

**PENGEMBANGAN SISTEM INFORMASI BERBASIS WEB
UNTUK MENDUKUNG PENGELOLAAN AKADEMIK DAN
ADMINISTRASI PADA MA'HAD AL-JAMI'AH
UIN AR-RANIRY BANDA ACEH**

TUGAS AKHIR

Diajukan Oleh :

Feril Ferdian
220705062

Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi



**FAKULTAS SAINS DAN TEKNOLOGI
UIVERSITAS ISLAM NEGERI AR-RANIRY
BANDA ACEH
2026/1448 M**

LEMBAR PERSETUJUAN

PENGEMBANGAN SISTEM INFORMASI BERBASIS WEB UNTUK Mendukung PENGELOLAAN AKADEMIK DAN ADMINISTRASI PADA MA'HAD AL-JAMI'AH UIN AR- RANIRY BANDA ACEH

TUGAS AKHIR

Diajukan Kepada Fakultas Sains dan Teknologi
Universitas Islam Negeri (UIN) Ar-Raniry Banda Aceh
Sebagai Salah Satu Beban Studi Memperoleh Gelar Sarjana (S1)
Dalam Ilmu Teknologi Informasi

Oleh:
Feril Ferdian
220705062

Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi

Disetujui Untuk Diseminarkan Oleh:

Pembimbing I, AR - RANIRY


Mulkan Fadhli, S.T., M.T.
NIP. 19881282020121006

Pembimbing II,


Malahayati, M.T.
NIP. 198301272015032003

Mengetahui,
Ketua Program Studi Teknologi Informasi


Malahayati, M.T.
NIP. 198301272015032003

LEMBAR PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI BERBASIS WEB UNTUK MENDUKUNG PENGELOLAAN AKADEMIK DAN ADMINISTRASI PADA MA'HAD AL-JAMI'AH UIN AR-RANIRY BANDA ACEH

TUGAS AKHIR

Telah Diuji oleh Panitia Ujian Munaqasyah Tugas Akhir
Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh dan Dinyatakan Lulus
Serta Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S1)
Dalam Program Studi Teknologi Informasi

Pada Hari/Tanggal: Rabu, 26 Agustus 2026 M

13 Rabi'ul Awal 1448 H

Di Darussalam, Banda Aceh

Panitia Ujian Munaqasyah Tugas Akhir:

Ketua,

Sekretaris,

Mulkan Fadhli, S.T., M.T.

NIP. 198811282020121006

Malahayati, M.T.

NIP. 198301272015032003

Penguji I,

Penguji II,

Dr. Hendri Ahmadian, M.I.M.

NIP. 198301042014031002

Aulia Syarif Aziz, S.Kom., M.Sc.

NIP. 199305212022031001

Mengetahui:

Dekan Fakultas Sains dan Teknologi
UIN Ar-Raniry Banda Aceh



Prof. Dr. Ir. Muhammad Dirhamsyah, M.T., IPU

NIP. 196210021988111001

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Feril Ferdian
NIM : 220705062
Program Studi : Teknologi Informasi
Judul : PENGEMBANGAN SISTEM INFORMASI
BERBASIS WEB UNTUK Mendukung
PENGELOLAAN AKADEMIK DAN
ADMINISTRASI PADA MA'HAD AL-JAMI'AH
UIN AR-RANIRY Banda Aceh

Dengan ini menyatakan bahwa dalam penulisan tugas akhir ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah karya orang lain;
3. Tidak menggunakan karya orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik karya;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri karya ini dan mampu bertanggungjawab atas karya ini.

Bila dikemudian hari ada tuntutan dari pihak lain atas karya saya, dan telah melalui pembuktian yang dapat dipertanggungjawabkan dan ternyata memang ditemukan bukti bahwa saya telah melanggar pernyataan ini, maka saya siap dikenai sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.

Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh, 21 Agustus 2026

Yang Menyatakan,



Feril Ferdian

ABSTRAK

Pengelolaan akademik dan administrasi Ma'had Al-Jami'ah UIN Ar-Raniry masih menghadapi kendala berupa pemetaan kelas (*uniting*) secara manual, belum tersedianya portal terpusat bagi ustadz/ustadzah untuk mengelola presensi dan nilai, serta keterbatasan layanan pengecekan kelulusan dan pengunduhan sertifikat. Penelitian ini bertujuan merancang dan mengembangkan Sistem Informasi Operasional Ma'had berbasis web, menguji kesesuaian fungsi utama, dan mengevaluasi kebergunaannya berdasarkan penilaian pengguna. Pengembangan sistem menggunakan metode Waterfall yang meliputi analisis kebutuhan, perancangan, implementasi, dan pengujian. Data kebutuhan diperoleh melalui studi literatur, observasi, dan wawancara. Sistem dikembangkan dengan arsitektur *client-server* menggunakan Next.js, NestJS, PostgreSQL, autentikasi berbasis peran, serta integrasi REST API secara *read-only* dengan Sistem Ma'had lama. Sistem mencakup dashboard koordinator untuk sinkronisasi data dan *uniting* kelas, dashboard ustadz/ustadzah untuk mengelola presensi dan nilai, serta layanan publik berbasis NIM untuk mengecek status kelulusan dan mengunduh sertifikat. Pengujian fungsional dilakukan menggunakan *Black-Box Testing* terhadap sembilan skenario fungsi vital. Evaluasi kebergunaan menggunakan *System Usability Scale* (SUS) melibatkan 31 responden, terdiri atas 2 koordinator, 9 ustadz/ustadzah, dan 20 mahasiswa. Hasil pengujian menunjukkan seluruh sembilan skenario berstatus lulus dengan tingkat kesesuaian fungsi utama sebesar 100%. Rata-rata skor SUS adalah 72,50 untuk koordinator, 74,44 untuk ustadz/ustadzah, dan 88,88 untuk mahasiswa, seluruh kelompok berada dalam kategori *Acceptable*. Hasil tersebut menunjukkan bahwa sistem memenuhi fungsi utama yang diuji dan dapat diterima dari aspek kebergunaan dalam konteks responden penelitian. Hasil penelitian bersifat deskriptif dan tidak digunakan untuk menyatakan peningkatan efisiensi administratif secara kuantitatif.

Kata kunci: Sistem Informasi Ma'had, Waterfall, *Black-Box Testing*, *System Usability Scale*, REST API.

ABSTRACT

The academic and administrative management of Ma'had Al-Jami'ah at UIN Ar-Raniry continues to face several challenges, including manual class allocation (*uniting*), the absence of a centralized portal for instructors to manage attendance and grades, and limitations in graduation-status checking and certificate download services. This study aims to design and develop a web-based Ma'had Operational Information System, test the conformity of its main functions, and evaluate its usability based on user assessments. The system was developed using the Waterfall method, comprising requirements analysis, design, implementation, and testing. Requirements data were collected through a literature review, observation, and interviews. The system adopts a client-server architecture using Next.js, NestJS, PostgreSQL, role-based authentication, and read-only REST API integration with the legacy Ma'had System. It comprises a coordinator dashboard for data synchronization and class allocation, an instructor dashboard for attendance and grade management, and a student-facing public service for checking graduation status and downloading certificates based on student identification numbers. Functional testing was conducted using Black-Box Testing across nine vital functional scenarios. Usability evaluation using the System Usability Scale (SUS) involved 31 respondents, consisting of 2 coordinators, 9 instructors, and 20 Ma'had students. The results showed that all nine scenarios passed, producing a 100% conformity rate for the main functions. The average SUS scores were 72.50 for coordinators, 74.44 for instructors, and 88.88 for Ma'had students, with all groups classified as *Acceptable*. These results indicate that the system fulfilled the tested primary functions and was acceptable in terms of usability within the context of the participating respondents. The findings are descriptive and are not intended to demonstrate a quantitative improvement in administrative efficiency.

Keywords: Ma'had Information System, Waterfall, Black-Box Testing, System Usability Scale, REST API.

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan ke hadirat Allah Swt. atas limpahan rahmat, nikmat, dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Pengembangan Sistem Informasi Berbasis Web untuk Mendukung Pengelolaan Akademik dan Administrasi pada Ma’had Al-Jami’ah UIN Ar-Raniry Banda Aceh”. Selawat dan salam senantiasa tercurahkan kepada Nabi Muhammad saw., beserta keluarga, sahabat, dan seluruh umatnya yang istiqamah mengikuti ajaran beliau.

Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknologi Informasi, Fakultas Sains dan Teknologi, Universitas Islam Negeri Ar-Raniry Banda Aceh. Penelitian ini berfokus pada pengembangan sistem informasi operasional berbasis web yang mencakup dashboard koordinator untuk mendukung pengelolaan data dan pemetaan kelas (*uniting*), dashboard ustadz/ustadzah untuk pengelolaan presensi dan nilai, serta layanan publik bagi mahasantri untuk melakukan pengecekan status kelulusan dan mengunduh sertifikat berdasarkan Nomor Induk Mahasiswa (NIM).

Dalam proses penyusunan skripsi dan pengembangan sistem, penulis memperoleh banyak bantuan, arahan, dukungan, serta doa dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati, penulis menyampaikan terima kasih kepada:

1. Ibu Malahayati, M.T., selaku Ketua Program Studi Teknologi Informasi Fakultas Sains dan Teknologi Universitas Islam Negeri Ar-Raniry Banda Aceh, sekaligus Pembimbing II dan juga pembimbing akademik yang telah memberikan arahan kepada penulis selama menjalani perkuliahan.
2. Bapak Mulkan Fadhli, S.T., M.T, selaku Pembimbing I, yang telah meluangkan waktu untuk memberikan bimbingan, arahan, saran, dan masukan selama pelaksanaan penelitian dan penyusunan skripsi ini.
3. Seluruh dosen dan staf Program Studi Teknologi Informasi Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh yang telah memberikan ilmu, pengalaman, dan pelayanan akademik kepada penulis selama masa perkuliahan.

4. Pimpinan dan seluruh pihak Ma'had Al-Jami'ah UIN Ar-Raniry Banda Aceh, khususnya koordinator serta ustadz/ustadzah, yang telah memberikan izin, informasi, masukan, dan kesempatan kepada penulis untuk melaksanakan penelitian dan pengujian sistem.
5. Para mahasiswa yang telah bersedia berpartisipasi sebagai responden dalam evaluasi sistem yang dikembangkan.
6. Kedua orang tua tercinta, almarhum Bapak Jufri dan Ibu Ariwani, serta seluruh keluarga yang senantiasa memberikan doa, kasih sayang, dukungan, pengorbanan, dan motivasi kepada penulis.
7. Sahabat, teman dan rekan-rekan mahasiswa Program Studi Teknologi Informasi, khususnya angkatan 2022, yang telah memberikan dukungan, bantuan, dan semangat selama proses perkuliahan dan penyusunan skripsi.
8. Seluruh pihak yang tidak dapat disebutkan satu per satu yang telah membantu penulis, baik secara langsung maupun tidak langsung, dalam menyelesaikan penelitian dan skripsi ini.

Penulis menyadari bahwa skripsi ini masih memiliki keterbatasan dan belum sepenuhnya sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun sebagai bahan perbaikan dan pengembangan pada masa mendatang. Penulis berharap hasil penelitian ini dapat memberikan manfaat bagi Ma'had Al-Jami'ah UIN Ar-Raniry Banda Aceh, khususnya dalam mendukung pengelolaan kegiatan akademik dan administratif, serta menjadi referensi bagi penelitian dan pengembangan sistem informasi selanjutnya.

Akhir kata, penulis memohon kepada Allah Swt. agar segala bantuan, dukungan, dan kebaikan yang telah diberikan oleh seluruh pihak mendapatkan balasan yang berlipat ganda. Semoga skripsi ini dapat memberikan manfaat bagi pembaca dan pihak-pihak yang membutuhkan

Banda Aceh, 21 Agustus 2026

Penulis



Feril Ferdian

DAFTAR ISI

ABSTRAK	i
ABSTRACT	ii
KATA PENGANTAR.....	iii
DAFTAR ISI.....	v
DAFTAR GAMBAR	viii
DAFTAR TABEL	x
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian.....	3
1.4 Manfaat Penelitian.....	4
1.5 Batasan Masalah.....	5
BAB 2 LANDASAN TEORI	6
2.1 Sistem Informasi.....	6
2.2 Sistem Informasi Berbasis Web.....	7
2.3 Ma'had Al-Jami'ah	8
2.4 Konsep Pengelolaan Akademik dan Administrasi.....	9
2.5 Konsep Pemetaan Kelas (Uniting)	10
2.6 Sistem Absensi dan Pengelolaan Nilai	11
2.7 Self-Service Technology dalam Layanan Akademik.....	12
2.8 Integrasi Sistem dan REST API.....	12
2.9 Arsitektur Sistem Berbasis Peran (Role-Based Access Control / RBAC)...	14
2.10 Teknologi Pengembangan Sistem (Tech Stack)	15
2.10.1 Framework Frontend (Next.js)	15
2.10.2 Framework Backend (Nest.js)	16
2.10.3 Sistem Manajemen Basis Data (PostgreSQL)	16

2.11 Metodologi Pengembangan dan Pengujian Sistem	17
2.12 Penelitian Terdahulu	18
BAB 3 METODOLOGI PENELITIAN	21
3.1 Tahapan Penelitian.....	21
3.1.1 Metode Pengumpulan Data.....	23
3.1.2 Metode Pengembangan Sistem.....	25
3.1.3 Tahap Analisis Kebutuhan Sistem	25
3.1.4 Tahap Perancangan Sistem	29
3.1.5 Tahap Implementasi Sistem.....	52
3.1.6 Tahap Pengujian Sistem.....	57
3.2 Populasi dan Sampel	60
3.3 Instrumen Penelitian.....	62
3.3.1 Pedoman Observasi dan Wawancara	62
3.3.2 Lembar Pengujian Fungsional.....	63
3.3.3 Skenario Penggunaan dan Kuesioner SUS	64
3.4 Teknik Analisis Data.....	66
3.4.1 Analisis Hasil Black-Box Testing.....	67
3.4.2 Analisis System Usability Scale	67
BAB 4 HASIL DAN PEMBAHASAN	70
4.1 Gambaran Umum Hasil Pengembangan	70
4.1.1 Arsitektur Implementasi Sistem.....	71
4.1.2 Struktur Direktori Implementasi.....	73
4.1.3 Batas Implementasi Penelitian.....	74
4.2 Hasil Implementasi Sistem.....	74
4.2.1 Implementasi Autentikasi dan Otorisasi	74
4.2.2 Implementasi Dashboard Koordinator.....	76

