**DOKUMEN DESAIN PERANGKAT LUNAK   
(SOFTWARE DESIGN DOCUMENT - SDD)   
KEUANGAN DAN PEMBAYARAN**

**1. Pendahuluan**

**1.1 Tujuan**

Dokumen ini bertujuan untuk memberikan panduan teknis dan arsitektural dalam pengembangan sistem Keuangan dan Pembayaran berbasis web. Sistem ini akan memfasilitasi pengelolaan pembayaran SPP/UKT, memonitor status pembayaran mahasiswa, dan menyajikan laporan keuangan akademik.

**1.2 Ruang Lingkup**

Sistem ini mencakup:

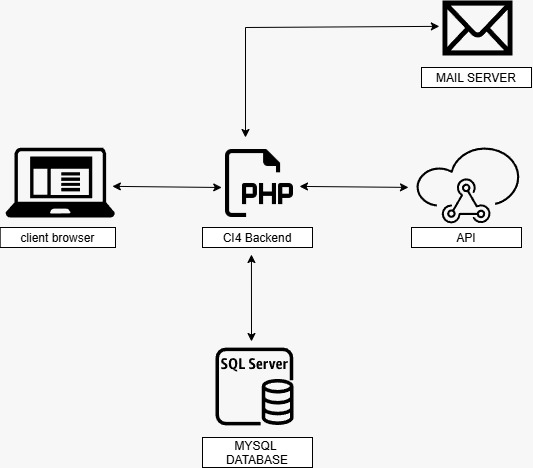
* Autentikasi & otorisasi pengguna (admin dan mahasiswa).
* Pengelolaan data pembayaran dan tagihan.
* Monitoring status pembayaran mahasiswa.
* Laporan keuangan dalam bentuk grafik dan file.
* Integrasi API untuk notifikasi pembayaran.
* Backup data otomatis.

**1.3 Referensi**

* CI4 Documentation.
* Bootstrap 5 Design Guide.
* OWASP Security Practices.
* Chart.js for data visualization.
* IEEE 1016-2009 for SDD format.

**2. Desain Arsitektur**

**2.1 Diagram Arsitektur Sistem**



* 1. **Client (Browser)**
* Pengguna akhir (admin atau mahasiswa) mengakses sistem melalui web browser.
* Mengirim permintaan (request) ke server dan menerima respons (response) dalam bentuk HTML atau data JSON (untuk AJAX/API).
  1. **CI4 Backend (Web Server)**
* Framework **CodeIgniter 4** menangani logika aplikasi menggunakan pola **MVC (Model-View-Controller)**.
* Komponen Controller menerima permintaan dari client dan menghubungkan ke Model dan View.
  1. **Database (MySQL)**
* Menyimpan data pengguna, tagihan, pembayaran, dan log aktivitas.
* Terhubung langsung ke backend melalui Model.
  1. **RESTful API**
* API endpoint digunakan untuk komunikasi internal antar modul atau eksternal (misalnya integrasi sistem pembayaran pihak ketiga).
* Contoh: GET /api/invoices/student/{id}, POST /api/payment, dll.
  1. **Mail Server**
* Digunakan untuk mengirim email notifikasi kepada mahasiswa atau admin, seperti:
  + Pemberitahuan status pembayaran.
  + Pengingat tagihan jatuh tempo.
  1. **Alur Komunikasi:**
* **Browser → Web Server (CI4)**: Mengirim request seperti login, cek tagihan, dsb.
* **CI4 → Database**: Ambil/simpan data.
* **CI4 → API Layer**: Jika data perlu diakses melalui API (internal atau eksternal).
* **CI4 → Mail Server**: Untuk mengirim notifikasi ke email pengguna.

**2.2 Komponen Utama Sistem**

* **Model:** Representasi entitas data (Mahasiswa, Tagihan, Pembayaran).
* **View:** UI responsif menggunakan Bootstrap.
* **Controller:** Logika rute dan interaksi frontend-backend.
* **API Layer:** REST API untuk integrasi & layanan internal.
* **Mail:** Modul pengirim notifikasi email.
* **Log & Monitoring:** Audit aktivitas dan status pembayaran.

**3. Desain Modul dan Komponen**

**3.1 Modul Autentikasi**

* **Fungsi:** Login, logout, dan reset password.
* **Input:** Email/NIM, password.
* **Teknologi:** JWT Session Token, Google reCAPTCHA.

**3.2 Modul Tagihan & Pembayaran**

* **Fungsi:** CRUD tagihan dan riwayat pembayaran.
* **Fitur Tambahan:** Upload bukti pembayaran.
* **Validasi:** Status otomatis setelah dikonfirmasi admin.

**3.3 Modul Monitoring**

* **Fungsi:** Lihat status pembayaran tiap mahasiswa.
* **Filter:** Berdasarkan bulan, jurusan, angkatan.
* **Output:** Tabel & grafik status pembayaran.

