



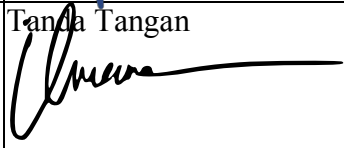


Topik Capstone	Smart business	
Siklus / Tahun	2 / 2024	
Judul Dokumen	Capstone TA Pengembangan Aplikasi Sistem Informasi Diniyah pada Pondok Pesantren Nurul Hikmah	
Jenis Dokumen	DESAIN PRODUK Catatan: Penggunaan dan penyebaran dokumen ini dikendalikan oleh Departemen Teknik Komputer Universitas Diponegoro	
Nomor Dokumen	C300.S2T24K04	
Nomor Revisi	01	
Nama File	C300.S2T24K04.docx	
Tanggal Penerbitan	4 Desember 2024	
Unit Penerbit	Departemen Teknik Komputer Universitas Diponegoro	
Jumlah Halaman		Tidak termasuk sampul

Data Pengusul			
Pengusul	Nama NIM	Hilmy Dhiya Ulhaq 21120121120008	Jabatan Anggota 
	Tanggal		Tanda Tangan
	Nama NIM	Dimas Fajar Awaludin 21120121120016	Jabatan Anggota 
	Tanggal		Tanda Tangan
	Nama NIM	Feri Syahrul Arifin 21120121120022	Jabatan Anggota 
	Tanggal		Tanda Tangan
Pembimbing 1 (Utama)	Nama	Rinta Kridalukmana, S.Kom., M.T., Ph.D NIP. 197706152008011011	Tanda Tangan 
Pembimbing 2	Nama	Ilmam Fauzi Hashbil Alim, S.T., M.Kom. NPPU.7.199611182022101001	Tanda Tangan 
	Tanggal		

Daftar Isi

1. Pendahuluan	5
1.1. Ringkasan isi dokumen.....	5
1.2. Aplikasi Dokumen	5
1.3. Referensi	5
1.4. Daftar Singkatan.....	6
2. Pemilihan Desain Produk.....	6
2.1. Alternatif Solusi	6
2.1.1. Solusi 1	6
2.1.2. Solusi 2	7
2.1.3. Solusi 3	7
2.2. Proses Pemilihan Solusi	7
3. Desain Produk yang Diusulkan.....	13
3.1. Arsitektur Sistem.....	13
3.2. Desain Detail Sistem	14
3.2.1. Data description.....	14
3.2.2. Class Diagram/Sequence Diagram atau diagram yang sesuai	Error! Bookmark not defined.
3.2.3. Standar-standar yang dipergunakan	21
3.2.4. Method/API	22
3.3. Traceable.....	23
3.3.1. Data.....	23
3.3.2. Requirements.....	24
4. Verifikasi Desain Produk	24
4.1. Hasil Simulasi Awal Produk.....	24
5. Rencana Implementasi dan Pengujian.....	31
5.1. Gaant Chart	31
5.2. Metode pengujian	31
5.2.1. Black Box Testing	31
5.2.2. Performance Testing (Pengujian Kinerja)	32
5.2.3. Response Time Testing	32
5.2.4. Bandwidth Testing.....	32
5.2.5. White Box Testing	32

Daftar Gambar

Gambar 3. 1 Arsitektur Aplikasi ALSID.....	13
Gambar 3. 2 ER Diagram Aplikasi ALSID	20
Gambar 4. 1 Halaman Login	25
Gambar 4. 2 Halaman Detail Santri	26
Gambar 4. 3 Halaman Edit User	26
Gambar 4. 4 Gambaran Halaman Manajemen Kelas.....	27
Gambar 4. 5 Halaman Menambahkan Tahun Ajaran	27
Gambar 4. 6 Halaman Perizinan	28
Gambar 4. 7 Halaman Request Perizinan.....	28
Gambar 4. 8 History Perizinan	29
Gambar 4. 9 Delete Perizinan	29
Gambar 4. 10 List Pembayaran.....	30
Gambar 5. 1 Gantt Chart Tugas Akhir	31

Daftar Tabel

Tabel 1. 1 Tabel Daftar Singkatan	6
Tabel 2. 1 Tabel Pemilihan Solusi	8
Tabel 3. 1 Hubungan Komponen Arsitektur Sistem.....	14
Tabel 3. 2 Tabel Admin.....	15
Tabel 3. 3 Tabel Santri	15
Tabel 3. 4 Tabel Kelas.....	16
Tabel 3. 5 Tabel Jadwal.....	16
Tabel 3. 6 Tabel Mapel	17
Tabel 3. 7 Tabel Nilai.....	17
Tabel 3. 8 Tabel Transkrip	17
Tabel 3. 9 Tabel Keuangan.....	18
Tabel 3. 10 Tabel Izin	18
Tabel 3. 11 Tabel Wifi	19
Tabel 3. 12 Tabel Log Aktivitas	19
Tabel 3. 13 Tabel Guru	20

Versi, Tanggal, Oleh	Perbaikan
1, 17 Desember 2024, Dosen Pembimbing 2	<ol style="list-style-type: none"> 1. Menambahkan indikator alternatif solusi pada proses pemilihan solusi. 2. Menambahkan mikrotik pada arsitektur sistem. 3. Menambahkan panjang data pada tipe data. 4. Menambahkan jenis enkripsi pada fitur autentikasi. 5. Penambahan caption gambar dan tabel. 6. Menyesuaikan kembali Gantt Chart

1. Pendahuluan

1.1. Ringkasan isi dokumen

Dokumen ini memaparkan informasi mengenai alasan pembuatan, analisa sistem, kebutuhan, dan usulan proses pengembangan dari sistem yang akan dibuat. Dokumen ini juga membahas seluruh aspek dari desain produk yang diusulkan mulai dari arsitektur sistem hingga rencana implelementasi serta pengujian yang akan dilakukan. Dokumen ini dapat membantu menjelaskan sistem kepada pengguna. Dengan demikian, pembaca diharapkan dapat mengerti proses pengembangan dan solusi yang ditawarkan oleh penulis.

Selain itu, dokumen ini dapat digunakan sebagai dasar proses pengembangan sistem oleh penulis. Di dalam pengembangan sistem terdapat sebuah proses evaluasi ataupun hasil pengembangan dan dokumen ini dapat menjadi sebuah bahan acuan evaluasi dari pengembangan sistem yang dilakukan.

