**Dokumen Desain Perangkat Lunak (Software Design Document - SDD)**

**1. Pendahuluan**

**1.1 Tujuan**

Dokumen ini menjelaskan desain teknis dan arsitektur modul Penilaian KHS pada Sistem Informasi Akademik. Tujuannya untuk memberikan panduan implementasi bagi pengembang dan referensi teknis bagi pihak terkait.

**1.2 Ruang Lingkup**

Modul ini memungkinkan dosen memasukkan nilai akhir mahasiswa, serta menghasilkan dan menampilkan KHS kepada mahasiswa. Modul terintegrasi dengan sistem registrasi, manajemen mata kuliah, dan autentikasi pengguna.

**1.3 Definisi Istilah**

* **KHS**: Kartu Hasil Studi
* **IP**: Indeks Prestasi
* **IPK**: Indeks Prestasi Kumulatif
* **SIAKAD**: Sistem Informasi Akademik

**1.4 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**

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

 **Backend**: PHP Yii2 Framework

 **Frontend**: HTML, CSS, Bootstrap 5

 **Database**: MySQL

 **API**: RESTful API

 **Caching**: Redis

**2.2 Komponen Utama Sistem**

1. **Model**:

* Representasi data nilai, mahasiswa, dan mata kuliah.
* Berfungsi untuk manipulasi dan pengambilan data dari database menggunakan ORM Yii2.

1. **View**:

* Antarmuka pengguna untuk input nilai (dosen), serta tampilan KHS (mahasiswa).
* Menggunakan Bootstrap 5 agar responsif di berbagai perangkat.

1. **Controller**:

* Mengelola alur data antara View dan Model.
* Menangani logika perhitungan IP dan IPK.

1. **Database**:

* MySQL menyimpan data nilai, mata kuliah, dan riwayat akademik mahasiswa.
* Indexing digunakan untuk mempercepat query nilai dan IPK.

1. **API Layer**:

* RESTful API untuk akses nilai oleh sistem eksternal, misalnya sistem pelaporan kampus.
* Mendukung format JSON dengan autentikasi API key.

1. **Caching Layer**:

* Redis digunakan untuk menyimpan hasil hitung IPK dan data KHS yang sering diakses untuk meningkatkan performa.

**3. Desain Modul dan Komponen**

**3.1 Modul Autentikasi Pengguna**

 **Fungsi:**  
Memastikan hanya pengguna terautentikasi (mahasiswa, dosen, admin) yang dapat mengakses atau memproses data KHS, seperti input nilai (dosen) dan melihat KHS (mahasiswa).

 **Input:**

* Username/email
* Password
* Token autentikasi (JWT)
* CAPTCHA (saat login)

 **Output:**

* Token sesi pengguna
* Status autentikasi (sukses/gagal)
* Informasi peran pengguna (dosen, mahasiswa)

 **Teknologi:**

* **OAuth2** untuk otorisasi
* **JWT** (JSON Web Token) untuk manajemen sesi
* **CAPTCHA** sebagai perlindungan brute force

 **Validasi:**

* Email/username dan password diverifikasi
* CAPTCHA wajib jika ada upaya login gagal berturut-turut
* Token JWT divalidasi untuk akses ke fitur KHS

**3.2 Modul Manajemen Data**

 **Fungsi:**  
CRUD (Create, Read, Update, Delete) data nilai mahasiswa, mata kuliah, dan riwayat KHS per semester.

 **Input:**

* ID Mahasiswa
* ID Mata Kuliah
* Nilai angka (0–100)
* Semester, tahun ajaran

 **Output:**

* Data nilai tersimpan dalam database
* Perhitungan otomatis nilai huruf, bobot, IP, IPK
* Respons JSON melalui API (misal: untuk tampilan KHS)

 **Teknologi:**

* **ORM Yii2**: Untuk pengelolaan data nilai, mahasiswa, dan mata kuliah
* **Indexing MySQL**: Diterapkan pada kolom id\_mahasiswa, semester, dan id\_matkul untuk mempercepat pencarian dan filtering data KHS

**3.3 Modul Laporan dan Analitik**

 **Fungsi:**  
Menyusun laporan KHS mahasiswa per semester serta menyajikan analitik IP/IPK dalam bentuk grafik dan dokumen.