**3.4 Modul Laporan Keuangan**

* **Output:** Grafik pemasukan bulanan, file PDF & Excel
* **Tools:** Chart.js,.

**3.5 Modul Notifikasi & API**

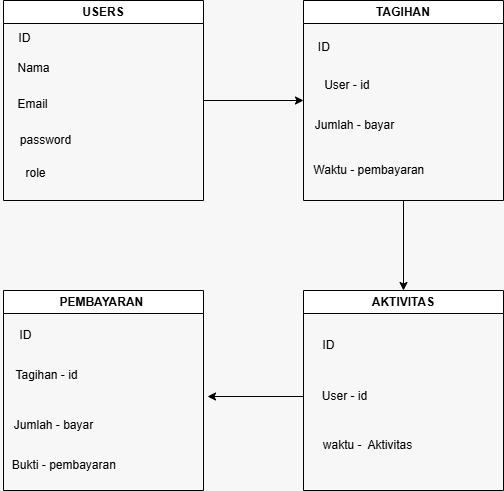
* **API:** REST untuk integrasi mobile/layanan eksternal.
* **Notifikasi:** Email otomatis saat tagihan dibuat/dibayar.

**4. Desain Basis Data**

**4.1 Struktur Tabel Utama**

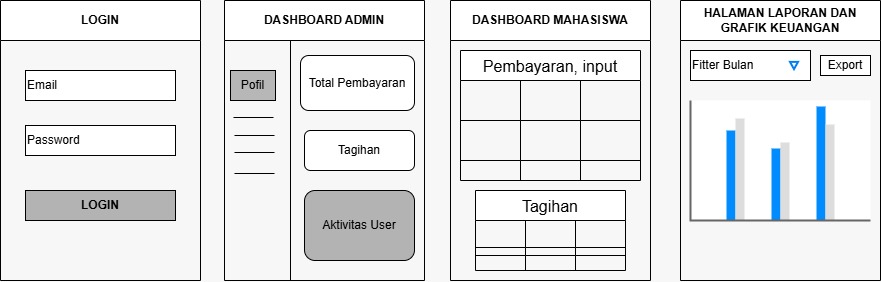
* Users (Id, Name, Email, Role, Password,)
* Tagihan (Id, User\_Id, Jumlah-Bayar, Waktu-Pembayaran)
* Pembayaran (Id, Tagihan\_Id, User\_Id, , Jumlah-Bayar, Bukti-Pembayaran)
* Aktivitas (Id, User\_Id, Waktu-Aktivitas)

**4.2 ERD**



**5. Antarmuka Pengguna**

**5.1 Wireframe**



* **Login Page:** Form login dengan notifikasi error.
* **Dashboard Admin:** Grafik total pemasukan, status pembayaran, filter jurusan.
* **Dashboard Mahasiswa:** Daftar tagihan & status pembayaran serta Form upload bukti dan konfirmasi pembayaran.
* **Laporan Keuangan:** Grafik pemasukan & download laporan.

**5.2 Responsif**

* Bootstrap 5, Flexbox layout, AJAX untuk aksi tanpa reload halaman.

**6. Keamanan**

* **Enkripsi Data:** Password dengan enskripsi.
* **RBAC:** Hanya admin yang bisa validasi pembayaran.
* **Log Aktivitas:** Tercatat dalam tabel logs.
* **Firewall Web:** SQL Injection.

**7. Pengujian dan Validasi**

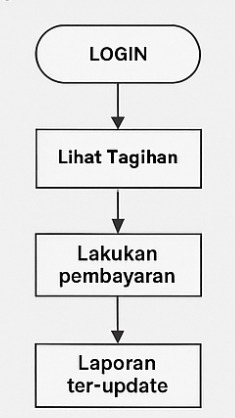
**7.1 Unit Testing**

* Pengujian per modul dengan PHP Unit.

**7.2 Pengujian Keamanan**

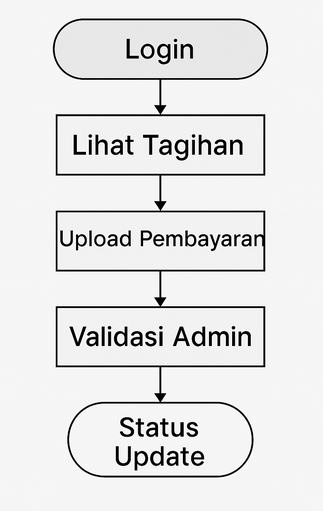
* Pengujian penetrasi untuk mengidentifikasi kerentanan sistem.
* Uji beban untuk memastikan sistem dapat menangani lonjakan pengguna.

**7.3 Pengujian Fungsional**



**8. Lampiran**

**8.1 Diagram Alur Proses (Flowchart)**



**8.2 Daftar API Endpoint**

