----------+------------------+-----+-----------+

**5.2 Desain responsive**

Desain responsif adalah pendekatan dalam pengembangan antarmuka pengguna (UI) yang memastikan bahwa tampilan dan interaksi situs web atau aplikasi dapat menyesuaikan diri dengan berbagai ukuran layar dan perangkat. Berikut adalah beberapa prinsip dan praktik terbaik dalam desain responsif yang dapat diterapkan:

Prinsip Desain Responsif

1. Tata Letak Fleksibel

Gunakan tata letak berbasis grid yang dapat menyesuaikan lebar dan tinggi elemen sesuai dengan ukuran layar. Ini memungkinkan konten untuk ditampilkan dengan baik di berbagai perangkat, dari smartphone hingga desktop.

2. Media Queries

Terapkan media queries dalam CSS untuk mengubah gaya elemen berdasarkan karakteristik perangkat, seperti lebar layar dan orientasi. Ini memungkinkan elemen untuk beradaptasi secara dinamis.

3. Ukuran Elemen yang Sesuai

Pastikan tombol, tautan, dan elemen UI lainnya cukup besar dan memiliki spasi yang memadai untuk memudahkan interaksi pada perangkat sentuh. Ukuran tombol minimal yang disarankan adalah 44x44 piksel.

4. Navigasi Intuitif

Ciptakan navigasi yang jelas dan mudah dipahami, dengan menu yang dapat diakses dan ikon yang mudah dikenali. Menu harus tetap konsisten di semua perangkat.

5. Konten Responsif

Konten seperti teks, gambar, dan video harus dapat menyesuaikan ukuran dan tata letaknya sesuai dengan ukuran layar perangkat. Gunakan gambar dengan resolusi yang tepat untuk mengurangi waktu muat.

6. Umpan Balik Pengguna

Berikan umpan balik yang jelas kepada pengguna saat mereka berinteraksi dengan elemen antarmuka, memastikan bahwa mereka mengetahui tindakan yang telah dilakukan.

Praktik Terbaik dalam Desain Responsif

- Mentalitas Mobile-First: Mulailah desain dengan mempertimbangkan pengalaman pengguna di perangkat mobile terlebih dahulu, kemudian sesuaikan untuk desktop.

- Breakpoint: Tentukan tiga atau lebih breakpoint dalam desain untuk menangani berbagai ukuran layar secara efektif.

- Scalable Vector Graphics (SVG): Gunakan format SVG untuk grafik 2D agar tetap tajam pada berbagai resolusi layar.

- Aksesibilitas Font: Pastikan font dapat dibaca dengan baik di semua perangkat, menggunakan kontras warna yang tepat dan ukuran font yang sesuai.

**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**

Metode Pengujian Unit

Pengujian unit dapat dilakukan menggunakan berbagai teknik, seperti:

- Black Box Testing: Menguji antarmuka pengguna dan memastikan input serta output sesuai dengan spesifikasi.

- White Box Testing: Menguji logika internal dari fungsi atau metode untuk memastikan bahwa semua jalur eksekusi berfungsi dengan baik.

- Gray Box Testing: Menggabungkan kedua pendekatan di atas untuk menganalisis kerusakan dengan pemahaman tentang struktur internal aplikasi.

Alat untuk Pengujian Unit

Beberapa alat populer yang digunakan untuk melakukan pengujian unit antara lain:

- JUnit: Untuk pengujian unit di Java.

- NUnit: Untuk .NET.

- PHPUnit: Untuk PHP.

- JMockit: Alat open-source untuk pengujian unit.

- EMMA: Untuk analisis dan pelaporan kode Java[6].

**Proses Pengujian Unit**

1. Menulis Kasus Uji: Pengembang menulis skrip pengujian yang mencakup berbagai skenario penggunaan untuk setiap unit kode.

2. Menjalankan Pengujian: Kasus uji dijalankan secara otomatis menggunakan alat pengujian unit.

3. Menganalisis Hasil: Hasil dari pengujian dianalisis untuk menentukan apakah unit berfungsi seperti yang diharapkan. Jika ada kegagalan, pengembang perlu memperbaiki kode dan menjalankan pengujian kembali.