 **Output:**

* **File PDF** KHS per mahasiswa
* **File Excel** rekap nilai per kelas/semester
* **Grafik** statistik: distribusi nilai, rata-rata IP per angkatan, IPK kumulatif

 **Teknologi:**

* **mPDF**: Untuk generate KHS PDF resmi
* **PhpSpreadsheet**: Untuk ekspor data nilai ke Excel
* **Chart.js**: Untuk visualisasi grafik IP/IPK di dashboard mahasiswa dan dosen

**3.4 Modul Integrasi API**

 **Fungsi:**  
Modul ini memungkinkan sistem untuk berkomunikasi dengan sistem eksternal seperti sistem pembayaran untuk biaya KHS, layanan pengecekan status pembayaran, atau sistem informasi akademik lain yang terkait.

 **Input:**

* Data permintaan dalam format **JSON** (misalnya, permintaan untuk verifikasi pembayaran atau integrasi data KHS dengan sistem lain)

 **Output:**

* **Respons API** dari layanan eksternal dalam format **JSON**, seperti status pembayaran atau informasi terkait transaksi

 **Teknologi:**

* **RESTful API**: Untuk komunikasi antar sistem
* **API Key**: Autentikasi untuk memastikan bahwa hanya sistem yang berwenang yang dapat mengakses API
* **JSON**: Format data standar untuk pertukaran informasi antara sistem