1.2. Aplikasi Dokumen

Dokumen ini berlaku berfungsi untuk menjelaskan:

- 1) Proses pemilihan desain alat dari beberapa alternatif yang ada.
- 2) Detail desain alat dari level tertinggi sampai terendah
- 3) Menjelaskan standar-standar yang dipergunakan
- 4) Refensi komponen/library yang digunakan
- 5) Verifikasi bahwa hasil rancangan dapat diaplikasikan
- 6) Rencana implementasi dan pengujian

1.3. Referensi

- [1] D. Aipina and H. Witriyono, "Pemanfaatan Framework Laravel Dan Framework Bootstrap Pada Pembangunan Aplikasi Penjualan Hijab Berbasis Web," *J. Media Infotama*, vol. 18, no. 1, p. 2022, 2022.
- [2] D. D. Randa, Y. M. Putra, and H. Sammir, "Implementasi framework codeigniter untuk sistem informasi potensi dan peluang investasi (studi kasus di dinas DPMPTSP provinsi Sumatera Barat)," *JRTI (Jurnal Ris. Tindakan Indones.*, vol. 8, no. 1, p. 87, 2023, doi: 10.29210/30033051000.
- [3] W. C. Umbu Dagha, "Web Event, Spring Boot, Java Pembangunan Aplikasi Web Event menggunakan Framework Spring Boot di PT XYZ," *JATISI (Jurnal Tek. Inform. dan Sist.*

1.4. Daftar Singkatan

Tabel 1. 1 Tabel Daftar Singkatan

AEM	<i>Adobe Experience Manager</i>
MVC	<i>Model, View, Controller</i>
ORM	<i>Object-Relational Mapping</i>
PHP	<i>Hypertext Preprocessor</i>
TCP	<i>Transmission Control Protocol</i>
ALSID	Aplikasi Sistem Informasi Diniyah

2. Pemilihan Desain Produk

2.1. Alternatif Solusi

Dalam Pengembangan ALSID, terdapat tiga alternatif pengembangan yang dapat digunakan, berikut merupakan alternatif solusi pengembangan ALSID

2.1.1. Solusi 1

Pengembangan Aplikasi Sistem Informasi Diniyah (ALSID) pada Pondok Pesantren Nurul Hikmah dikembangkan menggunakan kerangka kerja Laravel, yang berbasis PHP dan bersifat open-source. Laravel memiliki arsitektur MVC (*Model-View-Controller*). Controller akan memproses request user dengan mencari data sesuai permintaan di Model, kemudian menampilkan hasilnya melalui View. Kemudian mendukung ORM (Object Relational Mapping) database akan diolah dan dimanipulasi datanya menjadi sebuah object. Laravel cocok untuk aplikasi berskala kecil hingga besar dengan kebutuhan pengelolaan data yang kompleks[1].

2.1.2. Solusi 2

Pengembangan Aplikasi Sistem Informasi Diniyah (ALSID) pada Pondok Pesantren Nurul Hikmah menggunakan kerangka kerja CodeIgniter. Codeigniter merupakan web application network yang bersifat opensource yang digunakan untuk sebuah aplikasi dengan menggunakan Bahasa pemrograman PHP[2]. Kerangka kerja CodeIgniter memiliki sebuah arsitektur MVC (*Model-View-Controller*). tetapi lebih ringan dibanding Laravel. Kelebihannya adalah kecepatan dalam pengembangan proyek kecil hingga menengah, namun kurang mendukung fitur bawaan seperti ORM sehingga memerlukan lebih banyak kode untuk mengelola data kompleks.

2.1.3. Solusi 3

Pengembangan Aplikasi Sistem Informasi Diniyah (ALSID) pada Pondok Pesantren Nurul Hikmah menggunakan kerangka kerja *Spring Boot*. *Spring Boot* termasuk framework yang cukup ringan untuk digunakan, bersifat *open source*, memiliki cukup banyak modul - modul yang bisa digunakan untuk membantu pekerjaan pengembang aplikasi dan *Spring Boot* sendiri bisa dikombinasikan dengan bahasa pemrograman lain karena Spring Boot mendukung pembuatan aplikasi berbasis *restfull web Service* [3]. Spring Boot mendukung arsitektur MVC dan mikro-layanan (*microservices*), serta menyediakan integrasi bawaan dengan berbagai pustaka pengelolaan basis data. Dengan dukungan ORM, Spring Boot sangat cocok untuk aplikasi berskala besar dan sistem yang membutuhkan skalabilitas tinggi. Namun, pengembangannya memerlukan sumber daya dan waktu yang lebih besar dibanding Laravel dan CodeIgniter.

2.2. Proses Pemilihan Solusi

Dari alternatif solusi yang telah dijabarkan, perbandingan ketiga solusi dirangkum pada Tabel 2.1. Setelah mempertimbangkan kebutuhan sistem, spesifikasi teknis, dan batasan pengembangan, Laravel dipilih sebagai solusi terbaik. Laravel dipilih karena memiliki fitur bawaan yang lengkap, mendukung pengembangan aplikasi berskala kecil hingga besar, serta memiliki komunitas yang luas untuk mendukung pengembangan.

Tabel 2. 1 Tabel Pemilihan Solusi

Indikator	Solusi 1	Solusi 2	Solusi 3
Kerangka Kerja	Laravel adalah framework PHP modern dengan fokus pada pengembangan cepat dan fitur yang lengkap, termasuk ORM, routing, dan keamanan.	CodeIgniter adalah framework PHP yang ringan dan minimalis, cocok untuk aplikasi sederhana dengan kebutuhan dasar.	Spring Boot adalah framework berbasis Java yang mendukung pengembangan aplikasi kompleks dengan RESTful API dan mikro-layanan.
Arsitektur	Laravel menggunakan arsitektur MVC (Model-View-Controller) dengan integrasi Blade sebagai engine untuk tampilan.	CodeIgniter juga mengadopsi arsitektur MVC, tetapi implementasinya lebih fleksibel dan tidak memaksa pengguna untuk menggunakannya.	Spring Boot mendukung MVC, RESTful API, dan arsitektur mikro-layanan untuk aplikasi dengan kebutuhan yang lebih kompleks.
Basis Teknologi	Laravel menggunakan PHP, bahasa populer yang sering digunakan untuk aplikasi web dengan banyak komunitas dan library pendukung.	CodeIgniter juga berbasis PHP, tetapi lebih ringan dengan kebutuhan sistem yang minimal untuk pengembangan cepat.	Spring Boot menggunakan Java, yang dikenal karena skalabilitas tinggi dan cocok untuk aplikasi berskala besar atau enterprise.

Sifat	Open-source dengan dukungan komunitas besar, memungkinkan pengguna untuk berkontribusi dan memanfaatkan banyak plugin gratis.	Juga open-source, dengan dokumentasi yang mudah dipahami dan komunitas sedang, mendukung pemula dalam belajar framework.	Open-source dengan komunitas besar dan kuat, didukung oleh perusahaan seperti Pivotal dan VMware untuk kebutuhan enterprise.
Skala Proyek Web	Laravel mendukung pengembangan dari proyek kecil hingga besar, dengan fitur bawaan seperti migrasi database dan otentikasi.	CodeIgniter ideal untuk proyek kecil hingga menengah karena sederhana dan efisien tanpa memerlukan konfigurasi kompleks.	Spring Boot dirancang untuk proyek menengah hingga besar, khususnya yang membutuhkan keandalan tinggi dan fitur mikro-layanan.
Dukungan ORM	Laravel memiliki Eloquent ORM, mempermudah interaksi dengan database melalui model berbasis PHP.	Tidak memiliki dukungan ORM bawaan, tetapi pengembang dapat mengintegrasikan ORM secara manual jika diperlukan.	Mendukung ORM seperti Hibernate dan JPA secara bawaan, memungkinkan pengelolaan database yang lebih canggih.
Fitur Keamanan	Laravel menyediakan otentikasi bawaan, proteksi CSRF, hashing, dan validasi input yang terintegrasi.	CodeIgniter memiliki fitur keamanan dasar seperti validasi input, tetapi memerlukan konfigurasi manual untuk fitur lainnya.	Spring Boot memiliki fitur keamanan tingkat lanjut seperti Spring Security, cocok untuk aplikasi