4.2.3 Implementasi Dashboard Ustadz	81
4.2.4 Implementasi Layanan Publik Kelulusan dan Sertifikat.....	84
4.2.5 Implementasi Basis Data dan Integrasi.....	87
4.3 Hasil Pengujian Fungsional Menggunakan Black-Box Testing	88
4.3.1 Lingkungan dan Ketentuan Pengujian.....	89
4.3.2 Hasil Pengujian Modul Koordinator.....	89
4.3.3 Hasil Pengujian Modul Ustadz	92
4.3.4 Hasil Pengujian Layanan Publik Mahasantri.....	93
4.3.5 Rekapitulasi Hasil Pengujian Fungsional	93
4.4 Hasil Evaluasi Kebergunaan Menggunakan System Usability Scale.....	94
4.4.1 Karakteristik dan Realisasi Responden.....	95
4.4.2 Hasil Perhitungan Skor System Usability Scale (SUS).....	95
4.5 Pembahasan	100
4.5.1 Kesesuaian Implementasi dengan Kebutuhan Sistem	100
4.5.2 Pembahasan terhadap Proses Operasional	101
4.5.3 Pembahasan Integrasi dan Keamanan.....	102
4.5.4 Keterbatasan Hasil Pengembangan.....	103
4.5.5 Pembahasan terhadap Rumusan Masalah	103
BAB 5 KESIMPULAN DAN SARAN	105
5.1 Kesimpulan.....	105
5.2 Saran	105
DAFTAR PUSTAKA.....	107
LAMPIRAN.....	111

DAFTAR GAMBAR

Gambar 2.1 Konsep REST API.....	13
Gambar 3.1 Diagram Alur Penelitian.....	21
Gambar 3. 2 Diagram Pengembangan Sistem dengan Metode Waterfall	25
Gambar 3.3 Arsitektur Sistem	30
Gambar 3.4 Use Case Diagram Sistem	32
Gambar 3.5 Activity Diagram Sinkronasi Data Mahasantri	33
Gambar 3.6 Activity Diagram Uniting.....	34
Gambar 3.7 Activity Diagram Absensi & Nilai	35
Gambar 3.8 Activity Diagram Validasi Sertifikat	36
Gambar 3. 9 Sequence Diagram Sinkronisasi Data Mahasantri	37
Gambar 3. 10 Sequence Diagram Distribusi Kelas.....	38
Gambar 3. 11 Sequence Diagram Pengelolaan Data Akademik	39
Gambar 3. 12 Sequence Diagram Validasi Kelulusan.....	40
Gambar 3.13 Entity Relationship Diagram.....	43
Gambar 3.14 Halaman Login – Wireframe.....	44
Gambar 3.15 Halaman Dashboard Koordinator – Wireframe	46
Gambar 3.16 Halaman Uniting Mahasantri – Wireframe	47
Gambar 3.17 Halaman Dashboard Ustadz – Wireframe.....	49
Gambar 3.18 Halaman Pengelolaan Absensi & Nilai – Wireframe.....	50
Gambar 3.19 Halaman Validasi Sertifikat (Publik) – Wireframe.....	52
Gambar 4.1 Arsitektur Implementasi Sistem Informasi Operasional Ma’had	72
Gambar 4.2 Struktur Direktori Modul Sistem dalam Lingkup Penelitian	73
Gambar 4.3 Halaman Login Koordinator dan Ustadz	76
Gambar 4.4 Dashboard Ringkasan Koordinator	77
Gambar 4.5 Halaman Data Mahasantri dan Ringkasan Sinkronisasi	78
Gambar 4.6 Implementasi Sinkronisasi dan Upsert Data Mahasantri	78
Gambar 4.7 Halaman Kelas	80
Gambar 4.8 Implementasi Algoritma Auto-Uniting	80
Gambar 4.9 Rekap Distribusi dan Statistik Kelas.....	81
Gambar 4.10 Dashboard dan Daftar Kelas Ustadz	82

Gambar 4.11 Formulir Presensi Mahasantri per Pertemuan	83
Gambar 4.12 Formulir Pengisian dan Rekap Nilai Mahasantri	83
Gambar 4.13 Implementasi Validasi dan Perhitungan Nilai Akhir	84
Gambar 4.14 Halaman Pengecekan Kelulusan Berdasarkan NIM	85
Gambar 4.15 Hasil Kelulusan dan Tombol Unduh Sertifikat	86
Gambar 4.16 Implementasi Pengambilan Data Kelulusan dan Sertifikat.....	87
Gambar 4.17 Tampilan Implementasi Skema Basis Data	88



DAFTAR TABEL

Tabel 2.1 Perbandingan Penelitian Terdahulu dengan Penelitian Saat Ini	20
Tabel 3.1 Daftar Kebutuhan Fungsional Sistem Berdasarkan Aktor	26
Tabel 3.2 Spesifikasi Kebutuhan Non-Fungsional Sistem	27
Tabel 3.3 Kebutuhan Perangkat Keras	28
Tabel 3.4 Kebutuhan Perangkat Lunak dan Alat.....	28
Tabel 3.5 Entitas Periode Akademik	40
Tabel 3.6 Entitas Mata Pelajaran.....	41
Tabel 3.7 Entitas Lokasi Gedung	41
Tabel 3.8 Entitas Ustadz.....	41
Tabel 3.9 Entitas Kelas Unit.....	41
Tabel 3.10 Entitas Mahasantri.....	42
Tabel 3.11 Entitas Peserta Kelas	42
Tabel 3.12 Entitas Presensi	42
Tabel 3.13 Entitas Nilai.....	42
Tabel 3.14 Skenario Pengujian Fungsional Menggunakan Black-Box Testing	63
Tabel 3.15 Skenario Penggunaan Koordinator.....	64
Tabel 3.16 Skenario Penggunaan Ustadz.....	65
Tabel 3.17 Skenario Penggunaan Mahasantri	65
Tabel 3.18 Instrumen System Usability Scale.....	66
Tabel 3.19 Interpretasi Acceptability Ranges Skor SUS.....	69
Tabel 4.1 Ruang Lingkup Hasil Implementasi.....	71
Tabel 4.2 Teknologi Utama pada Implementasi Sistem	72
Tabel 4.3 Identitas Pelaksanaan Black-Box Testing	89
Tabel 4.4 Hasil Black-Box Testing Modul Koordinator	89
Tabel 4.5 Hasil Black-Box Testing Modul Ustadz.....	92
Tabel 4.6 Hasil Black-Box Testing Layanan Publik Mahasantri	93
Tabel 4.7 Rekapitulasi Black-Box Testing	93
Tabel 4.8 Rencana dan Realisasi Responden SUS.....	95
Tabel 4.9 Distribusi Skor Kontribusi Jawaban Responden Koordinator	96
Tabel 4.10 Distribusi Skor Kontribusi Jawaban Responden Ustadz/Ustadzah.....	96

Tabel 4.11 Distribusi Skor Kontribusi Jawaban Responden Mahasantri	98
Tabel 4.12 Rekapitulasi Hasil SUS Berdasarkan Kelompok Pengguna	99
Tabel 4.13 Keterkaitan Kebutuhan dan Hasil Implementasi.....	101



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Transformasi digital melalui implementasi Sistem Informasi Akademik (SIKAD) merupakan langkah esensial bagi institusi pendidikan tinggi untuk menciptakan tata kelola yang efisien, transparan, dan akuntabel. Penggunaan sistem informasi terintegrasi terbukti mampu menggantikan proses manual, mengurangi *human error*, serta meningkatkan kualitas layanan (Qo'du et al., 2025). Kebutuhan akan tata kelola yang sistematis ini tidak hanya berlaku pada perguruan tinggi secara umum, tetapi juga sangat krusial bagi institusi pendidikan yang memiliki struktur lebih spesifik.

Dalam ekosistem Perguruan Tinggi Keagamaan Islam Negeri (PTKIN), tata kelola pendidikan ini menjadi lebih kompleks dengan hadirnya lembaga Ma'had Al-Jami'ah. Sebagai entitas yang memadukan sistem pendidikan tinggi dan tradisi pesantren, pengelola Ma'had dituntut untuk menyinkronkan administrasi akademik dengan pembinaan karakter secara simultan, karena pengelolaannya berada di bawah koordinasi langsung birokrasi universitas (Rahman et al., 2023). Kompleksitas operasional inilah yang menjadikan infrastruktur teknologi informasi yang andal sebagai sebuah keharusan agar seluruh proses dapat berjalan efektif.

Berpijak pada urgensi tersebut, Ma'had Al-Jami'ah UIN Ar-Raniry Banda Aceh sejatinya telah memiliki sistem informasinya sendiri, namun pelaksanaannya di lapangan belum optimal. Berdasarkan wawancara dengan pihak pengelola, teridentifikasi sejumlah kendala administratif akibat belum terintegrasinya fungsionalitas sistem. Salah satu kendala utama pada tingkat manajerial adalah proses *uniting* (pemetaan kelas mahasiswa) yang masih dikerjakan secara manual menggunakan *spreadsheet*. Padahal, pengelolaan data akademik berskala besar yang masih mengandalkan perangkat lunak *spreadsheet* konvensional sangat rentan terhadap inkonsistensi data, redundansi, dan rentan terhadap kehilangan riwayat pembaruan (Kustitskaya et al., 2023). Ketiadaan *dashboard* khusus bagi koordinator menyebabkan pemantauan distribusi kelas menjadi lambat dan rentan

terhadap inkonsistensi data. Di samping itu, kendala juga dirasakan oleh tenaga pengajar (ustadz) yang belum memiliki akses portal terpusat untuk mencatat absensi dan merekapitulasi nilai secara langsung ke dalam sistem.

Disisi lain, permasalahan juga ditemukan pada aspek layanan mahasiswa. Alur pengunduhan sertifikat kelulusan saat ini dinilai kurang efisien karena mahasiswa kerap mengalami kendala akses. Beberapa kendala tersebut meliputi kegagalan pencetakan sertifikat akibat ketidaksesuaian pembacaan status pembayaran pada sistem lama, hingga kesulitan mengingat kata sandi akun kampus. Kondisi ini memaksa mahasiswa untuk mengurus pencetakan dokumen secara luring ke kantor operasional, yang pada akhirnya justru menambah beban kerja administratif staf Ma'had.

Upaya untuk mengembangkan modul sistem baru guna mengatasi berbagai permasalahan operasional tersebut kemudian memunculkan tantangan baru terkait integrasi data. Sistem yang akan dikembangkan dirancang terpisah dari sistem informasi utama Ma'had Al-Jami'ah. Keterpisahan ini memicu kendala pertukaran data, mengingat data induk mahasiswa masih tersimpan di sistem yang lama. Dalam arsitektur basis data, aplikasi yang berjalan sendiri-sendiri dengan basis data yang terpisah berisiko tinggi memunculkan ketidaksinkronan dan redundansi data antar-sistem (Koripalli, 2025). Oleh karena itu, Tanpa adanya mekanisme sinkronisasi yang tepat, staf koordinator akan terbebani dengan keharusan menginput ulang ribuan identitas mahasiswa secara manual ke dalam sistem baru sebelum proses pemetaan kelas dapat dieksekusi.

Sebagai solusi komprehensif atas permasalahan manajerial, layanan, dan integrasi data tersebut, penelitian ini bertujuan merancang dan mengembangkan Sistem Informasi Ma'had Berbasis Web yang terintegrasi. Sistem ini akan menghadirkan *dashboard* koordinator untuk memfasilitasi otomasi proses pemetaan kelas (*uniting*), yang didukung oleh fitur sinkronisasi data antar-sistem secara otomatis guna menarik data mahasiswa dari sistem lama tanpa perlu entri manual. Selanjutnya, sistem ini menyediakan *dashboard* khusus bagi ustadz untuk mengelola nilai dan absensi dengan pembagian hak akses pengguna yang terstruktur. Untuk mengatasi kendala layanan, sistem ini menyediakan portal publik

mandiri berbasis pencarian Nomor Induk Mahasiswa (NIM). Melalui portal ini, mahasiswa dapat langsung mengecek status dan mengunduh sertifikat kelulusan tanpa perlu melalui tahapan *login* yang rumit, sehingga diharapkan mampu mewujudkan tata kelola Ma'had yang efisien, terintegrasi, dan responsif.

1.2 Rumusan Masalah

Mengacu pada rumusan masalah yang telah ditetapkan, maka tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

1. Bagaimana merancang dan mengembangkan Sistem Informasi Operasional Ma'had berbasis web yang mendukung pengelolaan pemetaan kelas (*uniting*) oleh koordinator, pengelolaan presensi dan nilai oleh ustadz, serta layanan pengecekan status kelulusan dan pengunduhan sertifikat berdasarkan Nomor Induk Mahasiswa (NIM)?
2. Bagaimana hasil pengujian fungsional Sistem Informasi Operasional Ma'had menggunakan metode *Black-Box Testing* berdasarkan kesesuaian fungsi sistem dengan spesifikasi kebutuhan yang telah ditetapkan?
3. Bagaimana hasil evaluasi tingkat kebergunaan (*usability*) Sistem Informasi Operasional Ma'had menggunakan *System Usability Scale* (SUS) berdasarkan penilaian pengguna?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah ditetapkan, maka tujuan yang ingin dicapai dalam penelitian ini adalah:

1. Merancang dan mengembangkan Sistem Informasi Operasional Ma'had berbasis web yang mendukung pengelolaan pemetaan kelas (*uniting*) oleh koordinator, pengelolaan presensi dan nilai oleh ustadz, serta layanan pengecekan status kelulusan dan pengunduhan sertifikat berdasarkan Nomor Induk Mahasiswa (NIM).
2. Menguji fungsionalitas Sistem Informasi Operasional Ma'had menggunakan metode *Black-Box Testing* untuk mengetahui kesesuaian fungsi sistem dengan spesifikasi kebutuhan yang telah ditetapkan.

3. Mengevaluasi tingkat kebergunaan (*usability*) Sistem Informasi Operasional Ma'had menggunakan *System Usability Scale* (SUS) berdasarkan penilaian pengguna.

1.4 Manfaat Penelitian

Dengan adanya penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

A. Manfaat Praktis

Penelitian ini memberikan manfaat praktis bagi operasional Ma'had Al-Jami'ah UIN Ar-Raniry berupa implementasi sistem yang mampu mengotomatisasi proses pemetaan kelas (*uniting*), menyentralisasi rekapitulasi nilai dan absensi oleh tenaga pengajar, serta menyediakan portal layanan mandiri bagi mahasiswa. Penerapan sistem ini diharapkan dapat memangkas birokrasi layanan, mempercepat alur kerja administrasi, dan meminimalkan kesalahan pencatatan yang selama ini terjadi akibat penggunaan sistem manual.

B. Manfaat Teoritis

Penelitian ini memberikan kontribusi teoretis dalam penerapan konsep interoperabilitas sistem informasi dan evaluasi *usability* pada tata kelola akademik institusi pendidikan berasrama. Hasil penelitian ini diharapkan dapat memperkuat pemahaman mengenai pentingnya sinkronisasi data yang terdistribusi dan pemanfaatan *Self-Service Technology* sebagai bagian dari tata kelola administrasi yang efektif dan transparan.

C. Manfaat Akademik

Secara akademik, penelitian ini diharapkan dapat menambah referensi dan kajian ilmiah di bidang Sistem Informasi dan Rekayasa Perangkat Lunak, khususnya terkait pengembangan aplikasi manajemen operasional Ma'had berbasis web. Penelitian ini dapat menjadi contoh implementasi penyelesaian masalah integrasi basis data terpisah pada konteks Perguruan Tinggi Keagamaan Islam Negeri (PTKIN) yang memiliki kompleksitas manajerial spesifik.