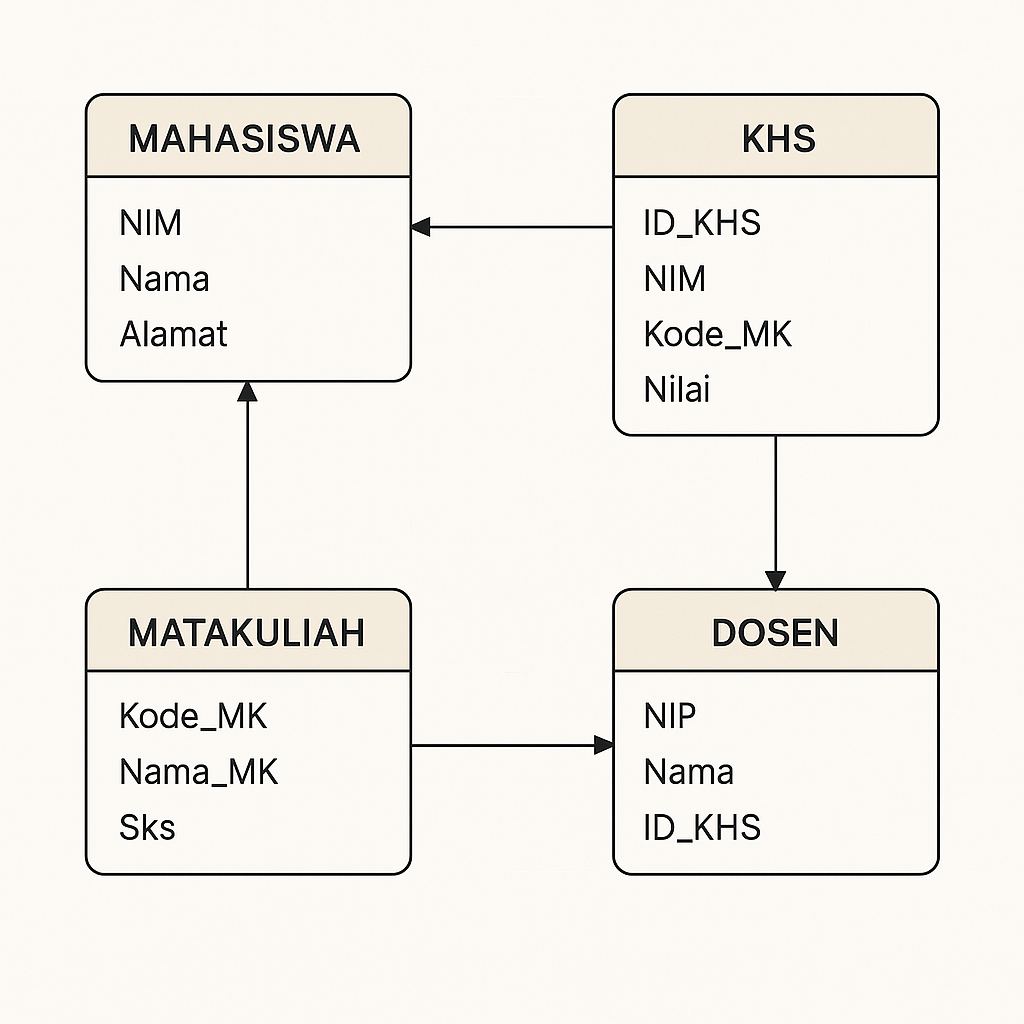
**4. Desain Basis Data**

**4.1 Model Data**

Struktur utama basis data mencakup tabel:

1. **users**
   1. id: ID pengguna (integer, primary key)
   2. nama: Nama pengguna (varchar)
   3. email: Email pengguna (varchar, unique)
   4. password: Password terenkripsi (varchar)
   5. role: Peran pengguna (enum: 'mahasiswa', 'dosen', 'admin')
   6. created\_at: Timestamp saat pengguna dibuat (datetime)
   7. updated\_at: Timestamp saat data terakhir diperbarui (datetime)
2. **transactions**
   1. id: ID transaksi (integer, primary key)
   2. user\_id: ID pengguna yang melakukan transaksi (foreign key ke users.id)
   3. amount: Jumlah biaya (decimal)
   4. date: Tanggal transaksi (date)
   5. status: Status transaksi (enum: 'pending', 'completed', 'failed')
3. **logs**
   1. id: ID log (integer, primary key)
   2. user\_id: ID pengguna yang melakukan aktivitas (foreign key ke users.id)
   3. activity: Deskripsi aktivitas (varchar)
   4. timestamp: Waktu aktivitas terjadi (datetime)
   5. ip\_address: Alamat IP pengguna yang melakukan aktivitas (varchar)
4. **api\_keys**
   1. id: ID API key (integer, primary key)
   2. user\_id: ID pengguna yang memiliki API key (foreign key ke users.id)
   3. key: API key yang unik (varchar)
   4. created\_at: Timestamp saat key dibuat (datetime)
   5. expired\_at: Timestamp saat key kadaluarsa (datetime)

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

****

**4.3 Skema Indexing dan Optimasi**

* **Indexing**:
  + Index pada kolom email di tabel users untuk mempercepat pencarian pengguna berdasarkan email.
  + Index pada kolom user\_id di tabel transactions, logs, dan api\_keys untuk mempercepat pencarian data terkait pengguna.
  + Index pada kolom date di tabel transactions untuk mempercepat pencarian berdasarkan tanggal transaksi.
* **Partisi Tabel Transaksi**:
  + Partisi tabel transactions berdasarkan periode (misal, per tahun atau per semester) untuk optimasi query yang memerlukan pencarian berdasarkan rentang waktu.
* **Backup Otomatis**:
  + **Backup database** dilakukan setiap 24 jam untuk memastikan data yang tersimpan dapat dipulihkan jika terjadi kegagalan sistem.

**5. Antarmuka Pengguna**

**5.1 Wireframe Desain**

 **Halaman Login**:

* **Form input**: Username dan password dengan opsi autentikasi dua faktor (misal: menggunakan OTP atau aplikasi autentikator seperti Google Authenticator).
* **Fitur tambahan**: Opsi pemulihan kata sandi, tombol "Login" dan "Daftar" bagi pengguna baru.

 **Dashboard**:

* **Ringkasan Data**: Menampilkan informasi penting seperti IPK terkini, daftar mata kuliah yang diambil, dan status KHS.
* **Laporan Interaktif**: Fitur filter berdasarkan semester, program studi, atau status pembayaran.
* **Grafik Statistik**: Visualisasi IP/IPK per semester atau tahun ajaran.