			yang membutuhkan keamanan tinggi.
Dukungan API	Laravel memiliki Laravel Sanctum dan Passport untuk membangun API yang aman dan terstruktur.	Mendukung pengembangan API dasar, tetapi tidak memiliki fitur seperti OAuth bawaan.	Mendukung REST API secara bawaan, dengan kemampuan untuk membangun API yang aman dan efisien untuk berbagai aplikasi.
Dokumentasi	Laravel memiliki dokumentasi yang sangat lengkap, interaktif, dan didukung banyak tutorial komunitas.	Dokumentasi CodeIgniter cukup lengkap tetapi lebih sederhana dibandingkan Laravel.	Dokumentasi Spring Boot sangat komprehensif, mencakup semua fitur dan banyak studi kasus untuk pengembang.
Komunitas	Komunitas besar dan aktif, sehingga mudah menemukan solusi, tutorial, atau library tambahan untuk pengembangan.	Komunitas cukup besar tetapi lebih kecil dibandingkan Laravel, lebih cocok untuk pengembangan sederhana.	Komunitas besar dengan banyak kontribusi dari pengembang enterprise dan organisasi besar.
Dukungan Testing	Laravel mendukung PHPUnit untuk unit testing dan Laravel Dusk untuk pengujian browser secara otomatis.	Mendukung PHPUnit tetapi pengujian tingkat lanjut memerlukan integrasi eksternal.	Spring Boot mendukung JUnit untuk pengujian unit dan MockMvc untuk pengujian API secara bawaan.

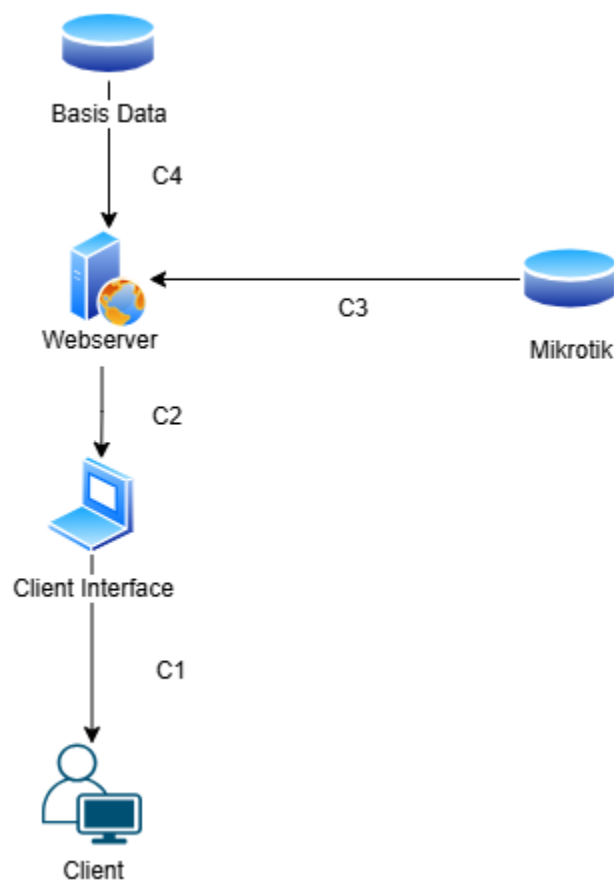
Kemudahan Penggunaan	Laravel mudah digunakan dengan banyak fitur bawaan dan sintaks yang ramah bagi pengembang, meskipun pemula sekalipun.	CodeIgniter sederhana dan cocok untuk pemula, tetapi memiliki keterbatasan untuk fitur kompleks.	Spring Boot cukup kompleks bagi pemula tetapi sangat fleksibel dan kuat untuk pengembang berpengalaman.
Performance	Laravel performa baik tetapi memerlukan optimasi untuk menangani aplikasi dengan lalu lintas tinggi.	CodeIgniter sangat cepat untuk aplikasi kecil dengan beban ringan.	Spring Boot memiliki performa tinggi dan skalabilitas yang sangat baik untuk aplikasi berskala besar.
Deployment	Laravel mendukung berbagai opsi deployment, termasuk Laravel Forge untuk kemudahan pengelolaan server.	Deployment sederhana dan dapat dilakukan secara manual tanpa alat khusus.	Mendukung deployment modern menggunakan Docker, Kubernetes, atau cloud provider seperti AWS dan GCP.
Modularitas	Laravel sangat modular dengan dukungan paket melalui Composer.	CodeIgniter kurang modular dibandingkan Laravel, tetapi tetap mendukung struktur folder yang rapi.	Spring Boot sangat modular, mendukung pengembangan dengan arsitektur mikro-layanan berbasis modul.
Penyimpanan Cache	Laravel mendukung Redis, Memcached,	CodeIgniter hanya memiliki fitur cache	Spring Boot mendukung berbagai

	dan penyimpanan cache bawaan untuk meningkatkan performa.	dasar tanpa integrasi yang kompleks.	opsi caching seperti Redis, Ehcache, dan lainnya.
Skalabilitas	Laravel cukup skalabel tetapi memerlukan konfigurasi tambahan untuk menangani lalu lintas sangat tinggi.	CodeIgniter kurang cocok untuk aplikasi dengan kebutuhan skalabilitas tinggi.	Spring Boot sangat skalabel, ideal untuk aplikasi enterprise dengan kebutuhan multi-instance dan arsitektur mikro-layanan.
Ekosistem Library	Laravel memiliki ekosistem library yang kaya, mudah diintegrasikan melalui Composer.	CodeIgniter memiliki ekosistem library yang terbatas tetapi cukup untuk aplikasi kecil hingga menengah.	Spring Boot memiliki ekosistem library yang luas, termasuk modul Spring untuk berbagai kebutuhan.
Integrasi Database	Laravel mendukung berbagai database seperti MySQL, PostgreSQL, SQLite, dan SQL Server.	Mendukung MySQL dan PostgreSQL tetapi kurang fleksibel dibandingkan Laravel.	Mendukung berbagai database termasuk MySQL, PostgreSQL, MongoDB, dan database NoSQL lainnya.
Kompatibilitas Cloud	Laravel mendukung integrasi dengan layanan cloud seperti AWS, GCP, dan Azure.	CodeIgniter memiliki kompatibilitas terbatas dengan layanan cloud modern.	Spring Boot sangat mendukung integrasi cloud dengan Spring Cloud dan berbagai layanan lainnya.

3. Desain Produk yang Diusulkan

3.1. Arsitektur Sistem

Dalam perancangan perangkat lunak, arsitektur sistem diperlukan untuk mendefinisikan serta memberikan gambaran dari elemen-elemen penting yang ada pada sistem perangkat lunak. Berikut merupakan gambar Arsitektur Sistem ALSID.