D. Manfaat bagi Pengembang dan Peneliti Selanjutnya

Penelitian ini dapat dijadikan sebagai acuan bagi peneliti selanjutnya yang ingin mengembangkan aplikasi tata kelola akademik serupa. Rancangan arsitektur,

alur kerja sinkronisasi, serta hasil evaluasi tingkat penerimaan pengguna (*usability*) yang dihasilkan dalam penelitian ini diharapkan dapat menjadi fondasi dasar untuk pengembangan sistem yang lebih komprehensif di masa mendatang.

1.5 Batasan Masalah

Agar ruang lingkup penelitian ini lebih terarah, fokus, dan tidak menyimpang dari tujuan yang diharapkan, maka ditetapkan batasan masalah sebagai berikut:

1. Pengembangan fungsionalitas sistem dibatasi pada tiga modul utama, yaitu *dashboard* koordinator (dengan fitur otomasi *uniting* dan sinkronisasi data), *dashboard* ustadz (untuk pengelolaan absensi dan nilai akademik), serta portal layanan publik mandiri (*self-service*) mahasiswa berbasis pencarian Nomor Induk Mahasiswa (NIM).
2. Arsitektur sistem dikembangkan sepenuhnya sebagai aplikasi berbasis web, dan tidak mencakup pengembangan aplikasi *native* untuk perangkat *mobile* (Android/iOS).
3. Pendekatan integrasi untuk mengatasi keterpisahan basis data dibatasi pada penarikan data (*data pulling*) identitas mahasiswa menggunakan antarmuka REST API. Sistem operasional baru ini hanya bersifat membaca data (*read-only*) dan tidak memiliki hak akses untuk mengubah atau memperbarui data master sensitif pada Sistem Informasi Ma'had Al-Jami'ah utama UIN Ar-Raniry.
4. Evaluasi keberhasilan sistem dibatasi pada dua pengujian utama, yakni pengujian fungsionalitas untuk memastikan seluruh modul berjalan sesuai spesifikasi (*Black-Box Testing*), serta evaluasi tingkat penerimaan dan kemudahan penggunaan aplikasi oleh pengguna (*Usability Testing*).

BAB 2

LANDASAN TEORI

2.1 Sistem Informasi

Sistem informasi merupakan kesatuan elemen yang saling berinteraksi secara sistematis untuk mengumpulkan, memproses, menyimpan, dan mendistribusikan data guna mendukung kegiatan manajerial dan operasional suatu organisasi. Integrasi antara teknologi informasi dan aktivitas pengguna ini sangat krusial dalam mengubah data mentah menjadi informasi yang bermakna bagi proses pengambilan keputusan. Menurut Inayah dkk. (2025), keberadaan sistem informasi yang dikelola dengan baik memungkinkan institusi untuk menjalankan operasinya secara lebih strategis, terstruktur, dan terukur, sekaligus menggantikan pola kerja manual yang lambat.

Dalam implementasinya, sebuah sistem informasi yang tangguh tidak hanya bergantung pada kecanggihan perangkat keras dan perangkat lunak, tetapi juga sangat mengandalkan manajemen basis data (*database*) yang terpusat serta prosedur kerja yang jelas. Anggraeni (2017) menegaskan bahwa sistem informasi terkomputerisasi yang dirancang secara optimal mampu meminimalkan redundansi data dan mempercepat proses penarikan informasi (*information retrieval*). Pendekatan ini sangat penting terutama bagi institusi yang menangani volume data berskala besar, dimana pencatatan secara konvensional sering kali memicu inkonsistensi, duplikasi, dan rentan terhadap kesalahan manusia (*human error*).

Penerapan sistem informasi memberikan dampak langsung terhadap efisiensi birokrasi dan peningkatan kualitas layanan administratif. Di era transformasi digital saat ini, organisasi pendidikan maupun instansi birokrasi dituntut untuk mengadopsi pengelolaan data yang lebih transparan dan akuntabel. Sutabri (2012) menyatakan bahwa digitalisasi melalui implementasi sistem informasi terpadu tidak hanya memangkas waktu operasional dan biaya, tetapi juga memfasilitasi kemudahan integrasi antar-divisi, sehingga alur kerja organisasi menjadi lebih lancar dan pelayanan kepada pemangku kepentingan dapat berjalan dengan maksimal.

2.2 Sistem Informasi Berbasis Web

Sistem informasi berbasis web merupakan evolusi dari perangkat lunak konvensional yang mengandalkan teknologi jaringan internet atau intranet untuk memproses dan mendistribusikan data. Secara arsitektural, sistem ini dibangun di atas konsep *client-server*, di mana peramban web (*web browser*) pada perangkat pengguna bertindak sebagai klien yang meminta layanan melalui protokol HTTP/HTTPS, sedangkan *server* merespons dengan memproses logika bisnis dan menyajikan antarmuka halaman. Rahmi et al. (2023) menjelaskan bahwa pengembangan sistem informasi dengan pendekatan *web-based* saat ini telah menjadi standar industri utama karena kemampuannya dalam menyediakan layanan yang selalu aktif (*always-on*) dan dapat diakses melampaui batasan ruang fisik komputer lokal.

Keunggulan paling fundamental dari aplikasi berbasis web terletak pada tingkat aksesibilitasnya yang fleksibel dan sifatnya yang lintas platform (*cross-platform*). Pengguna tidak diwajibkan untuk mengunduh, menginstal, maupun melakukan pembaruan aplikasi secara manual di perangkat masing-masing, mengingat seluruh konfigurasi dan pembaharuan sistem dieksekusi secara terpusat di sisi *server*. Nurdin et al. (2024) menegaskan bahwa sentralisasi pada arsitektur web tidak hanya menekan biaya pemeliharaan (*maintenance*) bagi pihak administrator, tetapi juga menjamin bahwa seluruh pengguna akan selalu menerima keluaran data dan versi antarmuka yang identik secara seketika (*real-time*), sehingga sangat efektif dalam meminimalisir risiko asimetri informasi.

Dalam ekosistem tata kelola akademik dan birokrasi institusi, kehadiran sistem informasi berbasis web menjadi solusi paling rasional untuk memecahkan masalah fragmentasi atau keterpisahan data (Eka et al., 2025). Ekosistem ini mampu memfasilitasi berbagai entitas yang berbeda peran mulai dari koordinator akademik, tenaga pengajar, hingga mahasiswa untuk berinteraksi dengan satu pangkalan data induk yang sama melalui antarmuka portal yang terintegrasi. Mobilitas operasional juga menjadi jauh lebih ringkas, karena staf dapat menyelesaikan pemetaan kelas maupun rekapitulasi nilai hanya dengan

bermodalkan peramban standar dari berbagai perangkat, tanpa harus terikat pada fasilitas komputer fisik di ruang tata usaha.

2.3 Ma'had Al-Jami'ah

Ma'had Al-Jami'ah merupakan entitas kelembagaan strategis yang bernaung di bawah Perguruan Tinggi Keagamaan Islam Negeri (PTKIN) di Indonesia, seperti halnya Universitas Islam Negeri (UIN) (Nurfajrina, 2023). Secara konseptual, Ma'had Al-Jami'ah dirancang sebagai wadah integrasi yang memadukan sistem pendidikan tinggi modern dengan tradisi keilmuan pesantren klasik. Nurfajrina (2023) menjelaskan bahwa pendirian Ma'had di lingkungan kampus bertujuan untuk membentuk landasan *tafaqquh fiddin* (pemahaman agama yang mendalam) bagi mahasiswa, sekaligus menjadi pusat pembinaan karakter keislaman yang tidak dapat sepenuhnya diakomodasi hanya melalui kurikulum reguler di dalam ruang kelas fakultas.

Secara operasional, Ma'had Al-Jami'ah berfungsi jauh lebih kompleks daripada sekadar fasilitas asrama mahasiswa. Institusi ini menyelenggarakan program akademik internal yang intensif dan wajib diikuti oleh mahasantri (mahasiswa yang tinggal di Ma'had), seperti pembinaan bahasa asing (Arab dan Inggris), *tahsin/tahfidz* Al-Qur'an, serta kajian kitab kuning (*dirosah islamiyah*). Ismail (2022) menegaskan bahwa eksistensi Ma'had sebagai pusat transformasi nilai dan ibadah menuntut adanya tata kelola manajemen yang sangat terstruktur, mengingat pengurus harus memantau rekam jejak kedisiplinan dan capaian akademik dari ribuan mahasiswa baru yang masuk pada setiap tahun ajaran.

Skala operasional yang masif ini pada akhirnya melahirkan tantangan administratif yang signifikan bagi pihak pengelola. Setiap awal semester, koordinator Ma'had dihadapkan pada rutinitas manajerial yang rumit, mulai dari distribusi mahasantri ke dalam kamar, pemetaan kelompok belajar atau *halaqah* (*uniting*), hingga rekapitulasi presensi dan nilai dari ratusan tenaga pengajar (*ustadz/ustadzah*). Fadhli (2025) menyoroti bahwa tingginya intensitas kegiatan komunal dalam metode *halaqah* di Ma'had sering kali memicu kendala birokrasi jika tidak didukung oleh sistem pencatatan yang mumpuni. Kompleksitas inilah yang mendasari urgensi digitalisasi secara mandiri, di mana Ma'had membutuhkan

sistem informasi operasional tersendiri yang mampu mengakomodasi alur kerja lokal mereka, namun tetap terhubung secara presisi dengan pangkalan data mahasiswa di universitas induk.

2.4 Konsep Pengelolaan Akademik dan Administrasi

Pengelolaan akademik dan administrasi merupakan jantung operasional dari setiap institusi pendidikan yang mencakup serangkaian proses perencanaan, pengorganisasian, dan pengawasan terhadap seluruh kegiatan pembelajaran beserta data peserta didik. Secara konseptual, tata kelola ini meliputi berbagai aspek manajerial yang vital, mulai dari registrasi ulang mahasiswa, pemetaan rombongan belajar, pendistribusian jadwal pengajar, hingga perekaman jejak evaluasi akademik. Kristanti dan Putra (2025) dalam penelitiannya menegaskan bahwa efisiensi tata kelola administrasi sangat menentukan seberapa baik sebuah lembaga pendidikan mampu memfasilitasi kebutuhan peserta didiknya, sekaligus memastikan bahwa seluruh ekosistem birokrasi berjalan selaras dengan standar operasional prosedur yang telah ditetapkan.

Dalam praktiknya, pengelolaan administrasi berskala besar yang masih bertumpu pada mekanisme konvensional sering kali memicu berbagai hambatan (*bottleneck*) birokrasi. Proses pencatatan yang menggunakan dokumen fisik atau perangkat lunak lembar kerja (*spreadsheet*) terpisah sangat rentan terhadap duplikasi data, kesalahan rekapitulasi, dan lambatnya proses penelusuran arsip. Rismawati et al. (2024) menyoroti bahwa ketiadaan integrasi sistem akan menyebabkan proses pertukaran informasi antar-divisi berjalan lamban. Hal ini berimbas langsung pada penumpukan beban kerja staf tenaga kependidikan pada masa-masa krusial, seperti saat awal semester ketika proses validasi mahasiswa dan pemetaan kelas harus dieksekusi secara serentak dalam tenggat waktu yang terbatas.

Untuk mengatasi kendala birokrasi tersebut, transformasi digital melalui implementasi sistem informasi manajemen pendidikan berbasis *database* terpusat menjadi sebuah keharusan mutlak. Sistem pengelolaan akademik yang terkomputerisasi tidak hanya mampu mengotomatisasi pengerjaan tugas-tugas administratif yang berulang, tetapi juga memberikan jaminan terhadap konsistensi

dan integritas pangkalan data. Laeliyah (2025) menyimpulkan bahwa sentralisasi aliran data dalam administrasi di pesantren, sekolah, maupun perguruan tinggi terbukti sangat efektif dalam mereduksi tingkat kesalahan manusia (*human error*). Dengan adanya arsitektur sistem yang saling terhubung, institusi dapat memangkas jalur pelaporan sehingga pimpinan maupun koordinator dapat mengambil keputusan berbasis data secara tepat waktu (*real-time*).

2.5 Konsep Pemetaan Kelas (*Uniting*)

Pemetaan kelas, yang dalam tata kelola akademik institusi tertentu lebih dikenal dengan istilah *uniting*, merupakan proses pengelompokan peserta didik ke dalam unit rombongan belajar (rombel) atau *halaqah* berdasarkan kriteria akademik dan administratif. Proses penyusunan ini bertujuan untuk mendistribusikan mahasiswa secara proporsional guna menciptakan rasio kelas yang ideal antara tenaga pengajar dan peserta didik. Wasilah et al. (2024) mengemukakan bahwa pengelolaan data rombongan belajar yang sistematis sangat esensial untuk memastikan pemerataan beban mengajar dan memfasilitasi kelancaran pemantauan aktivitas akademik oleh koordinator pendidikan dari awal hingga akhir semester.

Pada institusi dengan jumlah peserta didik yang masif, proses pemetaan kelas sering kali menjadi fase administratif yang paling membebani jika masih dieksekusi secara manual. Pengelompokan mahasiswa yang sekadar menggunakan fitur penyortiran pada perangkat lunak lembar kerja konvensional sangat berisiko memunculkan ketidakseimbangan kapasitas, redundansi data, dan bentrokan alokasi tenaga pengajar. Fadilah et al. (2022) menjelaskan bahwa kelemahan utama dari pembagian kelas secara manual adalah ketidakmampuannya dalam mengakomodasi variabel data berskala besar secara simultan. Akibatnya, penetapan status kelas mahasiswa di awal semester menjadi sangat lambat dan berpotensi menghambat dimulainya siklus kegiatan belajar mengajar.

Oleh karena itu, otomasi proses *uniting* melalui pengembangan fungsionalitas khusus di dalam sistem informasi operasional menjadi solusi arsitektural untuk mengatasi kerumitan manajerial tersebut. Melalui logika pemrograman yang terpusat, koordinator dapat mendistribusikan ribuan data

mahasiswa ke dalam kelas-kelas yang tersedia dalam hitungan detik tanpa perlu khawatir akan terjadinya tumpang tindih data. Amrulloh dan Sela (2021) menegaskan bahwa optimasi pemetaan kelas dan penjadwalan akademik berbasis komputasi tidak hanya mengeliminasi risiko manipulasi atau duplikasi data, tetapi juga berhasil membebaskan staf birokrasi dari rutinitas administratif yang melelahkan, sehingga mereka dapat berfokus pada peningkatan mutu layanan akademik.

2.6 Sistem Absensi dan Pengelolaan Nilai

Sistem absensi dan pengelolaan nilai merupakan bagian integral dari sistem informasi akademik yang berfungsi untuk mendokumentasikan kehadiran serta hasil evaluasi pembelajaran mahasiswa secara sistematis. Absensi digital memungkinkan pencatatan kehadiran dilakukan secara *real-time* dan tersimpan dalam basis data terpusat, sehingga meminimalkan kesalahan pencatatan manual serta mempermudah proses rekapitulasi. Penelitian oleh Pratisto et al. (2024) menunjukkan bahwa sistem informasi presensi berbasis web yang diuji menggunakan *System Usability Scale* (SUS) memperoleh skor 82,5 yang termasuk kategori “sangat baik”, serta terbukti meningkatkan efisiensi administrasi dan akurasi data dibandingkan metode konvensional berbasis kertas.