 **Halaman CRUD (Create, Read, Update, Delete)**:

* **Form Input Data**: Form untuk input nilai mahasiswa, edit data nilai, dan update status KHS.
* **Tabel Data**: Menampilkan data nilai dengan fitur sorting berdasarkan mata kuliah, mahasiswa, atau nilai.
* **Pagination**: Membagi data dalam beberapa halaman agar tidak memuat terlalu banyak data sekaligus.

**5.2 Desain Responsif**

 **Bootstrap 5**:

* Digunakan untuk memastikan tampilan antarmuka dapat menyesuaikan dengan baik di berbagai perangkat, baik desktop, tablet, maupun smartphone.
* **Grid system**: Menggunakan layout fleksibel yang responsif, sehingga halaman tetap terlihat baik meskipun diakses dengan perangkat dengan ukuran layar yang berbeda.

 **AJAX**:

* Menggunakan **AJAX** untuk mempercepat interaksi pengguna tanpa perlu reload halaman, seperti saat mengubah nilai mahasiswa atau memperbarui status KHS secara langsung.

**6. Pertimbangan Keamanan**

* **Enkripsi Data**:
  + **AES-256** digunakan untuk mengenkripsi data sensitif, seperti password pengguna dan informasi pribadi lainnya.
* **Kontrol Akses**:
  + **Role-Based Access Control (RBAC)** diterapkan untuk membatasi hak akses pengguna. Hanya **dosen** yang dapat menginput nilai, dan hanya **mahasiswa** yang dapat melihat KHS mereka sendiri.
* **Proteksi API**:
  + **API Key** dan **Token Autentikasi** digunakan untuk mengamankan API dan mencegah akses tidak sah.
* **Audit Log**:
  + Menyimpan semua aktivitas pengguna (login, pengeditan nilai, permintaan laporan) untuk memastikan kepatuhan dan memungkinkan pemantauan aktivitas mencurigakan.
* **Firewall Aplikasi Web (WAF)**:
  + Menggunakan WAF untuk melindungi aplikasi dari serangan umum seperti **SQL Injection**, **Cross-Site Scripting (XSS)**, dan serangan lainnya.

**7. Pengujian dan Validasi**

#### **7.1 Pengujian Unit**

* Pengujian setiap modul menggunakan **PHPUnit** untuk memastikan bahwa setiap fungsi bekerja dengan benar.
* **Simulasi skenario** input valid dan tidak valid, misalnya: pengujian input nilai yang tidak sesuai format atau mencoba login dengan kredensial yang salah.

#### **7.2 Pengujian Keamanan**

* **Pengujian penetrasi** untuk menemukan potensi celah keamanan dalam sistem, seperti SQL Injection atau XSS.
* **Uji beban** untuk memastikan sistem dapat menangani lonjakan pengguna, terutama saat akses untuk melihat atau mengunduh KHS dilakukan secara bersamaan.

#### **7.3 Uji Fungsionalitas**

* **Pengujian fitur utama** sistem, seperti login, akses data KHS, input nilai, dan pembuatan laporan.
* Penggunaan **Selenium** untuk mengotomatiskan pengujian antarmuka pengguna (UI) dan memastikan tampilan dan fungsi berjalan sebagaimana mestinya.

**8. Lampiran**

* **Diagram Kelas**: Menunjukkan struktur kelas dan hubungan antar komponen dalam sistem (misalnya, kelas untuk Pengguna, Transaksi, Nilai, dll.).
* **Diagram Urutan Proses**: Menyediakan gambaran alur proses interaksi pengguna dengan sistem, seperti alur login, input nilai, atau pembuatan laporan KHS.
* **Dokumentasi API**: Berisi detail tentang endpoint API yang digunakan untuk integrasi dengan sistem eksternal atau untuk mengambil data laporan KHS.
* **Laporan Hasil Pengujian dan Rekomendasi Perbaikan**: Hasil pengujian sistem beserta rekomendasi jika ditemukan masalah atau potensi perbaikan.