Gambar 3. 1 Arsitektur Aplikasi ALSID

Pada arsitektur sistem informasi yang telah dimodifikasi untuk integrasi MikroTik, terdapat beberapa hubungan utama antar komponen. Hubungan pertama adalah C1, di

mana pengguna seperti santri, admin, dan pengajar berinteraksi dengan sistem melalui *Client Interface* menggunakan perangkat seperti komputer atau smartphone. Hubungan kedua adalah C2, di mana *Client Interface* berkomunikasi dengan *web server* untuk mengirimkan permintaan seperti autentikasi atau pengajuan izin. Hubungan ketiga adalah C3, di mana *web server* terhubung dengan perangkat MikroTik untuk autentikasi pengguna melalui *MAC address*, manajemen *hotspot*, atau pengambilan *log* jaringan menggunakan API. Hubungan terakhir adalah C4, di mana *web server* berkomunikasi dengan basis data untuk mengelola informasi santri, pembayaran, jadwal, dan laporan. Penjelasan detail ketiga hubungan tersebut dapat dilihat pada tabel berikut:

Tabel 3. 1 Hubungan Komponen Arsitektur Sistem

Kode	Komponen Pengirim	Komponen Penerima	Media Transmisi	Metode Transmisi Data
C1	<i>User</i>	<i>Client Interface</i>	Internet	<i>HTTP Post</i> atau <i>HTTP Get</i>
C2	<i>Client Interface</i>	<i>Web Server</i>	Internet	<i>HTTP Post</i> atau <i>HTTP Get</i>
C3	<i>Web Server</i>	MikroTik	Internet	<i>HTTP Post</i> melalui <i>API</i>
C4	<i>Web Server</i>	Basis Data	Internet/Intranet	<i>HTTP Post</i> atau <i>Query Database</i>

3.2. Desain Detail Sistem

3.2.1. Data description

Basis data untuk pengembangan Sistem Informasi Diniyah pada Pondok Pesantren Nurul Hikmah memiliki dua belas tabel basis data yaitu *admin*, *santri*, *kelas*, *jadwal*, *mapel*, *nilai*, *transkrip*, *keuangan*, *izin*, *wifi*, *LogAktifitas*, dan *Guru*. Penjabaran lebih rinci mengenai struktur basis data dapat dilihat pada tabel-tabel di bawah.

Tabel 3. 2 Tabel Admin

Nama Kolom	Tipe Data	Constraint	Deskripsi
ID_Admin	SERIAL	PRIMARY KEY	ID unik untuk Admin
Username	VARCHAR(50)	NOT NULL	Nama pengguna untuk login
Password	VARCHAR(255)	NOT NULL	Kata sandi pengguna
Nama	VARCHAR(100)	NOT NULL	Nama lengkap Admin
ID_Admin	SERIAL	PRIMARY KEY	ID unik untuk Admin

Tabel 3. 3 Tabel Santri

Nama Kolom	Tipe Data	Constraint	Deskripsi
ID_Santri	SERIAL	PRIMARY KEY	ID unik untuk Santri
Nama	VARCHAR(150)	NOT NULL	Nama lengkap Santri
Tanggal_Lahir	DATE	NOT NULL	Tanggal lahir Santri
Alamat	TEXT	NOT NULL	Alamat tempat tinggal Santri
NIS	VARCHAR(15)	UNIQUE, NOT NULL	Nomor Induk Santri
MAC_Address	VARCHAR(17)	UNIQUE, NOT NULL	Alamat MAC untuk WiFi
ID_Kelas	INT	FOREIGN KEY	Referensi ke Kelas.ID_Kelas

Tabel 3. 4 Tabel Kelas

Nama Kolom	Tipe Data	Constraint	Deskripsi
ID_Kelas	SERIAL	PRIMARY KEY	ID unik untuk Kelas
Nama_Kelas	VARCHAR(50)	NOT NULL	Nama kelas
Tingkat	INT	NOT NULL	Tingkat kelas (1, 2, 3)
Wali_Kelas	VARCHAR(100)	NOT NULL	Nama wali kelas
Ketua_Kelas	VARCHAR(100)	NOT NULL	Nama ketua kelas

Tabel 3. 5 Tabel Jadwal

Nama Kolom	Tipe Data	Constraint	Deskripsi
ID_Jadwal	SERIAL	PRIMARY KEY	ID unik untuk Jadwal
Hari	VARCHAR(10)	NOT NULL	Hari pelaksanaan jadwal
Jam	TIME	NOT NULL	Jam pelaksanaan jadwal
ID_Mapel	INT	FOREIGN KEY	Referensi ke Mapel.ID_Mapel
ID_Kelas	INT	FOREIGN KEY	Referensi ke Kelas.ID_Kelas
ID_Guru	INT	FOREIGN KEY	Referensi ke Guru.ID_Guru

Tabel 3. 6 Tabel Mapel

Nama Kolom	Tipe Data	Constraint	Deskripsi
ID_Mapel	SERIAL	PRIMARY KEY	ID unik untuk Mapel
Nama_Mapel	VARCHAR(100)	NOT NULL	Nama mata pelajaran
Keterangan	TEXT	NULLABLE	Keterangan tambahan

Tabel 3. 7 Tabel Nilai

Nama Kolom	Tipe Data	Constraint	Deskripsi
ID_Nilai	SERIAL	PRIMARY KEY	ID unik untuk Nilai
ID_Santri	INT	FOREIGN KEY	Referensi ke Santri.ID_Santri
ID_Mapel	INT	FOREIGN KEY	Referensi ke Mapel.ID_Mapel
Tahun_Akademik	VARCHAR(9)	NOT NULL	Tahun akademik penilaian
Nilai	FLOAT	NOT NULL	Nilai yang diperoleh

Tabel 3. 8 Tabel Transkrip

Nama Kolom	Tipe Data	Constraint	Deskripsi
ID_Transkrip	SERIAL	PRIMARY KEY	ID unik untuk Transkrip
ID_Santri	INT	FOREIGN KEY	Referensi ke Santri.ID_Santri
Semester	INT	NOT NULL	Semester akademik
Tahun_Akademik	VARCHAR	NOT NULL	Tahun akademik

Tabel 3. 9 Tabel Keuangan

Nama Kolom	Tipe Data	Constraint	Deskripsi
ID_Keuangan	SERIAL	PRIMARY KEY	ID unik untuk Keuangan
ID_Santri	INT	FOREIGN KEY	Referensi ke Santri.ID_Santri
Tanggal	DATE	NOT NULL	Tanggal transaksi
Jumlah	NUMERIC(12,2)	NOT NULL	Jumlah transaksi
Jenis	VARCHAR(50)	NOT NULL	Jenis transaksi

Tabel 3. 10 Tabel Izin

Nama Kolom	Tipe Data	Constraint	Deskripsi
ID_Izin	SERIAL	PRIMARY KEY	ID unik untuk Izin
ID_Santri	INT	FOREIGN KEY	Referensi ke Santri.ID_Santri
Tanggal	DATE	NOT NULL	Tanggal izin
Keterangan	TEXT	NOT NULL	Keterangan izin
Status	VARCHAR(20)	NOT NULL	Status izin (disetujui/ditolak)

Tabel 3. 11 Tabel Wifi

Nama Kolom	Tipe Data	Constraint	Deskripsi
ID_WiFi	SERIAL	PRIMARY KEY	ID unik untuk WiFi
ID_Santri	INT	FOREIGN KEY	Referensi ke Santri.ID_Santri
MAC_Address	VARCHAR(17)	UNIQUE	Alamat MAC untuk WiFi
Tanggal_Registrasi	DATE	NOT NULL	Tanggal registrasi WiFi

Tabel 3. 12 Tabel Log Aktivitas

Nama Kolom	Tipe Data	Constraint	Deskripsi
ID_Log	SERIAL	PRIMARY KEY	ID unik untuk Log Aktivitas
Waktu	TIMESTAMP	NOT NULL	Waktu aktivitas terjadi
Aktivitas	VARCHAR(255)	NOT NULL	Deskripsi aktivitas
Detail	TEXT	NULLABLE	Detail tambahan aktivitas
ID_Admin	INT	FOREIGN KEY	Referensi ke Admin.ID_Admin