Selain presensi, pengelolaan nilai akademik merupakan komponen penting dalam menjamin transparansi dan akuntabilitas proses evaluasi pembelajaran. Sistem pengelolaan nilai berbasis web memungkinkan dosen atau pengajar melakukan input nilai secara langsung ke dalam sistem, yang kemudian dapat diproses secara otomatis untuk menghasilkan nilai akhir atau laporan akademik. Wahab et al. (2023) dalam pengembangan portal mahasiswa berbasis web menemukan bahwa integrasi modul nilai dan presensi dalam satu sistem terpusat meningkatkan kemudahan akses informasi akademik serta mempercepat proses distribusi hasil evaluasi kepada mahasiswa.

Dalam konteks Ma’had Al-Jami’ah, integrasi sistem absensi dan pengelolaan nilai dalam dashboard ustadz menjadi sangat relevan karena proses pembelajaran berlangsung dalam lingkungan terkontrol berbasis asrama. Sistem

yang terintegrasi memungkinkan ustadz mencatat kehadiran dan nilai dalam satu antarmuka yang terdokumentasi secara otomatis, sehingga meminimalkan duplikasi data serta mendukung monitoring akademik secara *real-time* oleh koordinator. Dengan demikian, penerapan sistem absensi dan pengelolaan nilai berbasis web tidak hanya meningkatkan efisiensi operasional, tetapi juga memperkuat tata kelola akademik yang transparan dan terdigitalisasi.

2.7 Self-Service Technology dalam Layanan Akademik

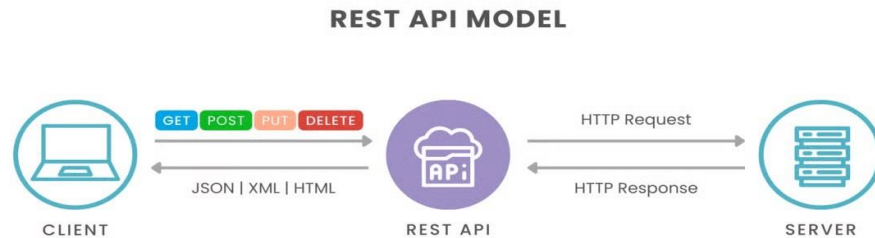
Kemajuan teknologi informasi mendorong pergeseran paradigma layanan publik dari model konvensional yang dilayani oleh staf (*staf-assisted*) menuju model layanan mandiri atau *Self-Service Technology* (SST). Dalam konteks tata kelola institusi pendidikan, SST didefinisikan sebagai antarmuka teknologi yang memungkinkan mahasiswa mengakses layanan informasi administratif secara independen tanpa memerlukan keterlibatan langsung dari pegawai administrasi (Ngcwabe & Chiliya, 2014).

Menurut penelitian Ngcwabe dan Chiliya (2014), Implementasi layanan mandiri dalam sistem akademik yang berjalan efektif terbukti mampu mengurangi beban kerja administratif, meminimalkan keterlambatan distribusi informasi, serta menjadi solusi untuk menghindari masalah antrean fisik. Pada ekosistem Ma'had Al-Jami'ah UIN Ar-Raniry, pemanfaatan konsep *self-service* diwujudkan melalui portal layanan publik pencarian kelulusan. Dengan mengandalkan parameter identitas tunggal berupa Nomor Induk Mahasiswa (NIM), sistem ini didesain untuk menjembatani hambatan teknis seperti hilangnya akses login mahasiswa pada sistem utama. Pendekatan ini tidak hanya menstimulasi efisiensi birokrasi pencetakan sertifikat, tetapi juga meningkatkan aksesibilitas dan kemandirian mahasiswa dalam mengelola dokumen kelulusannya secara cepat dan *real-time*.

2.8 Integrasi Sistem dan REST API

Integrasi sistem merupakan proses menghubungkan berbagai sub-sistem yang berbeda agar dapat bekerja secara harmonis sebagai satu kesatuan yang kohesif untuk memfasilitasi pertukaran data yang akurat. Dalam arsitektur sistem informasi modern, REST API (*Representational State Transfer Application Programming Interface*) telah menjadi protokol standar utama yang

memungkinkan komunikasi antar-aplikasi yang berbeda platform melalui jaringan internet menggunakan protokol HTTP (Gupta, 2025).



Gambar 2.1 Konsep REST API

Syarif dan Wahab (2024) dalam penelitian menjelaskan bahwa integrasi sistem di lingkungan institusi pendidikan sangat krusial untuk mencegah terjadinya isolasi data (*data silos*). Dengan memanfaatkan REST API, pengembang dapat membangun aplikasi operasional baru yang tetap terhubung dengan basis data utama tanpa harus melakukan perombakan besar pada infrastruktur yang sudah ada.

Kendala utama dalam pengembangan sistem informasi operasional baru sering kali terletak pada terbatasnya izin akses langsung ke basis data induk demi alasan keamanan dan integritas data master. Upaya sinkronisasi data mahasiswa melalui akses *direct query* ke server database pusat dinilai sangat berisiko karena dapat memicu beban berlebih pada performa server serta membuka celah kerentanan keamanan yang tidak diinginkan. Ikhwan (2026) dalam penelitiannya menyoroti bahwa perbedaan server, platform, dan basis data antar sistem menyebabkan pertukaran data menjadi kurang efisien dan berpotensi menimbulkan inkonsistensi informasi. Kondisi ini menuntut adanya mekanisme pertukaran data yang bisa mengatasi hal ini guna memastikan data sensitif pada sistem utama tetap terjaga tanpa kehilangan kemampuan untuk diperbarui secara otomatis.

Implementasi metode penarikan data (*data pulling*) berbasis arsitektur RESTful menjadi solusi teknis yang paling aman dan efisien untuk mengatasi

keterpisahan basis data tersebut. Melalui arsitektur ini, sistem baru dapat meminta data spesifik, seperti identitas mahasiswa, dan menerimanya dalam format JSON (*JavaScript Object Notation*) yang ringan serta mudah diolah oleh bahasa pemrograman modern. Ikhwan (2026) menyimpulkan bahwa penggunaan REST API sebagai jembatan integrasi tidak hanya mempercepat proses sinkronisasi informasi secara mandiri, tetapi juga memberikan fleksibilitas bagi pengembang untuk membangun sistem informasi akademik yang lebih terstruktur. Pendekatan ini memastikan aplikasi Ma'had tetap sinkron dengan data universitas induk secara presisi tanpa membahayakan integritas pangkalan data master sensitif.

2.9 Arsitektur Sistem Berbasis Peran (Role-Based Access Control / RBAC)

Role-Based Access Control (RBAC) merupakan pendekatan keamanan perangkat lunak yang membatasi hak akses pengguna berdasarkan peran fungsional mereka di dalam sebuah organisasi (IBM, 2025). Dalam model ini, setiap pengguna tidak diberikan izin akses secara langsung terhadap sistem, melainkan memperoleh hak akses melalui peran tertentu yang telah ditetapkan sebelumnya. Dengan demikian, pengelolaan keamanan sistem menjadi lebih terstruktur karena setiap peran memiliki batasan fungsi dan tanggung jawab yang jelas.

Dalam penelitian ini, mekanisme pengendalian akses diimplementasikan menggunakan pendekatan RBAC statis, yaitu model pengaturan hak akses di mana izin setiap peran telah ditetapkan secara tetap pada struktur sistem. Hak akses tersebut tidak diubah secara dinamis oleh pengguna atau administrator melalui antarmuka sistem, melainkan telah dirancai sejak tahap pengembangan sistem. Pendekatan ini dipilih karena struktur peran pada sistem relatif stabil dan telah memiliki pembagian tanggung jawab yang jelas, sehingga pengaturan akses dapat ditetapkan secara langsung pada level sistem.

Dalam konteks sistem informasi Pendidikan khususnya Ma'had Al-Ja'miah, penerapan RBAC menjadi sangat relevan karena terdapat berbagai aktor dengan fungsi yang berbeda, seperti admin, koordinator, dosen/ustadz, dan mahasiswa. Pembagian hak akses berdasarkan peran memungkinkan setiap pengguna hanya dapat mengakses fitur yang sesuai dengan tugas dan kewenangannya. Penelitian yang dilakukan oleh Aprisilia Anastasia et al. (2022) mengenai pengembangan

sistem informasi berbasis web untuk pengelolaan aktivitas dosen menunjukkan bahwa penerapan pembagian akses berbasis peran mampu meningkatkan efisiensi proses monitoring sekaligus mengurangi risiko penyalahgunaan akses terhadap data sistem.

Untuk mendukung arsitektur sistem berbasis peran tersebut, mekanisme pengelolaan akses pada sistem ini juga disesuaikan dengan perbedaan sumber identitas pengguna. Mengingat adanya perbedaan hierarki serta sumber data antara staf kampus dan tenaga pengajar Ma'had, proses otorisasi akses diimplementasikan melalui dua metode autentikasi yang berjalan secara paralel. Pertama, autentikasi terintegrasi menggunakan layanan *Single Sign-On* (SSO) UIN Ar-Raniry yang dikhususkan untuk akses akun Koordinator. Kedua, autentikasi lokal (manual) menggunakan kredensial yang dibuat dan dikelola langsung oleh Koordinator, yang diperuntukkan khusus bagi akun tenaga pengajar (Ustadz) agar dapat mengakses portal mereka secara mandiri.

2.10 Teknologi Pengembangan Sistem (Tech Stack)

Pengembangan Sistem Informasi Ma'had Al-Jami'ah ini meninggalkan pendekatan monolitik tradisional dan mengadopsi arsitektur terdistribusi (*decoupled architecture*). Dalam paradigma ini, antarmuka pengguna (*frontend*) dan logika bisnis serta komputasi server (*backend*) dipisahkan secara struktural namun saling berkomunikasi melalui *Application Programming Interface* (API). Pemisahan tanggung jawab (*separation of concerns*) ini bertujuan untuk memastikan skalabilitas sistem, mempermudah pemeliharaan (*maintainability*), serta menciptakan lapisan keamanan tambahan di mana logika manipulasi basis data terisolasi sepenuhnya dari sisi klien (Newman, 2021). Implementasi arsitektur ini didukung oleh kombinasi teknologi modern, meliputi Next.js, Nest.js, dan PostgreSQL.

2.10.1 Framework Frontend (Next.js)

Untuk menangani lapisan presentasi dan interaksi pengguna (*user interface*), sistem ini dikembangkan menggunakan Next.js, sebuah framework berbasis pustaka React. Berbeda dengan aplikasi *Single Page Application* (SPA) konvensional yang membebankan proses rendering pada peramban pengguna

(*Client-Side Rendering*), Next.js unggul melalui kapabilitas komputasi prapemuatan (pre-rendering), seperti *Server-Side Rendering* (SSR) dan *Static Site Generation* (SSG) (Vercel, 2025).

Dalam konteks portal Ma'had, kapabilitas SSR pada Next.js sangat krusial untuk mengamankan fitur layanan public (Pengecekan kelulusan dan sertifikat). Validasi NIM dieksekusi secara aman di level server sebelum dirender menjadi halaman HTML statis, sehingga menutupi celah manipulasi skrip dari sisi klien. Selain itu, arsitektur berbasis komponen (*component-based architecture*) pada ekosistem React memungkinkan penggunaan ulang elemen visual (*reusable components*) untuk mempercepat pengembangan antarmuka secara modular dan konsisten.

2.10.2 Framework Backend (Nest.js)

Logika bisnis, pemrosesan algoritma uniting kelas, dan manajemen otorisasi sistem dikendalikan sepenuhnya oleh Nest.js. Nest.js adalah framework Node.js sisi peladen (*server-side*) yang secara arsitektural terinspirasi dari prinsip desain Enterprise dan menggunakan bahasa TypeScript (Nestjs Documentation, 2025). Penggunaan TypeScript yang menerapkan sistem pengetikan statis (*static typing*) secara empiris terbukti mampu mengidentifikasi anomali kode sejak fase kompilasi dan mereduksi kesalahan saat *runtime* (Cherny, 2019), sebuah keandalan yang mutlak diperlukan dalam komputasi nilai akademik dan presensi mahasiswa.

Nest.js mengadopsi prinsip desain *Object-Oriented Programming* (OOP) dan *Dependency Injection* yang sangat terstruktur. Keunggulan utama Nest.js dalam penelitian ini terletak pada ketersediaan modul *Guards* dan *Middleware*. Modul tersebut difungsikan sebagai gerbang validasi berlapis untuk mengimplementasikan *Role-Based Access Control* (RBAC), memastikan bahwa *endpoint* API untuk manipulasi kelas hanya dapat diakses oleh *JSON Web Token* (JWT) terautentikasi milik Koordinator, sementara *endpoint* penilaian dikunci eksklusif untuk otorisasi Ustadz.

2.10.3 Sistem Manajemen Basis Data (PostgreSQL)

Sebagai fondasi penyimpanan data persisten, sistem ini menggunakan PostgreSQL, sebuah *Object-Relational Database Management System* (ORDBMS)

yang bersifat *Open Source* (Postgresql Team, 2023). Pengelolaan data akademik yang saling berelasi erat seperti entitas mahasiswa, kelas asrama, ustadz, absensi, dan nilai membutuhkan tingkat integritas struktural yang sangat presisi.

PostgreSQL dipilih karena kepatuhannya yang ketat terhadap standar arsitektur transaksi ACID (*Atomicity, Consistency, Isolation, Durability*). Sifat konsistensi dan isolasi pada PostgreSQL menjamin bahwa proses eksekusi otomatisasi uniting kelas yang melibatkan manipulasi pemindahan ratusan baris data ke tabel kelas secara simultan tidak akan mengalami anomali ras (*race conditions*) atau menghasilkan data yatim (*orphan data*) jika terjadi kegagalan sistem di tengah transaksi (Postgresql Team, 2023). Keandalan relasional ini menjadikan PostgreSQL sebagai platform yang jauh lebih robust untuk menggantikan metode pendataan konvensional berbasis spreadsheet.

2.11 Metodologi Pengembangan dan Pengujian Sistem

Rekayasa perangkat lunak yang sistematis membutuhkan kerangka kerja (*framework*) terstruktur yang dikenal sebagai *System Development Life Cycle* (SDLC). SDLC mendikte tahapan logis mulai dari inisiasi, analisis kebutuhan, perancangan, implementasi, pengujian, hingga pemeliharaan sistem. Dalam pengembangan Sistem Informasi Ma'had Al-Jami'ah ini, paradigma SDLC yang diadopsi adalah model Sekuensial Linier (*Linear Sequential Model*) atau yang secara luas diakui sebagai metode *Waterfall*. Menurut Pressman dan Maxim (2019), metode *Waterfall* menawarkan pendekatan pengembangan yang sangat sistematis dan berurutan, di mana satu fase harus diselesaikan dan divalidasi terlebih dahulu sebelum fase berikutnya dapat dimulai. Pemilihan model ini dilandasi oleh karakteristik kebutuhan sistem Ma'had seperti alur proses *uniting* kelas dan pengelolaan nilai ustadz yang telah didefinisikan secara komprehensif dan spesifik sejak awal (fase *requirement gathering*), sehingga meminimalkan risiko ambiguitas atau perubahan drastis di pertengahan proses pengembangan.