**7.2 pengujian keamanan**

1. Security Scanning: Memindai sistem untuk menemukan kerentanan yang ada dalam aplikasi, jaringan, atau file system.

2. Vulnerability Scanning: Mengidentifikasi kelemahan dalam sistem dengan menggunakan alat pemindaian untuk menemukan titik lemah yang mungkin dieksploitasi.

3. Penetration Testing: Simulasi serangan siber yang dilakukan oleh profesional keamanan (pentester) untuk menguji kekuatan pertahanan sistem. Ini melibatkan eksploitasi kerentanan yang ditemukan untuk menilai seberapa jauh seorang penyerang dapat masuk ke dalam sistem.

4. Risk Assessment: Mengidentifikasi dan menganalisis risiko yang mungkin dihadapi oleh sistem informasi. Ini membantu dalam menentukan langkah-langkah mitigasi yang tepat.

5. Security Audit: Proses evaluasi menyeluruh terhadap kebijakan dan prosedur keamanan organisasi untuk memastikan bahwa mereka memadai dan diterapkan dengan benar.

Alat untuk Pengujian Keamanan

Beberapa alat populer yang digunakan dalam pengujian keamanan meliputi:

- OWASP ZAP: Alat open-source untuk mendeteksi kerentanan dalam aplikasi web.

- Burp Suite: Alat komersial yang menyediakan fitur lengkap untuk identifikasi kerentanan. Nessus: Alat pemindaian kerentanan yang banyak digunakan di industri.

**7.2 uji fungsionalitas**

1. Identifikasi Persyaratan: Menentukan semua fungsi yang harus diuji berdasarkan dokumen spesifikasi.

2. Desain Test Case: Merancang skenario pengujian yang mencakup berbagai kondisi input dan hasil yang diharapkan.

3. Eksekusi Pengujian: Menjalankan skenario pengujian untuk memverifikasi apakah fungsionalitas bekerja dengan benar.

4. Evaluasi Hasil: Membandingkan hasil yang diperoleh dengan hasil yang diharapkan. Jika ada kesalahan, bug akan dilaporkan dan diperbaiki.

Jenis-Jenis Uji Fungsionalitas

- Unit Testing: Menguji bagian terkecil dari aplikasi, seperti fungsi atau modul individual.

- Integration Testing: Memastikan bahwa modul atau komponen yang berbeda dapat bekerja sama dengan baik.

- System Testing: Menguji keseluruhan aplikasi sebagai satu sistem terintegrasi untuk memastikan bahwa semua fungsi bekerja sesuai dengan harapan.

- User Acceptance Testing (UAT): Pengujian akhir untuk memastikan aplikasi siap digunakan oleh pengguna akhir.

**8. LAMPIRAN**

1. Diagram Kelas (Class Diagram)

Diagram kelas adalah representasi statis dari struktur sistem, menggambarkan kelas-kelas yang ada, atribut, metode, serta hubungan antar kelas. Berikut adalah contoh diagram kelas untuk sistem registrasi mata kuliah (KRS):

+-------------------+

| Mahasiswa |

+-------------------+

| - idMahasiswa |

| - nama |

| - nim |

| - email |

| - jurusan |

+-------------------+

| + daftarKRS() |

| + lihatKRS() |

+-------------------+

+-------------------+

| MataKuliah |

+-------------------+

| - kodeMataKuliah |

| - nama |

| - sks |

| - deskripsi |

+-------------------+

| + tambahMataKuliah()|

| + hapusMataKuliah()|

+-------------------+

+-------------------+

| KRS |

+-------------------+

| - idPendaftaran |

| - idMahasiswa |

| - kodeMataKuliah |

| - semester |

+-------------------+

| + simpanKRS() |

| + hapusKRS() |

+-------------------+

+-------------------+

| Dosen |

+-------------------+

| - idDosen |

| - nama |

| - email |

+-------------------+

| + verifikasiKRS() |

+-------------------+

2. Diagram Urutan Proses (Sequence Diagram)

Diagram urutan menggambarkan interaksi antara objek dalam sistem berdasarkan waktu. Berikut adalah contoh diagram urutan untuk proses pendaftaran mata kuliah oleh mahasiswa:

Mahasiswa KRS MataKuliah Dosen

| | | |

|----daftarKRS()---->| | |

| |----tambahMataKuliah()---->| |

| | || | |

|<--daftarMataKuliah--| | |

| | | |

```

Penjelasan Diagram

1. Diagram Kelas:

- Mahasiswa: Mewakili entitas mahasiswa dengan atribut seperti ID, nama, NIM, email, dan jurusan. Metode yang dimiliki termasuk daftarKRS untuk mendaftar mata kuliah dan lihatKRS untuk melihat daftar mata kuliah yang telah didaftarkan.

- MataKuliah: Mewakili entitas mata kuliah dengan atribut seperti kode, nama, SKS, dan deskripsi. Metode untuk menambah dan menghapus mata kuliah juga ada di sini.

- KRS: Mewakili pendaftaran mata kuliah dengan atribut ID pendaftaran, ID mahasiswa, kode mata kuliah, dan semester. Memiliki metode untuk menyimpan dan menghapus KRS.

- Dosen: Mewakili dosen yang memiliki metode untuk memverifikasi KRS mahasiswa.

2. Diagram Urutan Proses:

- Proses dimulai ketika mahasiswa memanggil metode daftarKRS untuk mendaftar mata kuliah.

- Sistem KRS kemudian memanggil metode tambahMataKuliah pada objek MataKuliah.

- Setelah penambahan berhasil, sistem mengirimkan konfirmasi kembali kepada mahasiswa.

- Mahasiswa juga dapat memanggil metode lihatKRS untuk melihat daftar mata kuliah yang telah didaftarkan.

2. Dokumentasi API untuk Integrasi Eksternal

- Deskripsi API: Penjelasan singkat tentang tujuan dan fungsi API.

- Kegunaan: Menjelaskan bagaimana API dapat digunakan dalam aplikasi dan manfaatnya bagi pengguna.

2. Autentikasi

- Metode Autentikasi: Menjelaskan mekanisme autentikasi yang digunakan (misalnya, token Bearer, OAuth 2.0).

- Contoh Permintaan Autentikasi:

http

POST /auth/login

Content-Type: application/json

{

"username": "user@example.com",

"password": "yourpassword"

}

```

3. Endpoint API

Berikan daftar endpoint yang tersedia beserta deskripsi fungsinya.

| Endpoint | Metode | Deskripsi |

|-----------------------------|------------|-----------------------------------------------|

| `/api/mata-kuliah` | GET | Mengambil daftar mata kuliah |

| `/api/mata-kuliah` | POST | Menambahkan mata kuliah baru |

| `/api/mata-kuliah/{id}` | GET | Mengambil detail mata kuliah berdasarkan ID |

| `/api/mata-kuliah/{id}` | PUT | Memperbarui informasi mata kuliah berdasarkan ID |

| `/api/mata-kuliah/{id}` | DELETE | Menghapus mata kuliah berdasarkan ID |

4. Parameter

Jelaskan parameter yang diperlukan untuk setiap endpoint.

- GET /api/mata-kuliah

- Query Parameters:

- `semester` (optional): Menentukan semester untuk filter mata kuliah.

- POST /api/mata-kuliah

- Body Parameters:

- `kode`: Kode mata kuliah (string)

- `nama`: Nama mata kuliah (string)

- `sks`: Jumlah SKS (integer)

- `deskripsi`: Deskripsi mata kuliah (string)

5. Contoh Permintaan dan Respons

Berikan contoh permintaan dan respons untuk setiap endpoint.