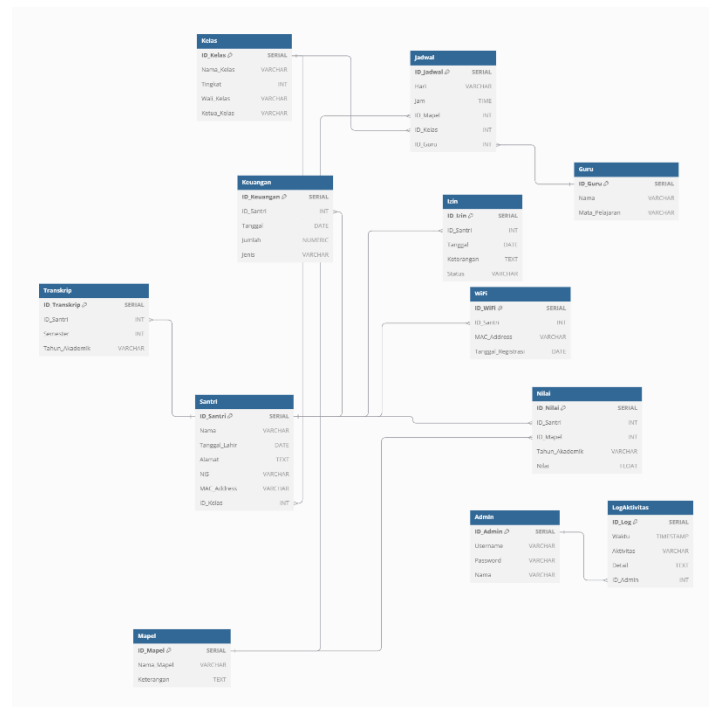
]

Tabel 3. 13 Tabel Guru

Tabel 3. 13 Tabel Guru

3.2.2. ER Diagram

ER Diagram merupakan Diagram yang digunakan untuk menjelaskan proses *use case* dan hubungan *use case* satu dengan yang lain pada sistem, ER Diagram ALSID dapat dilihat pada gambar di bawah



Gambar 3. 2 ER Diagram Aplikasi ALSID

3.2.3. Standar-standar yang dipergunakan

Sistem Informasi Diniyyah yang dikembangkan menggunakan PostgreSQL sebagai basis data menerapkan berbagai standar untuk memastikan keamanan, performa, dan integritas data. Dalam hal keamanan, koneksi antara client dan server dienkripsi menggunakan TLS dalam proses pembayaran, sementara bcrypt digunakan untuk menyimpan password dalam bentuk terenkripsi *hash*. PostgreSQL juga mendukung *Constraint* seperti *PRIMARY KEY*, *FOREIGN KEY*, dan *UNIQUE* serta *Triggers* untuk menjaga konsistensi dan integritas data.

Untuk performa, PostgreSQL memanfaatkan fitur *Indexing* dan optimasi *Query Execution Plan* agar proses pencarian dan pengambilan data lebih efisien. Selain itu, *Database Replication* diterapkan sebagai mekanisme cadangan untuk memastikan kecepatan akses dan ketersediaan data. Dengan dukungan *ORM Eloquent* dari Laravel, interaksi dengan PostgreSQL menjadi lebih sederhana dan terstruktur. Proses *backup* dan *recovery* data juga dilakukan menggunakan fitur *pg_dump* dan *pg_restore*. Dengan demikian, PostgreSQL memberikan keunggulan dalam aspek keamanan, efisiensi, dan skalabilitas, sehingga sistem dapat berjalan dengan *database backup* sesuai kebutuhan Pondok Pesantren Nurul Hikmah.

3.2.4. Method

Pada perancangan Sistem Informasi Diniyah Pondok Pesantren Nurul Hikmah, terdapat dua metode utama yang digunakan, yaitu (1) metode untuk mengakses dan berinteraksi dengan basis data, serta (2) metode untuk manipulasi dan pengolahan data. Sistem ini menggunakan basis data PostgreSQL dengan pendekatan RDBMS (Relational Database Management System). Untuk pengaksesan basis data, sistem memanfaatkan ORM (Object Relational Mapping) bawaan Laravel, yang menyederhanakan komunikasi antara aplikasi dan basis data.

Metode manipulasi data dilakukan melalui operasi CRUD (Create, Read, Update, Delete) yang mencakup:

- Create: Untuk membuat data baru, seperti menambahkan informasi santri atau jadwal pelajaran.
- Read: Untuk membaca dan menampilkan data yang ada dalam basis data, seperti melihat daftar santri, nilai akademik, atau laporan keuangan.
- Update: Untuk mengubah data yang ada, seperti memperbarui informasi perizinan atau pengelolaan kelas.
- Delete: Untuk menghapus data yang tidak lagi relevan, seperti menghapus data santri yang sudah tidak aktif.

Selain itu, sistem mendukung fungsi pencarian data menggunakan metode Search, yang mempermudah pengguna untuk menemukan informasi tertentu, seperti data santri atau jadwal pelajaran, berdasarkan kata kunci.

3.3. Traceable

3.3.1. Data

Hubungan tabel-tabel dalam basis-data dengan desain entity relationship yang telah dibuat dapat dilihat pada tabel

Tabel 3. 13 Hubungan Tabel dalam Basis Data

<i>Nama Tabel</i>	<i>Primary key</i>	<i>Entity Class</i>	<i>ER</i>	<i>Deskripsi isi</i>
admin	id_admin	Admin	<i>Master</i>	Berisi data pengguna dengan hak akses admin
santri	id_santri	Santri	<i>Master</i>	Berisi daftar data santri
kelas	id_kelas	Kelas	<i>Master</i>	Berisi informasi kelas, tingkat, dan wali kelas
jadwal	id_jadwal	Jadwal	<i>Need</i>	Menyimpan jadwal pelajaran dan pengajar
mapel	id_mapel	MataPelajaran	<i>Need</i>	Berisi informasi mata pelajaran
nilai	id_nilai	Nilai	<i>Transaction</i>	Menyimpan nilai akademik santri
keuangan	id_keuangan	Keuangan	<i>Transaction</i>	Berisi transaksi keuangan santri
izin	id_izin	Izin	<i>Transaction</i>	Menyimpan data izin santri
wifi	id_wifi	Wifi	<i>Need</i>	Berisi data registrasi MAC Address WiFi
logaktifitas	id_log	LogAktifitas	<i>Transaction</i>	Menyimpan catatan aktivitas admin
guru	id_guru	Guru	<i>Master</i>	Berisi daftar guru dan mata pelajaran yang diajar

3.3.2. Requirements

Hubungan *method* yang ada pada aplikasi ALSID dapat dilihat pada tabel:

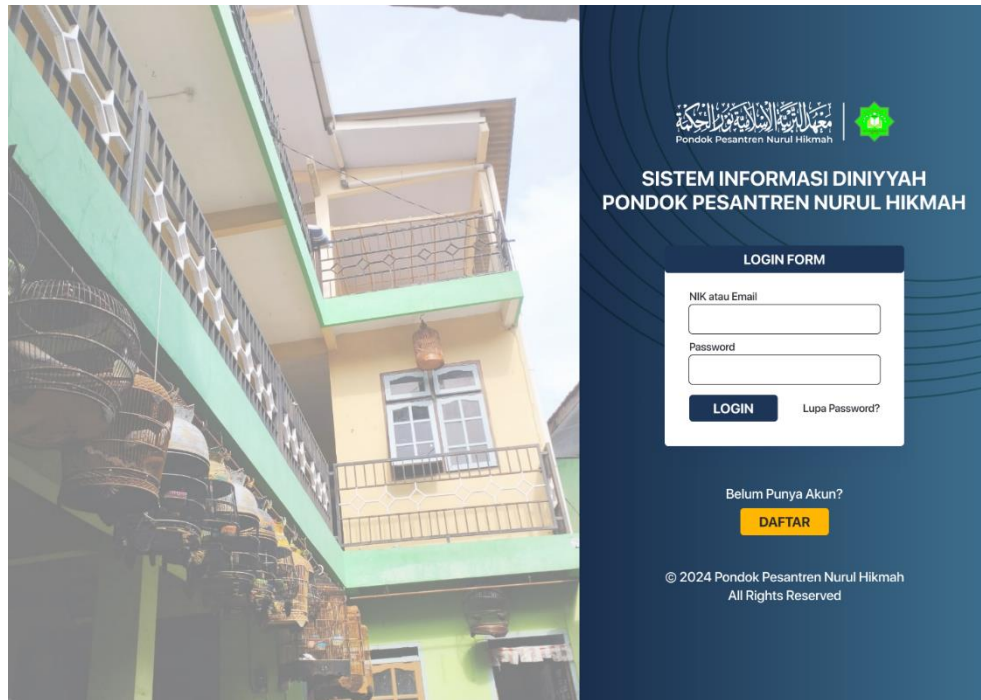
Tabel 3. 14 Hubungan Method

SRS-ID	Nama Method	Keterangan
SW-ALSID-SRS-0001	Register	Mendaftarkan pengguna baru ke dalam sistem dengan menyimpan data autentikasi.
SW-ALSID-SRS-0002	Login	Autentikasi pengguna dengan validasi username dan password.
SW-ALSID-SRS-0003	Create	Pengguna Mendambahkan data baru ke dalam sistem.
SW-ALSID-SRS-0004	Update	Pengguna memperbarui data yang ada di dalam sistem sesuai kebutuhan pengguna.
SW-ALSID-SRS-0005	Delete	Pengguna menghapus data yang tidak diperlukan dari sistem.
SW-ALSID-SRS-0006	Upload	Pengguna mengunggah file atau dokumen ke dalam sistem.
SW-ALSID-SRS-0007	Download	Pengguna mengunduh file atau laporan dari sistem.
SW-ALSID-SRS-0008	Logout	Pengguna mengakhiri sesi pengguna dengan aman dari sistem.

4. Verifikasi Desain Produk

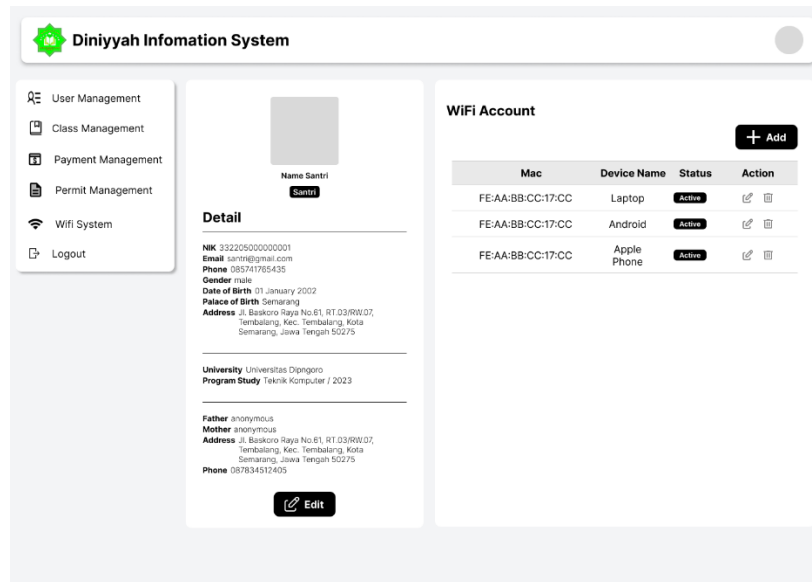
4.1. Hasil Simulasi Awal Produk

Simulasi Produk yang pertama adalah pada bagian *login* pada aplikasi ALSID yang ditunjukkan pada Gambar 4.1. Aplikasi akan menampilkan halaman login dan pengguna dapat memasukkan email dan password lalu menekan tombol Sign In. Jika proses login berhasil maka akan menampilkan halaman Dashboard, jika login gagal maka sistem akan menampilkan pesan *error*.

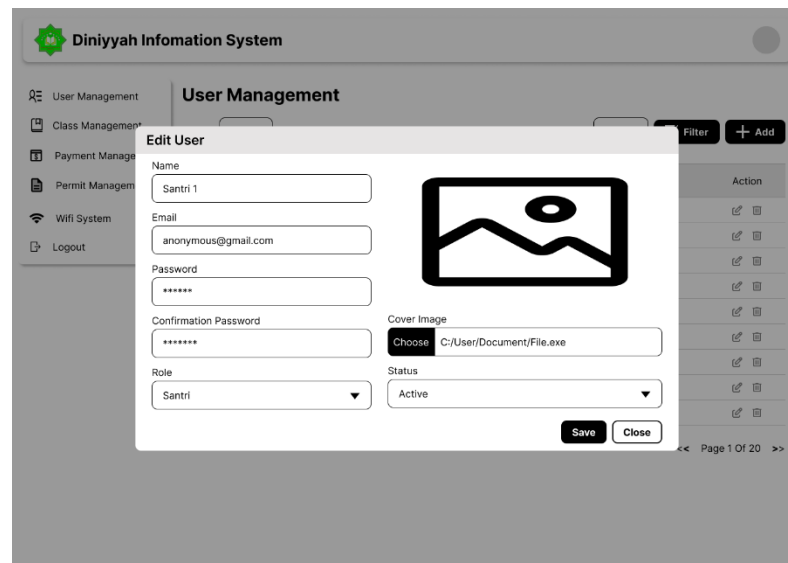


Gambar 4. 1 Halaman Login

Setelah Proses *Login* dan *Register* berhasil, maka sistem akan meneruskan pengguna menuju halaman dashboard yang mana pengguna dapat melakukan tugas sesuai *role* yang diberikan. Tampilan halaman *dashboard* dapat dilihat pada gambar.