Fase paling esensial dalam kerangka *Waterfall* untuk menjamin integritas dan keandalan fungsional produk adalah fase pengujian (*testing*). Pengujian sistem ini diimplementasikan melalui pendekatan *Black Box Testing*. Pendekatan ini merupakan teknik pengujian perangkat lunak yang berfokus secara eksklusif pada

spesifikasi fungsional tanpa membedah struktur internal, logika kontrol, atau kode sumber aplikasi. Pengujian ini memvalidasi apakah setiap *input* operasional seperti pencarian status kelulusan menggunakan NIM pada portal publik, atau eksekusi algoritma *uniting* di *dashboard* coordinator menghasilkan *output* yang sesuai dengan spesifikasi rancangan. Sahyudi dan Voutama (2025) menegaskan bahwa teknik pengujian *Black Box* sangat esensial dan efektif dalam sistem informasi akademik untuk mengidentifikasi anomali antarmuka, kesalahan struktur data, dan inisialisasi fungsi yang tidak semestinya sebelum sistem diserahkan kepada pengguna.

Di sisi lain, keandalan fungsional (*functional reliability*) yang dibuktikan melalui *Black Box Testing* belum sepenuhnya merepresentasikan tingkat penerimaan dan kemudahan penggunaan aplikasi di lapangan. Oleh karena itu, evaluasi sistem ini dilanjutkan dengan pendekatan *Usability Testing* menggunakan instrumen *System Usability Scale* (SUS) sebagai metode validasi akhir. SUS merupakan kuesioner terstandarisasi yang dirancang untuk mengukur tingkat kepuasan, efisiensi, dan kebergunaan subjektif dari sebuah antarmuka sistem secara cepat dan terukur. Dalam konteks aplikasi tata kelola Ma'had ini, pengujian SUS akan didistribusikan langsung kepada pihak terkait setelah mereka menyimulasikan penggunaan sistem. Menurut penelitian Pratama, Umam, dan Yakok (2024), metrik evaluasi menggunakan SUS sangat krusial karena menghasilkan skor numerik terstandarisasi yang memberikan bukti kuantitatif valid terkait pengalaman pengguna (*user experience*). Skor empiris tersebut akan menjawab secara pasti apakah integrasi sistem baru ini benar-benar dapat diterima secara ergonomis dan terbukti menunjang efisiensi administratif dibandingkan metode manual *spreadsheet* sebelumnya.

2.12 Penelitian Terdahulu

Penelitian terdahulu menjadi landasan fundamental untuk mengetahui sejauh mana sebuah topik telah dieksplorasi, serta untuk menemukan celah penelitian (*research gap*) guna menegaskan posisi dan kebaruan (*novelty*) dari penelitian yang sedang dilakukan. Terdapat beberapa literatur relevan terkait pengembangan sistem informasi di lingkungan Ma'had yang menjadi rujukan utama dalam perancangan Sistem Informasi Ma'had ini.

Penelitian Pertama dilakukan oleh Ahmad dan Wahid (2024) yang berjudul “*Pemanfaatan Aplikasi Sistem Informasi Santri di Ma’had Tahfidh Al-Qur’an Putri Al-Amien*”. Penelitian ini menyoroti urgensi digitalisasi data santri dan operasional akademik di lingkungan asrama/ma’had melalui pendekatan aplikasi berbasis web. Hasil penelitian membuktikan bahwa sistem terkomputerisasi efektif memangkas waktu pencarian data. Namun, kelemahan sistem ini adalah arsitekturnya yang dibangun murni berdiri sendiri (*standalone*). Sistem tersebut tidak merancang skenario keberlanjutan data (*data continuity*), sehingga belum mengakomodasi mekanisme integrasi penarikan data (*data pulling*) dari basis data atau sistem informasi Ma’had yang lama (*legacy system*).

Penelitian Kedua dikemukakan dalam riset tentang sistem tata kelola Ma’had Al-Jami’ah oleh Makmur (2024) dengan judul “*Implementasi Sistem Informasi Manajemen Pendidikan Di Ma’had Syaikh Muhammad Bin Sholeh*”. Penelitian ini berfokus pada perancangan sistem informasi untuk mengawasi aktivitas mahasiswa secara terorganisir di bawah birokrasi perguruan tinggi. Walaupun penelitian ini sangat relevan dengan tata kelola Ma’had, fungsionalitas aplikasinya masih terbatas pada pencatatan aktivitas dasar dan belum mencakup otomatisasi pemetaan kelas (*uniting*) secara terpusat bagi koordinator maupun penyediaan portal layanan mandiri (*self-service*) bagi mahasiswa.

Penelitian Ketiga dilakukan oleh Ikhwani (2026) dalam jurnal ilmiah *RIGGS* dengan judul “*Pengembangan RESTful Web Service Untuk Integrasi Data Pada Sistem Informasi Akademik*”. Meskipun tidak secara spesifik berlokasi di Ma’had, penelitian ini memecahkan masalah arsitektural yang sama persis dengan yang dihadapi Ma’had Al-Jami’ah UIN Ar-Raniry, yakni bagaimana menyinkronkan aplikasi baru dengan sistem kampus yang lama. Penelitian ini membuktikan bahwa penggunaan antarmuka REST API sangat aman dan efektif dalam mencegah entri data ganda. Namun, penelitian ini belum menerapkan kontrol hak akses keamanan spesifik (RBAC) untuk memisahkan *dashboard* tenaga pengajar dan administrator.

Berdasarkan tinjauan dari ketiga literatur di atas, posisi dan kebaruan (*novelty*) dari penelitian ini dapat dilihat pada Tabel 2.1

Tabel 2.1 Perbandingan Penelitian Terdahulu dengan Penelitian Saat Ini

Judul & Peneliti	Fokus Penelitian	Kelemahan / Celah (Gap)	Posisi Penelitian Saat Ini (Rejoul)
Pemanfaatan Aplikasi Sistem Informasi Santri di Ma'had Tahfidh Al-Qur'an Putri Al-Amien (Mardhatillah Ahmad & Musleh Wahid, 2024)	Pemanfaatan sistem informasi berbasis web untuk pencatatan data santri di Ma'had.	Sistem bersifat tunggal (<i>standalone</i>) dan tidak mengakomodasi skenario keberlanjutan data dari sistem operasional lama (<i>legacy system</i>).	Mengembangkan sistem Ma'had yang terintegrasi untuk menarik data (<i>data pulling</i>) secara langsung dari pangkalan data sistem Ma'had terdahulu melalui jalur API.
Implementasi Sistem Informasi Manajemen Pendidikan Di Ma'had Syaikh Muhammad Bin Sholeh (Makmur, 2024)	Implementasi sistem informasi manajemen pendidikan dan pemantauan di lingkungan Ma'had.	Belum mengakomodasi fitur otomatisasi pembagian kelas (<i>uniting</i>) dan pencetakan dokumen mandiri.	Menghadirkan <i>dashboard uniting</i> khusus koordinator dan portal layanan <i>self-service</i> untuk unduh sertifikat.
Pengembangan RESTful Web Service Untuk Integrasi Data Pada Sistem Informasi Akademik (Ikhwan, 2026)	Penggunaan arsitektur REST API untuk integrasi data akademik lintas sistem.	Hanya berfokus pada jalur API, tidak membahas portal tenaga pengajar maupun tata kelola Ma'had.	Memadukan REST API dengan <i>Role-Based Access Control (RBAC)</i> untuk mengelola <i>dashboard</i> Koordinator dan Ustadz Ma'had.

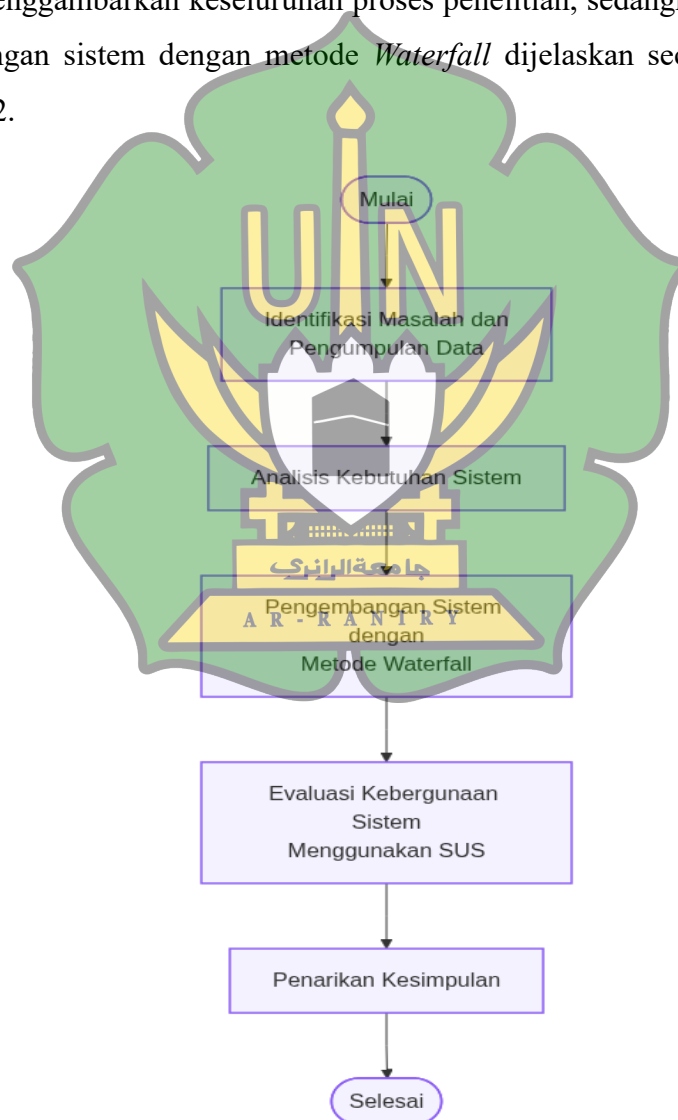
BAB 3

METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Penelitian ini berfokus pada pengembangan dan evaluasi perangkat lunak berupa Sistem Informasi Operasional Ma'had Al-Jami'ah berbasis web. Pelaksanaan penelitian disusun secara sistematis mulai dari identifikasi masalah dan pengumpulan data hingga penarikan kesimpulan.

Alur penelitian yang digunakan ditunjukkan pada Gambar 3.1. Diagram tersebut menggambarkan keseluruhan proses penelitian, sedangkan tahapan teknis pengembangan sistem dengan metode *Waterfall* dijelaskan secara terpisah pada Gambar 3.2.



Gambar 3.1 Diagram Alur Penelitian

Berdasarkan Gambar 3.1, tahap pertama penelitian dimulai dengan identifikasi masalah dan pengumpulan data. Pada tahap ini, peneliti mengidentifikasi permasalahan dalam pengelolaan akademik dan administrasi Ma'had Al-Jami'ah UIN Ar-Raniry Banda Aceh, khususnya pada proses pemetaan kelas (*uniting*), pengelolaan presensi dan nilai, serta pelayanan pengecekan status kelulusan dan pengunduhan sertifikat. Data penelitian dikumpulkan melalui studi literatur, observasi, dan wawancara untuk memperoleh informasi mengenai proses bisnis yang sedang berjalan, kendala yang dihadapi pengguna, dan kebutuhan terhadap sistem yang akan dikembangkan.

Tahap kedua adalah analisis kebutuhan sistem. Pada tahap ini, data yang telah dikumpulkan dianalisis untuk menentukan kebutuhan fungsional dan nonfungsional sistem. Kebutuhan fungsional disusun berdasarkan peran pengguna, yaitu koordinator, ustadz/ustadzah, dan mahasantri. Sementara itu, kebutuhan nonfungsional mencakup keamanan, kemudahan penggunaan, kinerja sistem, kompatibilitas peramban, serta pembatasan akses berdasarkan peran pengguna. Hasil analisis tersebut menjadi dasar dalam menentukan fungsi, ruang lingkup, dan batasan sistem yang dikembangkan.

Tahap ketiga adalah pengembangan sistem menggunakan metode Waterfall. Pada tahap ini, kebutuhan yang telah ditetapkan diterjemahkan ke dalam perancangan arsitektur sistem, basis data, diagram UML, dan antarmuka pengguna. Hasil perancangan kemudian diimplementasikan menjadi aplikasi berbasis web yang mencakup dashboard koordinator, dashboard ustadz/ustadzah, serta layanan publik bagi mahasantri. Sistem yang telah diimplementasikan selanjutnya diuji secara fungsional menggunakan Black-Box Testing untuk memastikan bahwa setiap fungsi utama berjalan sesuai dengan spesifikasi kebutuhan. Dengan demikian, pengujian fungsional merupakan bagian dari tahapan pengembangan sistem menggunakan metode Waterfall.

Tahap keempat adalah evaluasi kebergunaan sistem menggunakan System Usability Scale (SUS). Pada tahap ini, pengguna menjalankan fungsi sistem sesuai dengan perannya, kemudian memberikan penilaian melalui sepuluh pernyataan standar SUS dengan skala Likert 1 sampai 5. Evaluasi melibatkan kelompok

koordinator, ustadz/ustadzah, dan mahasantri sebagai pengguna sasaran sistem. Jawaban responden selanjutnya dihitung menggunakan ketentuan SUS untuk memperoleh skor kebergunaan sistem pada masing-masing kelompok pengguna.

Tahap kelima adalah penarikan kesimpulan. Pada tahap ini, hasil pengembangan sistem, pengujian fungsional menggunakan Black-Box Testing, dan evaluasi kebergunaan menggunakan SUS dianalisis secara keseluruhan. Hasil analisis tersebut digunakan untuk menjawab rumusan masalah, menilai ketercapaian tujuan penelitian, serta mengidentifikasi keterbatasan sistem dan kemungkinan pengembangan lebih lanjut. Setelah kesimpulan diperoleh, seluruh rangkaian penelitian dinyatakan selesai.