- Contoh Permintaan untuk Menambahkan Mata Kuliah:

```http

POST /api/mata-kuliah

Content-Type: application/json

{

"kode": "MK001",

"nama": "Pemrograman Dasar",

"sks": 3,

"deskripsi": "Mata kuliah dasar pemrograman."

}

```

- \*\*Contoh Respons Berhasil\*\*:

```json

{

"message": "Mata kuliah berhasil ditambahkan.",

"data": {

"id": 1,

"kode": "MK001",

"nama": "Pemrograman Dasar",

"sks": 3,

"deskripsi": "Mata kuliah dasar pemrograman."

}

}

```

6. Penanganan Kesalahan

Jelaskan bagaimana kesalahan ditangani dan contoh respons kesalahan.

- Contoh Respons Kesalahan:

```json

{

"error": {

"code": 400,

"message": "Kode mata kuliah sudah ada."

}

}

3. laporan hasil pengujian dan rekomendasi perbaikan

### Laporan Hasil Pengujian dan Rekomendasi Perbaikan

1. Pendahuluan

Laporan ini menyajikan hasil pengujian perangkat lunak yang telah dilakukan pada sistem registrasi mata kuliah (KRS). Pengujian ini bertujuan untuk mengevaluasi fungsionalitas, keamanan, dan kinerja sistem, serta memberikan rekomendasi perbaikan berdasarkan temuan yang diperoleh.

2. Metodologi Pengujian

Pengujian dilakukan dengan menggunakan metode Black Box Testing, yang berfokus pada pengujian fungsionalitas aplikasi tanpa mempertimbangkan struktur internal kode. Pengujian mencakup:

- Uji Fungsionalitas: Memastikan bahwa semua fitur berfungsi sesuai dengan spesifikasi.

- Uji Keamanan: Mengidentifikasi potensi kerentanan dalam sistem.

- Uji Kinerja: Mengukur responsivitas dan stabilitas sistem di bawah beban.

3. Hasil Pengujian

3.1 Uji Fungsionalitas

- Test Case: Pendaftaran Mata Kuliah

- Hasil yang Diharapkan: Mahasiswa dapat mendaftar mata kuliah dengan sukses.

- Hasil Aktual: Berhasil mendaftar, tetapi terdapat keterlambatan dalam proses penyimpanan data.

- Test Case: Login Mahasiswa

- Hasil yang Diharapkan: Mahasiswa dapat login dengan kredensial yang benar.

- Hasil Aktual: Login berhasil, tetapi terdapat kesalahan penanganan saat memasukkan password yang salah (tidak memberikan umpan balik yang jelas).

3.2 Uji Keamanan

- Vulnerability Scan: Ditemukan beberapa kerentanan, termasuk:

- Input tidak divalidasi dengan baik pada formulir pendaftaran, memungkinkan serangan SQL Injection.

- Tidak ada mekanisme untuk membatasi percobaan login, meningkatkan risiko serangan brute force.

3.3 Uji Kinerja

- Load Testing: Sistem mengalami penurunan kinerja saat lebih dari 100 pengguna mengakses secara bersamaan, dengan waktu respons meningkat menjadi lebih dari 5 detik.

4. Rekomendasi Perbaikan

1. Optimasi Proses Penyimpanan Data:

- Tinjau dan optimalkan algoritma penyimpanan data untuk mengurangi keterlambatan saat pendaftaran mata kuliah.

2. Perbaikan Penanganan Kesalahan Login:

- Tambahkan umpan balik yang lebih informatif ketika pengguna memasukkan kredensial yang salah untuk meningkatkan pengalaman pengguna.

3. Peningkatan Keamanan:

- Implementasikan validasi input yang ketat pada semua formulir untuk mencegah serangan SQL Injection.

- Tambahkan fitur pembatasan percobaan login untuk mencegah serangan brute force, seperti penguncian akun setelah beberapa percobaan gagal.

4. Peningkatan Kinerja:

- Lakukan pengoptimalan pada database dan server untuk meningkatkan responsivitas di bawah beban tinggi.

- Pertimbangkan penggunaan caching untuk mengurangi waktu muat halaman.