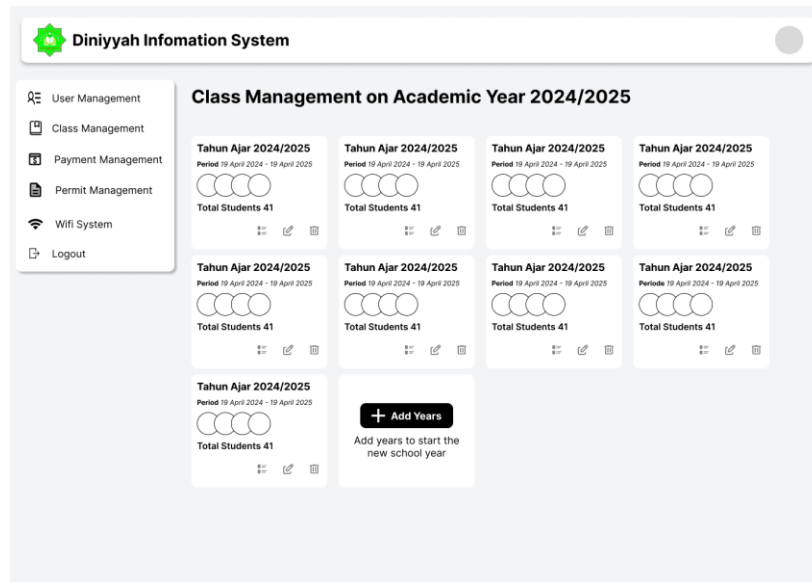


Gambar 4. 2 Halaman Detail Santri

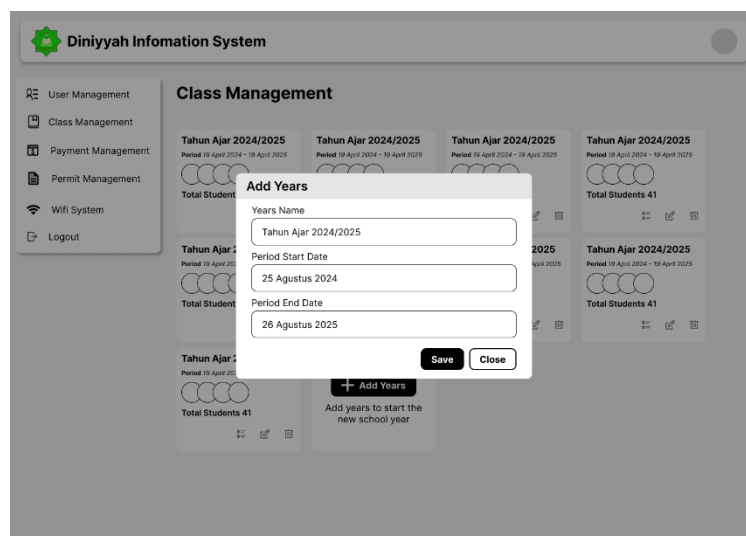


Gambar 4. 3 Halaman Edit User

Pada Halaman Tersebut Santri Juga dapat melakukan edit dari data yang sebelumnya telah dimasukkan melalui *card* yang ada pada fitur *register*. Selanjutnya terdapat menu kelas yang mana akan menampilkan kelas-kelas yang ada tahun ajaran tertentu. Pada guru dapat melihat detail total santri, namun dari santri hanya bisa melihat jumlah kelas dan total siswanya saja.



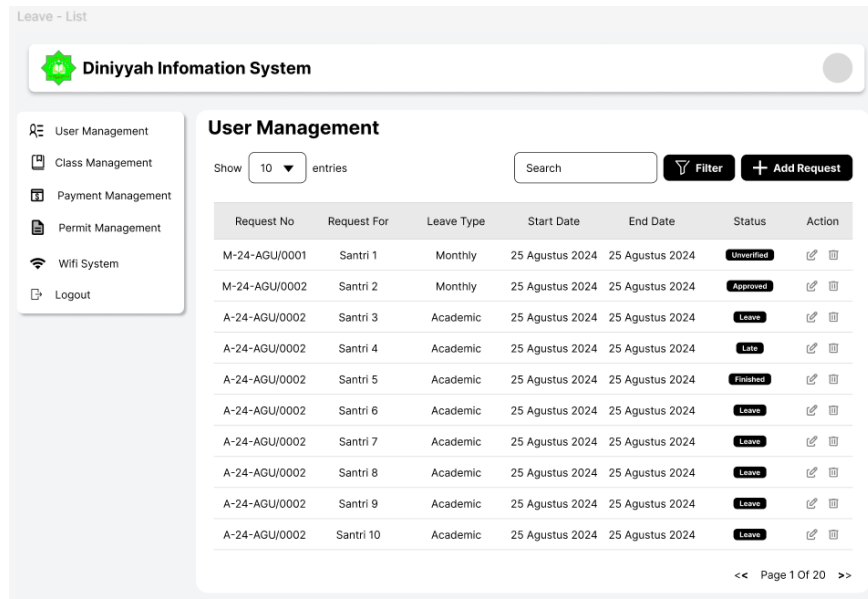
Gambar 4. 4 Gambaran Halaman Manajemen Kelas



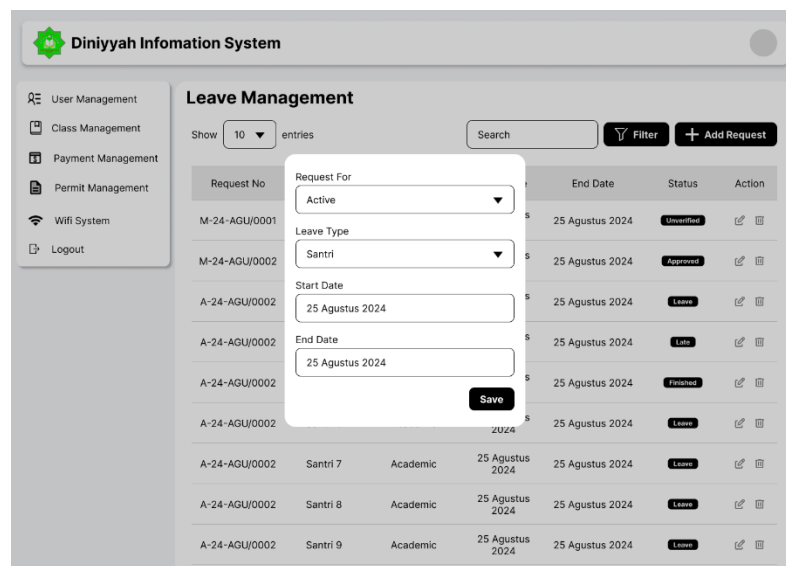
Gambar 4. 5 Halaman Menambahkan Tahun Ajaran

Pada gambar di atas merupakan gambaran pada sisi pengajar dan admin yang dapat melakukan penambahan tahun ajaran pada sistem informasi.

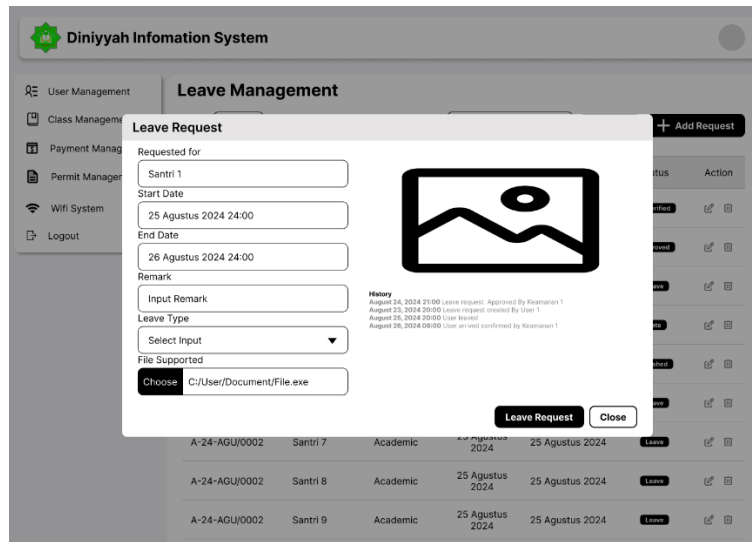
Selanjutnya merupakan fitur perizinan, yang mana bisa dilakukan oleh santri ketika akan dalam proses magang, sakit, dan perizinan lainnya, pada tab perizinan ini, terdapat history dari perizinan mencakup tanggal, dan bukti izin dapat berupa gambar ataupun *file upload*.