3.1.1 Metode Pengumpulan Data

Dalam tahap rekayasa Sistem Informasi Ma'had Al-Jami'ah UIN Ar-Raniry, pengumpulan data empiris dan teoritis merupakan langkah esensial untuk mengidentifikasi kelemahan pada sistem konvensional yang sedang berjalan, serta merumuskan spesifikasi kebutuhan perangkat lunak secara presisi. Metode pengumpulan data yang diimplementasikan dalam penelitian ini mencakup tiga pendekatan utama, yaitu:

3.1.1.1 Studi Literatur

Studi literatur merupakan metode pengumpulan data sekunder yang dieksekusi dengan mengumpulkan, mengkaji, dan mensintesis berbagai referensi ilmiah maupun dokumen teknis yang relevan dengan domain penelitian. Rujukan utama mencakup artikel jurnal yang berkaitan dengan arsitektur sistem informasi berbasis web, integrasi *Representational State Transfer Application Programming Interface* (REST API), manajemen basis data relasional, keamanan *Role-Based Access Control* (RBAC), metode pengujian perangkat lunak, serta dokumen kebijakan resmi penyelenggaraan Ma'had dari berbagai sumber yang tersedia. Lebih lanjut, studi literatur ini juga melibatkan kajian mendalam terhadap dokumentasi resmi (*official whitepaper*) dari teknologi tumpukan (*tech stack*) yang digunakan, yakni Next.js, Nest.js, dan PostgreSQL, guna memastikan bahwa perancangan arsitektur perangkat lunak mematuhi standar industri modern.

3.1.1.2 Observasi

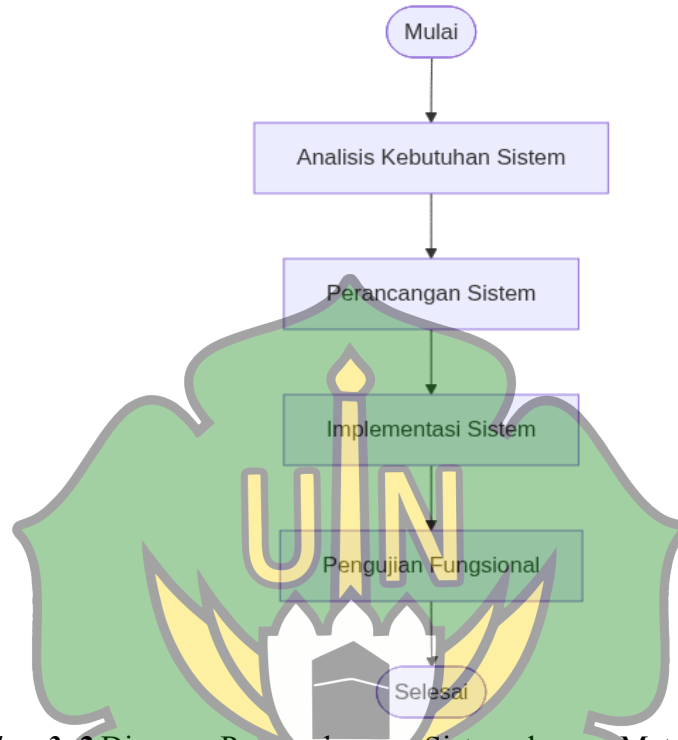
Observasi adalah teknik pengumpulan data primer yang dilakukan melalui pengamatan sistematis secara langsung terhadap objek penelitian di lapangan. Pada fase ini, peneliti mengamati secara saksama alur proses bisnis (*business process*) tata kelola administratif di lingkungan Ma'had Al-Jami'ah UIN Ar-Raniry Banda Aceh. Fokus utama pengamatan diarahkan pada prosedur *uniting* kelas asrama oleh staf koordinator yang saat ini masih bergantung pada manipulasi data manual menggunakan perangkat lunak *spreadsheet*. Selain itu, observasi juga menyoroti mekanisme rekapitulasi presensi dan nilai akademik oleh ustadz, serta prosedur pengecekan status kelulusan publik. Melalui observasi ini, peneliti dapat memetakan titik kemacetan informasi (*bottleneck*) dan inefisiensi operasional yang menjadi dasar pengembangan modul sistem baru.

3.1.1.3 Wawancara

Wawancara dilakukan secara langsung dengan pihak pengelola Ma'had, termasuk staf administrasi dan koordinator, untuk memperoleh informasi yang lebih mendalam mengenai kebutuhan sistem. Wawancara ini bertujuan untuk menggali permasalahan yang dihadapi pengguna, kebutuhan fitur tambahan seperti dashboard ustadz, serta harapan terhadap sistem yang akan dikembangkan. Teknik wawancara yang digunakan bersifat semi-terstruktur, sehingga memungkinkan peneliti mendapatkan informasi yang terarah, namun tetap fleksibel sesuai dengan kondisi lapangan.

3.1.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah Waterfall. Metode ini ditempatkan sebagai bagian khusus dari keseluruhan alur penelitian karena hanya mengatur proses teknis pembangunan perangkat lunak. Tahapan Waterfall yang diterapkan ditunjukkan pada Gambar 3.2.



Gambar 3. 2 Diagram Pengembangan Sistem dengan Metode Waterfall

Berdasarkan Gambar 3.2, pengembangan sistem dimulai dari analisis kebutuhan sistem, dilanjutkan dengan perancangan sistem, implementasi, dan pengujian fungsional. Hasil setiap tahap menjadi dasar bagi tahap berikutnya agar sistem yang dikembangkan tetap sesuai dengan kebutuhan operasional Ma'had.

3.1.3 Tahap Analisis Kebutuhan Sistem

Tahap analisis kebutuhan merupakan proses awal dalam pengembangan sistem yang bertujuan untuk mengidentifikasi, memahami, dan mendokumentasikan kebutuhan pengguna secara menyeluruh. Tahap ini dilakukan berdasarkan hasil observasi dan wawancara dengan pihak pengelola Ma'had Al-Jami'ah UIN Ar-Raniry Banda Aceh. Analisis kebutuhan menjadi dasar dalam menentukan ruang lingkup sistem, fitur yang dikembangkan, serta batasan implementasi agar sistem yang dibangun sesuai dengan kebutuhan operasional.

Pada tahap ini, kebutuhan sistem diklasifikasikan 3 kategori yaitu kebutuhan fungsional, kebutuhan non-fungsional dan Kebutuhan Perangkat Keras dan Perangkat Lunak.

3.1.3.1 Kebutuhan Fungsional

Kebutuhan fungsional mendefinisikan layanan interaktif, proses bisnis, dan perilaku (*behavior*) spesifik yang harus mampu dieksekusi oleh sistem sesuai dengan hak akses masing-masing aktor (pengguna). Rincian fungsionalitas yang akan dibangun pada Sistem Informasi Ma'had ini dapat dilihat pada Tabel 3.1.

Tabel 3.1 Daftar Kebutuhan Fungsional Sistem Berdasarkan Aktor

No	Aktor	Kebutuhan Fungsional
1	Koordinator	Melakukan sinkronisasi atau penarikan data mahasantri secara <i>read-only</i> dari Sistem Ma'had lama melalui REST API.
		Melakukan autentikasi masuk (<i>login</i>) menggunakan integrasi SSO kampus.
		Mengelola (CRUD) data master pendukung seperti Lokasi Gedung, Mata Pelajaran, dan data akun Ustadz.
		Membuat dan mengonfigurasi kapasitas unit kelas baru.
		Mengeksekusi proses pemetaan kelas (<i>uniting</i>) mahasantri secara manual maupun massal (otomatis).
		Memantau rekapitulasi distribusi kelas dan keterisian kapasitas asrama.
2	Ustadz	Melakukan autentikasi masuk (<i>login</i>) menggunakan kredensial lokal yang dibuat oleh Koordinator.
		Melihat daftar unit kelas yang diampu beserta daftar nama mahasantri di dalamnya.
		Melakukan pencatatan kehadiran (presensi) harian mahasantri.
		Menginput nilai mahasantri.

No	Aktor	Kebutuhan Fungsional
3	Mahasantri (Publik)	Memvalidasi identitas menggunakan Nomor Induk Mahasiswa (NIM).
		Mengecek status kelulusan akhir akademik Ma'had tanpa perlu <i>login</i> ke <i>dashboard</i> .
		Mengunduh dokumen sertifikat kelulusan dalam format PDF secara mandiri (<i>self-service</i>).

3.1.3.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional mendefinisikan batasan teknis, parameter keamanan, dan standar performa yang harus dipenuhi agar arsitektur sistem dapat beroperasi secara optimal dan aman. Rincian spesifikasi non-fungsional ini diuraikan pada Tabel 3.2

Tabel 3.2 Spesifikasi Kebutuhan Non-Fungsional Sistem

No	Parameter Kualitas	Deskripsi Kebutuhan
1	Keamanan Akses (Security)	Sistem menerapkan arsitektur <i>Role-Based Access Control</i> (RBAC) Statis untuk mengisolasi <i>dashboard</i> Koordinator dan Ustadz, serta menggunakan autentikasi ganda (Hibrida: SSO & Lokal).
2	Interoperabilitas (Interoperability)	Sistem harus mampu melakukan penarikan data (<i>data pulling</i>) mahasiswa dari pangkalan data sistem Ma'had yang lama (<i>legacy system</i>) melalui jalur komunikasi <i>REST API</i> .
3	Ketersediaan (Availability)	Sistem berbasis web (<i>web-based</i>) penuh sehingga dapat diakses melalui berbagai perangkat (PC, laptop, tablet, atau ponsel) tanpa memerlukan instalasi perangkat lunak tambahan.

No	Parameter Kualitas	Deskripsi Kebutuhan
4	Keandalan Data (Reliability)	Operasi pemindahan massal (<i>uniting</i>) seperti ratusan data mahasiswa ke dalam unit kelas harus dijamin oleh arsitektur transaksi basis data (<i>ACID Compliance</i>) untuk mencegah terjadinya anomali data jika terjadi kegagalan koneksi di tengah proses.

3.1.3.3 Kebutuhan Perangkat Keras dan Perangkat Lunak

Untuk merealisasikan rancangan sistem secara utuh, tahapan rekayasa perangkat lunak memerlukan spesifikasi instrumen teknis yang memadai guna mendukung proses penulisan kode (*development*) hingga kompilasi (*rendering*). Rincian alat dan bahan penelitian yang digunakan dijabarkan pada Tabel 3.3 dan Tabel 3.4.

Tabel 3.3 Kebutuhan Perangkat Keras

No	Kategori	Spesifikasi Digunakan
1	Prosesor (CPU)	Intel Core i5 i5-9300H
2	Memori (RAM)	16 GB DDR4
3	Penyimpanan (Storage)	Solid State Drive (SSD) dengan kapasitas 256 GB

Tabel 3.4 Kebutuhan Perangkat Lunak dan Alat

No	Kategori	Teknologi yang Digunakan	Versi
1	Text Editor / IDE	Visual Studio Code (VS Code)	1.134.0
2	Runtime Environment	Node.js	22.14.0
3	Package Manager	NPM	10.9.2
4	Version Control System	Git	2.39.1.windows.1

No	Kategori	Teknologi yang Digunakan	Versi
5	Framework Frontend	Next.js	16.2.10
6	Framework Backend	NestJS	11.1.18
7	Database Management	PostgreSQL	16
8	Object-Relational Mapping	Prisma ORM	6.19.3
9	Design Tool	Figma	126.7.10
10	API Testing Tool	Postman	11.99.0
11	Web Browser	Google Chrome	151.0.7922.174

3.1.4 Tahap Perancangan Sistem

Tahap perancangan sistem merupakan proses menerjemahkan kebutuhan yang telah dianalisis ke dalam bentuk desain teknis yang terstruktur. Perancangan ini bertujuan untuk memberikan gambaran menyeluruh mengenai arsitektur sistem, interaksi antar pengguna, struktur basis data, serta rancangan antarmuka sebelum sistem diimplementasikan.

Dalam penelitian ini, perancangan sistem meliputi Perancangan Arsitektur Sistem, Perancangan *Use Case Diagram*, Perancangan *Activity Diagram*, Perancangan Basis Data (ERD), dan Perancangan Antarmuka (User Interface).

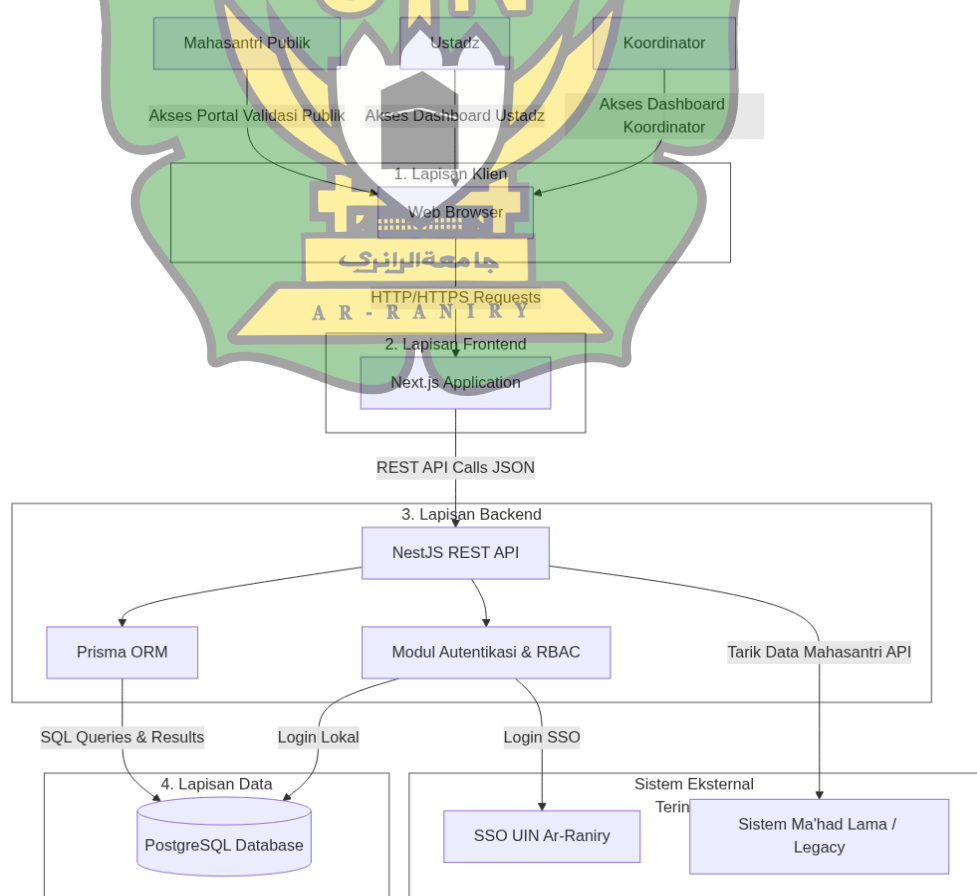
3.1.4.1 Perancangan Arsitektur Sistem

Sistem Informasi Ma'had dirancang menggunakan arsitektur *Client-Server* terpisah (*decoupled architecture*). Pendekatan ini memisahkan lapisan presentasi antarmuka dari lapisan logika bisnis untuk menjamin skalabilitas dan keamanan aplikasi.

- **Lapisan Klien (*Frontend Layer*):** Dibangun menggunakan Next.js. Lapisan ini bertugas merender antarmuka pengguna di peramban web (*browser*) dan mengirimkan permintaan (*HTTP Request*) ke peladen.

- **Lapisan Peladen (*Backend Layer*):** Dibangun menggunakan NestJS yang diintegrasikan dengan Prisma ORM. Lapisan ini bertindak sebagai otak komputasi sistem yang memproses algoritma *uniting* kelas, mengelola modul autentikasi dan otorisasi *Role-Based Access Control* (RBAC), serta menyediakan jalur komunikasi *RESTful API* berbasis JSON.
- **Lapisan Data (*Data Layer*):** Menggunakan pangkalan data relasional PostgreSQL untuk menyimpan entitas akademik secara persisten.
- **Lapisan Integrasi Eksternal (*External Systems*):** Merupakan jalur komunikasi ke luar sistem yang mencakup integrasi layanan Single Sign-On (SSO) UIN Ar-Raniry untuk autentikasi Koordinator dan REST API sistem Ma'had lama untuk menarik identitas mahasiswa, membaca status kelulusan, serta menerima tautan sertifikat secara read-only.

Untuk gambaran visual dari arsitektur sistem yang dibangun bisa dilihat pada Gambar 3.3



Gambar 3.3 Arsitektur Sistem

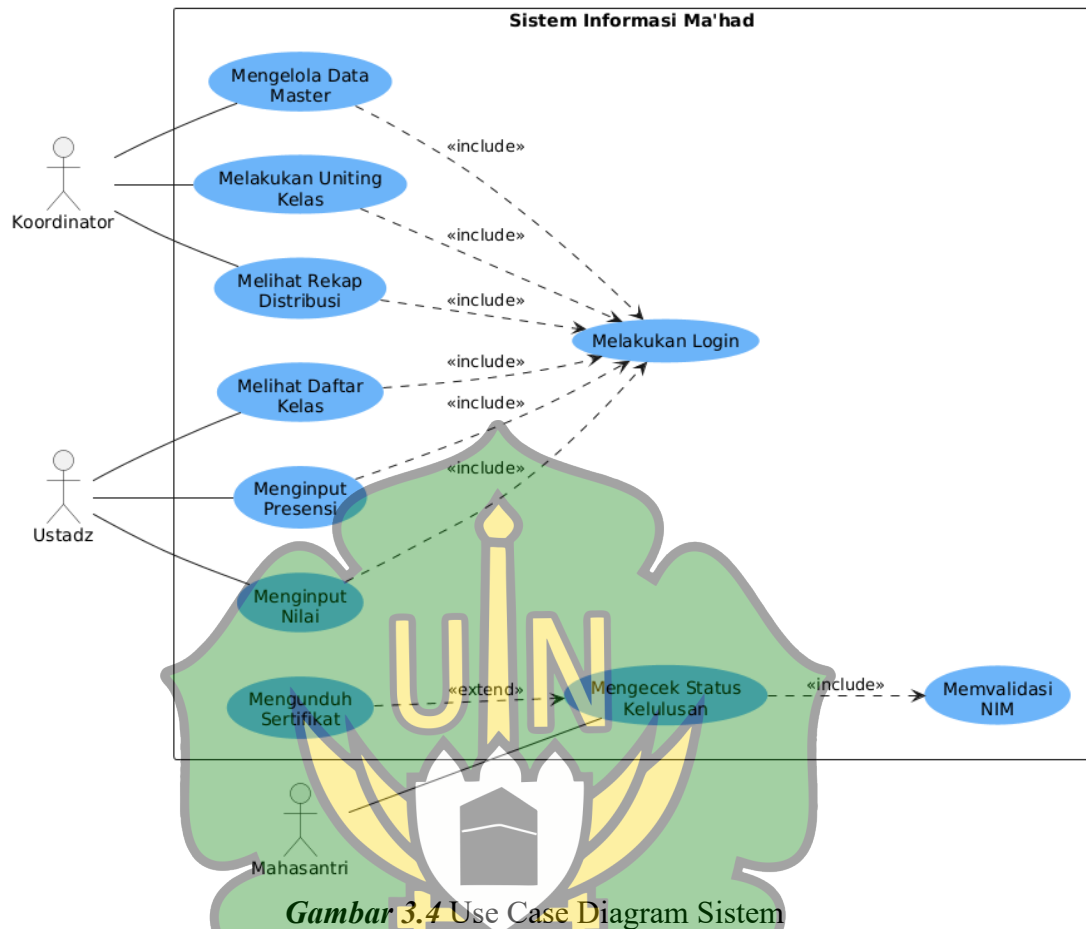
3.1.4.2 Perancangan Use Case Diagram

Use Case Diagram merupakan bagian dari *Unified Modeling Language* (UML) yang difungsikan untuk memetakan perilaku interaktif sistem dari sudut pandang pengguna akhir. Diagram ini mendeskripsikan “siapa” yang menggunakan sistem dan “apa” saja yang dapat mereka lakukan, tanpa merincikan secara teknis bagaimana fungsionalitas tersebut bekerja.

Berdasarkan analisis arsitektur berbasis peran (RBAC), sistem ini mendefinisikan tiga aktor utama dengan batasan wewenang yang saling terisolasi:

- **Koordinator:** Aktor dengan hak akses manajerial (*Super Admin*). Aktor ini dapat melakukan autentikasi login (SSO), melakukan sinkronisasi penarikan data mahasiswa dari sistem lama (*API Pulling*), mengelola data master (ruang, kelas, ustadz, mata pelajaran), mengeksekusi proses uniting kelas, serta melihat rekapitulasi distribusi kelas.
- **Ustadz:** Aktor dengan hak akses operasional akademik. Aktor ini melakukan autentikasi login lokal untuk mengakses *dashboard* kelas yang diampunya, lalu melakukan dua fungsi utama: input presensi harian dan input nilai evaluasi mahasiswa.
- **Mahasiswa:** Aktor publik (tanpa hak otentikasi/sesi *login*). Aktor ini hanya berinteraksi dengan portal layanan mandiri untuk memvalidasi NIM, mengecek status kelulusan, dan mengunduh dokumen sertifikat.

Untuk Ilustrasi *Use Case Diagram* dari ketiga aktor tersebut bisa dilihat pada gambar 3.3.



Gambar 3.4 Use Case Diagram Sistem

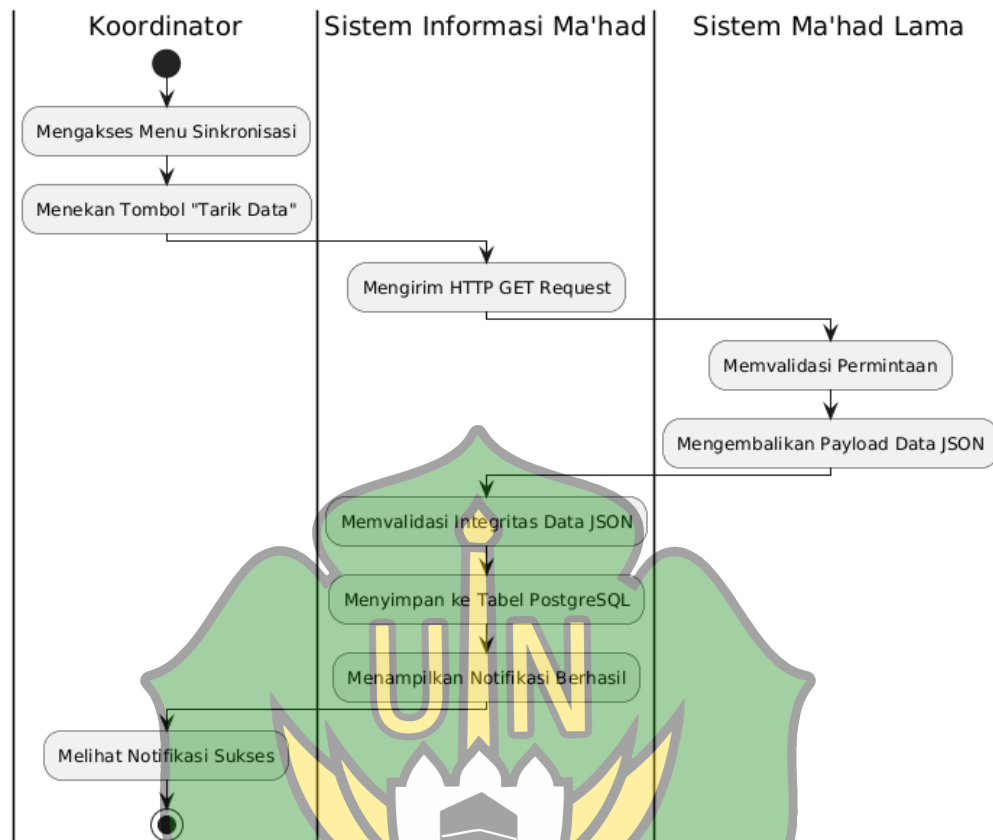
3.1.4.3 Perancangan Activity Diagram

Activity Diagram digunakan untuk menggambarkan alur aktivitas atau proses bisnis yang terjadi dalam sistem secara berurutan. Diagram ini secara visual mendeskripsikan bagaimana sistem merespons setiap interaksi atau instruksi dari pengguna, mulai dari status awal (*initial state*) hingga status akhir (*final state*). Pada pengembangan Sistem Informasi Ma'had ini, terdapat empat alur aktivitas operasional utama yang dirancang, yaitu:

A. Activity Diagram Sinkronisasi Data Mahasantri (*API Pulling*)

Proses ini merupakan alur integrasi krusial di mana sistem baru menarik data master mahasantri dari pangkalan data sistem Ma'had yang lama (*legacy system*) tanpa memerlukan entri data manual.

Proses ini menggambarkan alur pemetaan mahasiswa ke dalam kelas tertentu oleh coordinator, untuk alurnya lihat pada gambar 3.4.

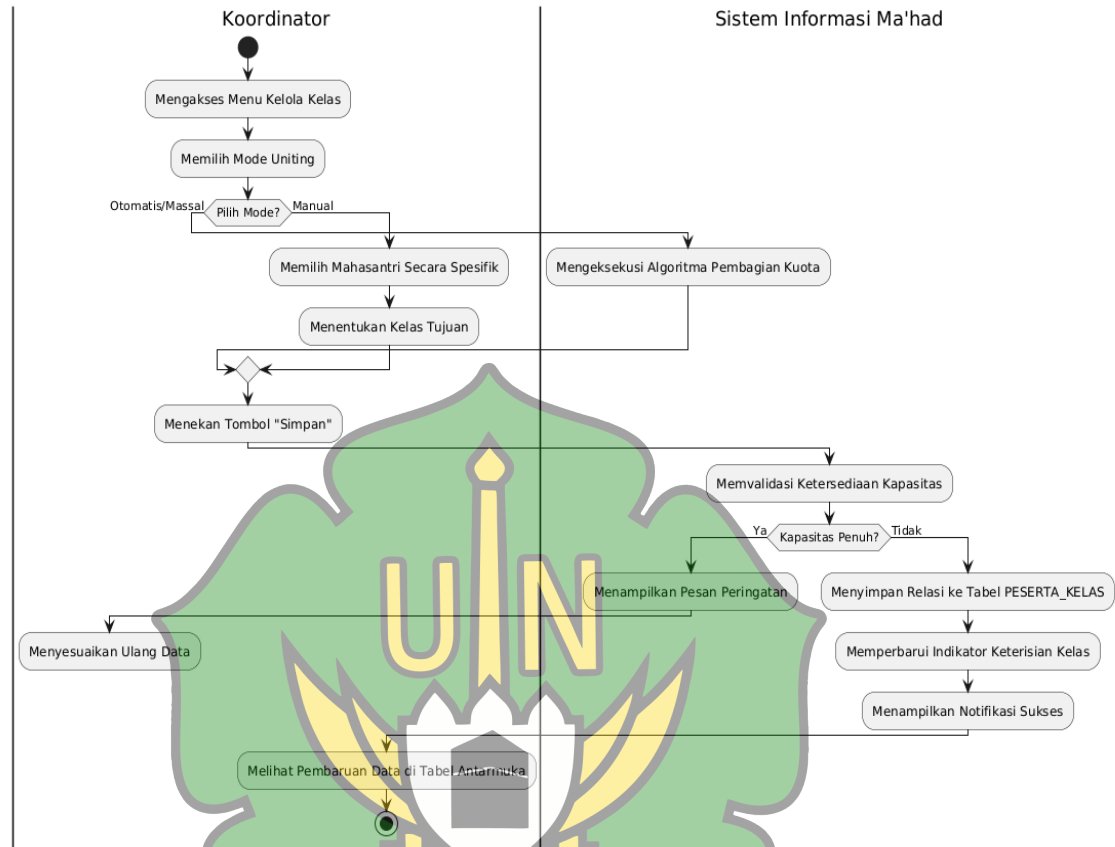


Gambar 3.5 Activity Diagram Sinkronasi Data Mahasantri

Proses dimulai ketika Koordinator mengakses menu Sinkronisasi dan menekan tombol “Tarik Data”. Sistem backend (NestJS) kemudian mengirimkan permintaan (*HTTP GET Request*) ke *endpoint* REST API sistem lama. Setelah sistem lama merespons dengan mengirimkan *payload* data mahasantri dalam format JSON, sistem Rejoul memvalidasi integritas data tersebut, lalu menyimpannya secara otomatis ke dalam tabel basis data PostgreSQL lokal. Proses diakhiri dengan munculnya notifikasi keberhasilan sinkronisasi pada layar Koordinator.

B. Activity Diagram Proses Uniting Mahasantri (Koordinator)

Proses ini memodelkan alur kerja Koordinator dalam mendistribusikan dan memetakan mahasantri ke dalam unit-unit kelas asrama.

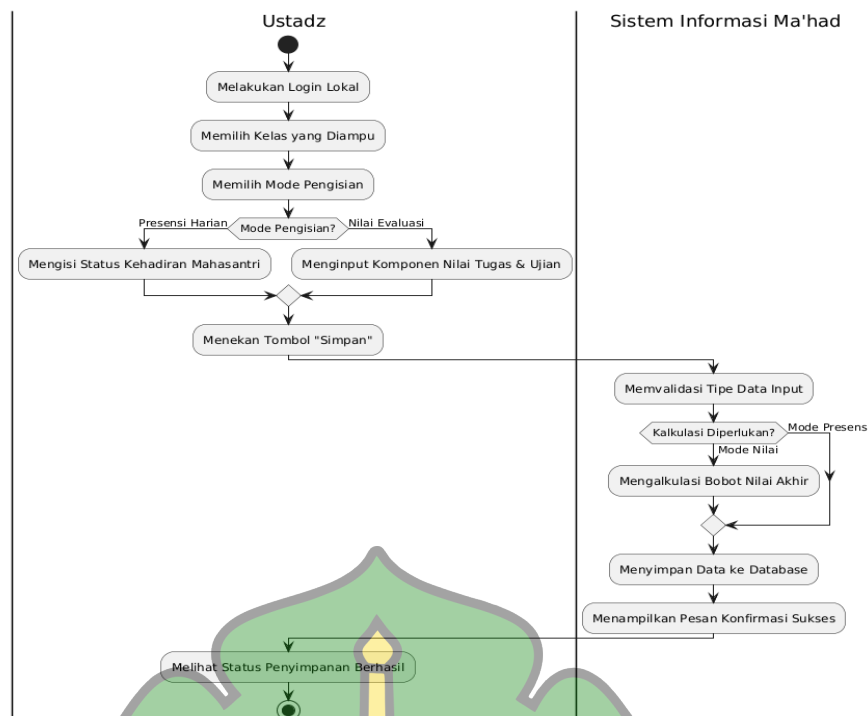


Gambar 3.6 Activity Diagram Uniting

Proses dimulai ketika Koordinator yang telah terautentikasi memilih menu Kelola Kelas dan memilih mode *uniting* (Manual atau Massal/Otomatis). Pada mode otomatis, sistem akan mengeksekusi algoritma pembagian berdasarkan sisa kuota kapasitas masing-masing kelas. Sistem kemudian melakukan validasi ketersediaan ruang, menyimpan relasi penempatan ke dalam tabel transaksi, dan mengembalikan notifikasi sukses beserta pembaruan data pada tabel antarmuka.