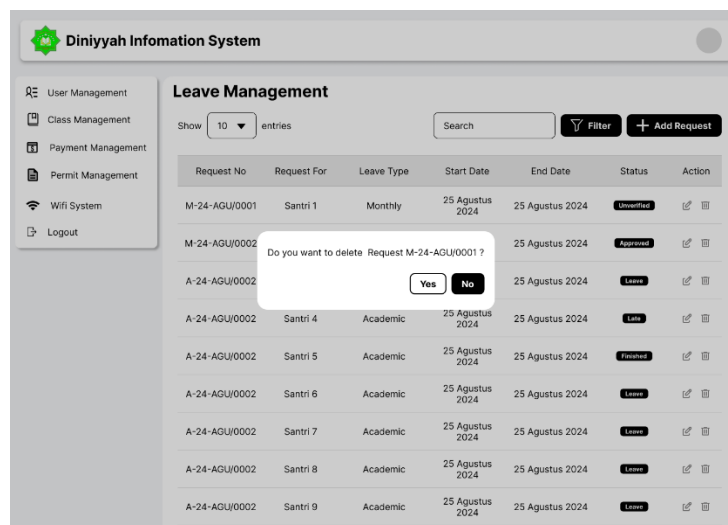
Gambar 4. 6 Halaman Perizinan



Gambar 4. 7 Halaman *Request* Perizinan



Gambar 4. 8 History Perizinan



Gambar 4. 9 Delete Perizinan

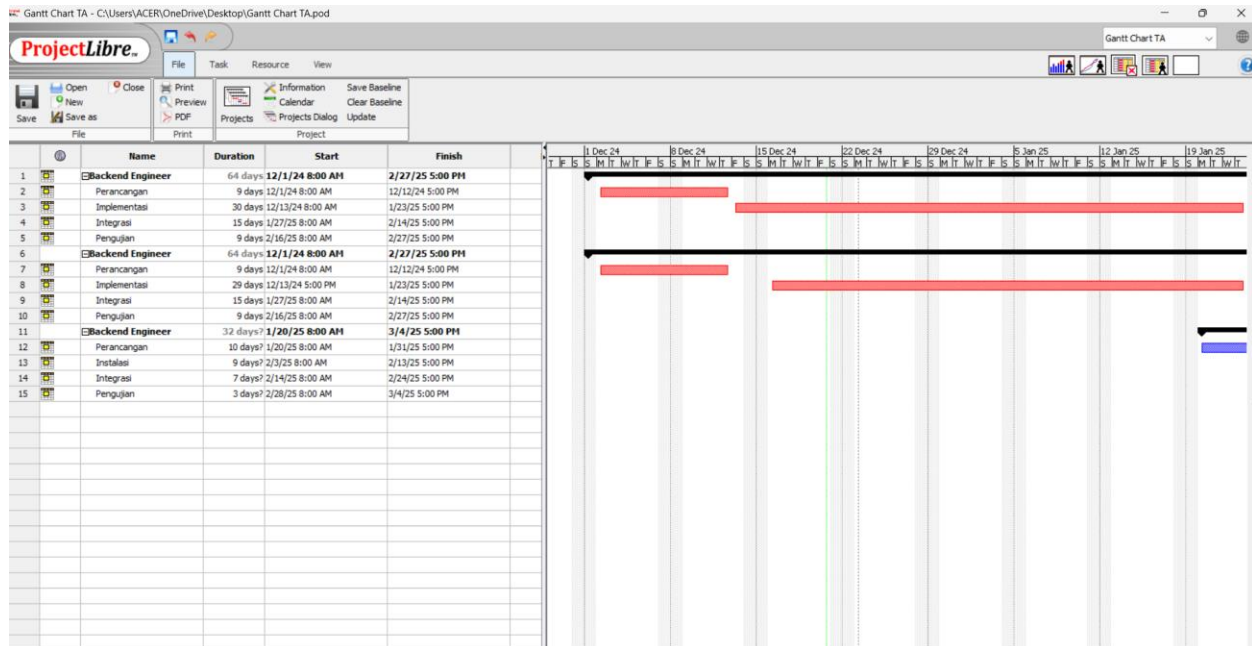
Pada Pembayaran terdapat list pembayaran yang dilakukan oleh santri pada tab pembayaran yang dapat dilihat oleh *admin*.

Gambar 4. 10 List Pembayaran

5. Rencana Implementasi dan Pengujian

5.1. Gantt Chart

Diagram Gantt dari pengembangan Sistem Informasi Diniyah pada Pondok Pesantren Nurul Hikmah Semarang digunakan sebagai petunjuk dependensi pekerjaan serta estimasi waktu pengerjaan proyek dapat dilihat pada Gambar ... Pada diagram Gantt tersebut tertera tugas-tugas yang perlu dikerjakan oleh para pengembang.



Gambar 5. 1 Gantt Chart Tugas Akhir

5.2. Metode pengujian

5.2.1. Black Box Testing

Black box testing merupakan pengujian kualitas perangkat lunak yang berfokus pada fungsionalitas perangkat lunak. Pengujian *black box* bertujuan untuk menemukan fungsi yang tidak benar, kesalahan antarmuka, kesalahan pada struktur data, kesalahan performansi, kesalahan inisialisasi dan terminasi. Pengujian *black box* merupakan pengujian yang didasarkan pada detail aplikasi seperti tampilan *website*, fungsi-fungsi yang ada pada *website*, dan kesesuaian alur fungsi dengan bisnis proses yang diinginkan oleh pengguna.

5.2.2. Performance Testing (Pengujian Kinerja)

Performance Testing adalah jenis pengujian untuk memastikan perangkat lunak akan bekerja dengan baik di bawah beban kerja yang diharapkan. *Performance Testing* (Pengujian Kinerja) mencakup berbagai metodologi yang secara kolektif memungkinkan penilaian daya tanggap, skalabilitas, dan stabilitas aplikasi perangkat lunak dalam berbagai kondisi.

5.2.3. Response Time Testing

Response Time Testing adalah pengujian waktu *respons* yang mengacu pada waktu yang diperlukan untuk satu node sistem untuk menanggapi permintaan dari yang lain. Waktu yang dibutuhkan ketika sistem atau unit mengambil langkah untuk bereaksi input tertentu sampai proses selesai. Waktu *respons* dimulai ketika pengguna mengirim permintaan dan berakhir saat waktu aplikasi menyatakan bahwa permintaan tersebut telah selesai. Cara mengukur waktu tanggapan respons diukur dengan bantuan alat tes dengan mencoba proses bisnis penting dari transaksi mulai hingga transaksi akhir. Contoh alat dari tesnya adalah JMeter, Load Runner, dan AEM. *Response time testing* memiliki dua karakteristik yang paling penting yaitu:

- Waktu respon rata-rata: waktu rata-rata yang diperlukan untuk menanggapi permintaan oleh pengguna sistem informasi.
- Waktu respon maksimum

Pada umumnya, waktu respons harus secepat mungkin yaitu dalam interval 0,1 – 1 detik. Namun, beberapa orang dapat beradaptasi dengan waktu respons yang lebih lambat, tetapi mereka tidak akan pernah senang dengan waktu tanggapan lebih dari 2 detik.

5.2.4. Bandwidth Testing

Bandwidth Testing dapat digunakan untuk mengukur *throughput* ke router MikroTik lain (baik kabel atau nirkabel) dan dengan demikian membantu menemukan "hambatan" jaringan. Pengujian tersebut menggunakan protokol TCP standar dengan mengikuti algoritma TCP, dengan proses yaitu mengirimkan beberapa paket yang sesuai dengan latensi.

5.2.5. White Box Testing

White Box Testing merupakan salah satu cara untuk menguji suatu aplikasi atau *software* dengan melihat modul untuk memeriksa dan menganalisa kode program apakah ada yang salah atau tidak.