C. Activity Diagram Pengelolaan Absensi dan Nilai (Ustadz)

Aktivitas ini mendeskripsikan alur kerja tenaga pengajar (Ustadz) dalam mengelola data evaluasi akademik harian secara terpadu.

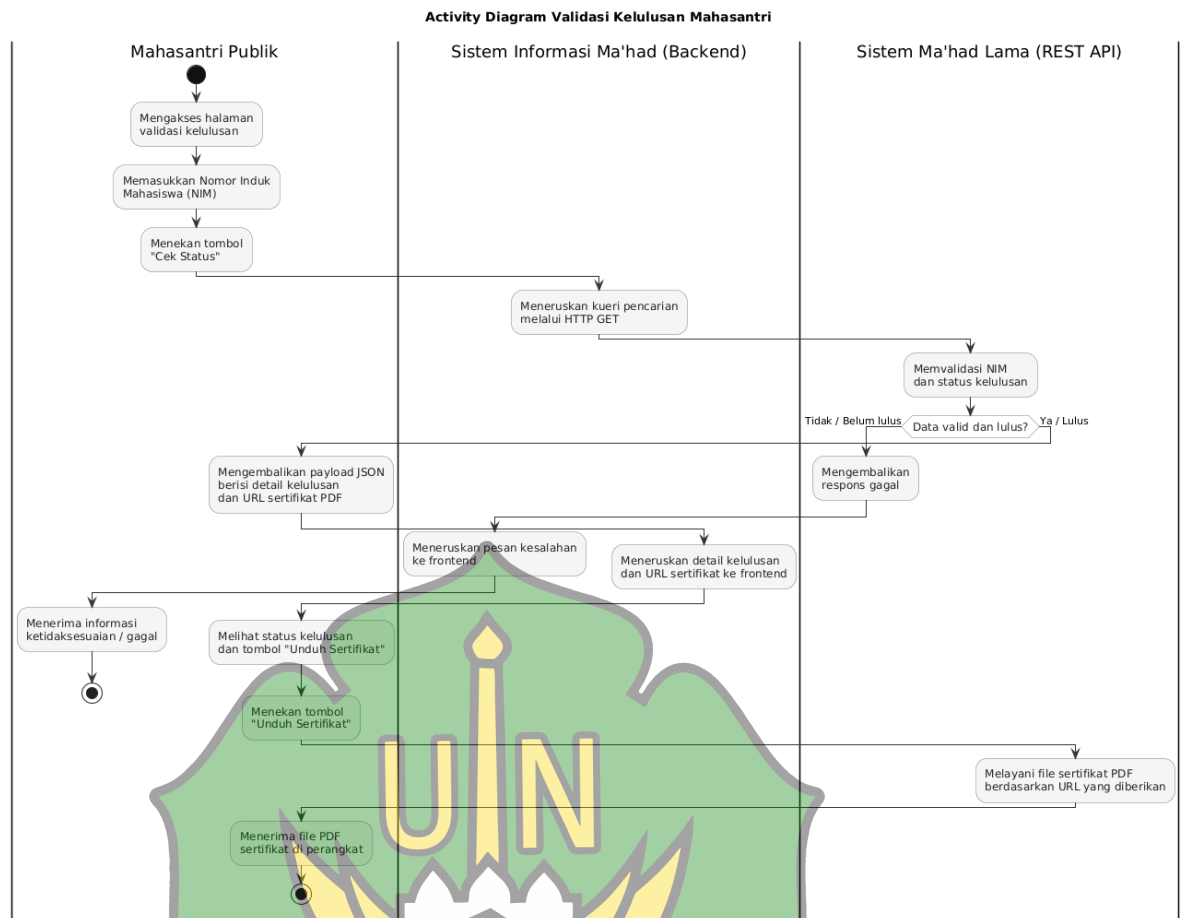


Gambar 3.7 Activity Diagram Absensi & Nilai

Proses dimulai setelah Ustadz berhasil melakukan *login* lokal, sistem akan memuat daftar kelas yang menjadi tanggung jawabnya. Ustadz memilih kelas dan menentukan mode pengisian (Presensi Harian atau Nilai Evaluasi). Setelah formulir antarmuka diisi dan tombol “Simpan” ditekan, sistem *backend* melakukan validasi terhadap tipe data (numerik/status), mengalkulasi bobot nilai akhir (jika dalam mode penilaian), lalu menyimpannya secara permanen ke pangkalan data dan menampilkan pesan konfirmasi.

D. Activity Diagram Validasi dan Unduh Sertifikat (Layanan Publik)

Proses ini memodelkan alur layanan mandiri (*self-service*) pada portal publik yang dapat diakses oleh mahasantri tanpa memerlukan pembuatan sesi otentikasi.



Gambar 3.8 Activity Diagram Validasi Sertifikat

Proses dimulai ketika Mahasantri mengakses halaman validasi kelulusan dan memasukkan Nomor Induk Mahasiswa (NIM). Sistem meneruskan permintaan tersebut ke REST API sistem Ma'had lama. Apabila NIM tidak ditemukan atau statusnya belum lulus, sistem menampilkan pesan yang sesuai. Jika NIM tervalidasi dan berstatus lulus, API sistem lama mengembalikan informasi kelulusan beserta tautan dokumen PDF, kemudian sistem baru menampilkan tombol untuk mengunduh sertifikat tersebut.

3.1.4.4 Perancangan Interaksi Objek (Sequence Diagram)

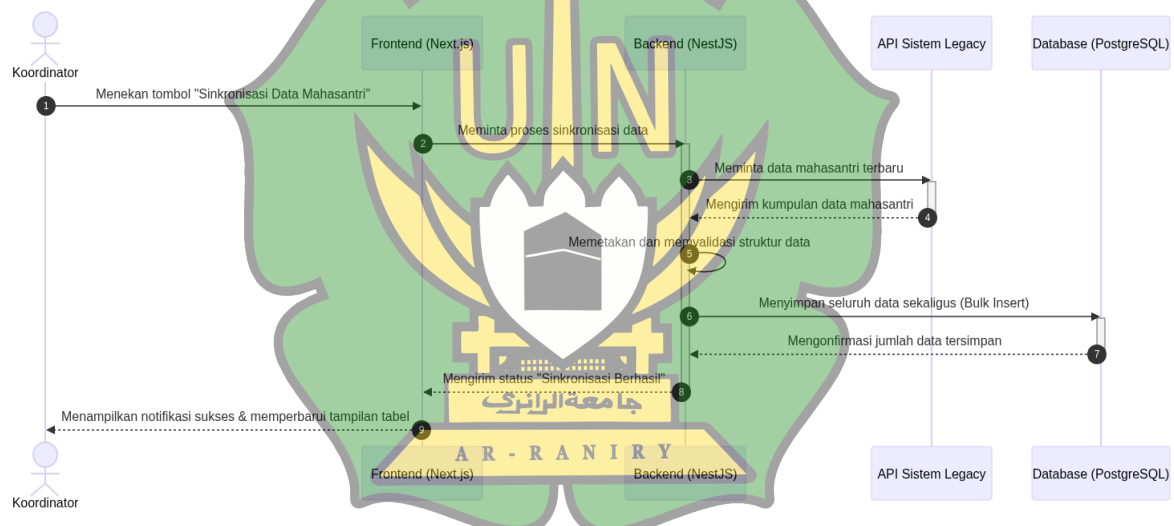
Sequence Diagram digunakan untuk memodelkan interaksi dinamis antar-objek di dalam sistem berdasarkan urutan waktu (*time-sequencing*). Pada arsitektur *Client-Server* terpisah yang diterapkan dalam penelitian ini, diagram ini memetakan aliran pertukaran data (*HTTP Requests/Responses*) antara antarmuka (Next.js),

backend (NestJS), basis data PostgreSQL, serta integrasi dengan sistem eksternal (*legacy system*).

Berikut adalah pemodelan Sequence Diagram untuk fungsionalitas inti pada Sistem Informasi Ma'had:

A. Sequence Diagram Sinkronisasi Data Mahasantri (API Pulling)

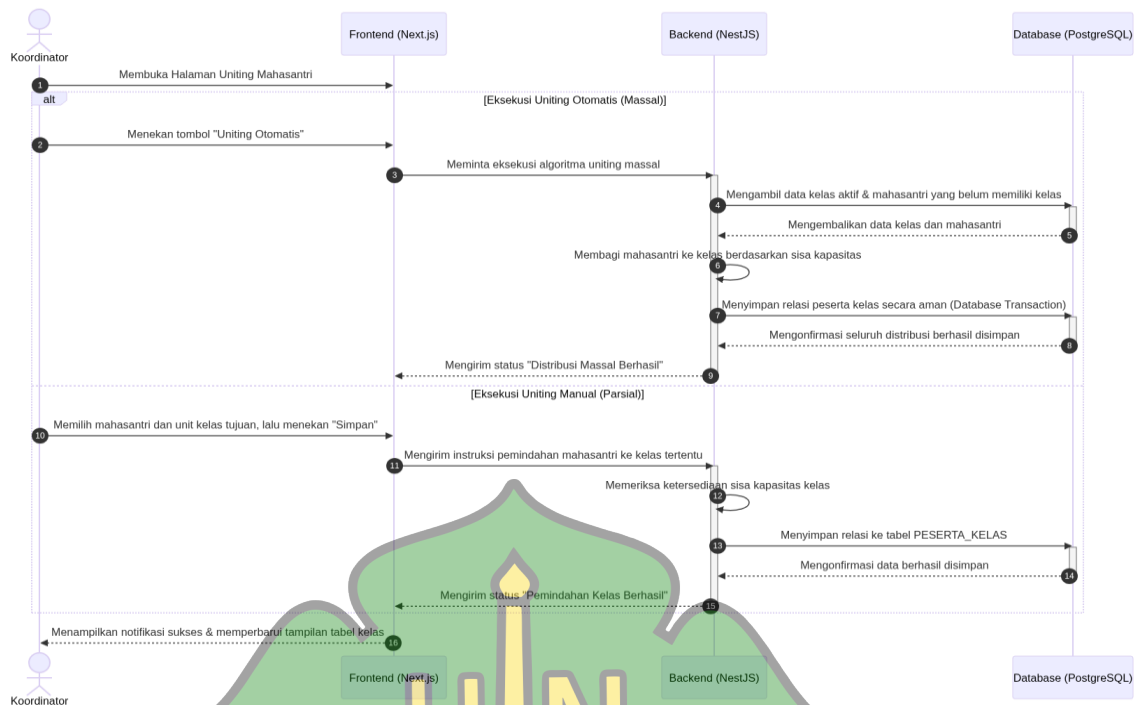
Diagram ini merepresentasikan alur penarikan data secara massal oleh Koordinator untuk menghindari entri data manual. Sistem menggunakan pendekatan *API Pulling*, di mana backend NestJS bertindak sebagai klien yang meminta data dari basis data sistem lama. Setelah *payload* JSON diterima, NestJS menggunakan *Data Transfer Object* (DTO) untuk memvalidasi struktur data sebelum mengeksekusi perintah *bulk insert* ke PostgreSQL lokal melalui Prisma ORM.



Gambar 3. 9 Sequence Diagram Sinkronisasi Data Mahasantri

B. Sequence Diagram Distribusi Kelas (Uniting Mahasantri)

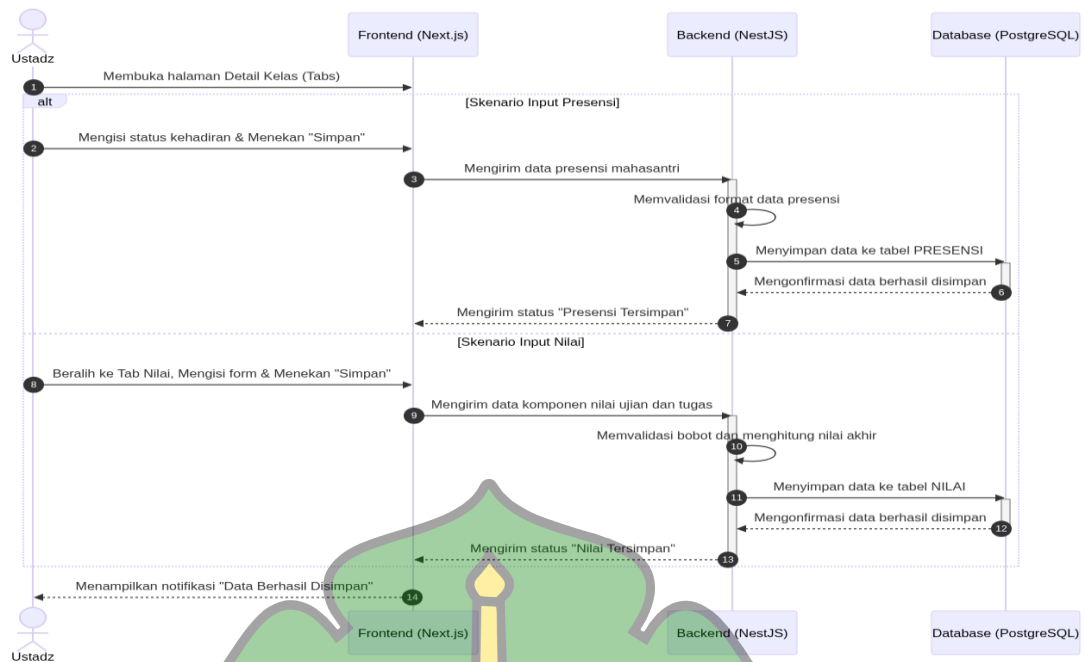
Diagram ini memodelkan proses pemetaan mahasantri ke dalam unit kelas operasional yang dikelola oleh Koordinator. Diagram menggunakan blok alternatif (*alt fragment*) untuk mengakomodasi dua mode distribusi. Pada mode otomatis (massal), backend NestJS akan mengeksekusi algoritma pembagian berdasarkan sisa kapasitas kelas dan membungkus kueri penyimpanannya ke dalam *Database Transaction* (Prisma) guna mencegah anomali atau kelebihan kapasitas (*overload*). Sedangkan pada mode manual, sistem memvalidasi kapasitas spesifik dari kelas yang dipilih sebelum mengeksekusi perintah *insert* tunggal ke dalam basis data.



Gambar 3. 10 *Sequence Diagram Distribusi Kelas*

C. Sequence Diagram Pengelolaan Data Akademik (Presensi dan Nilai)

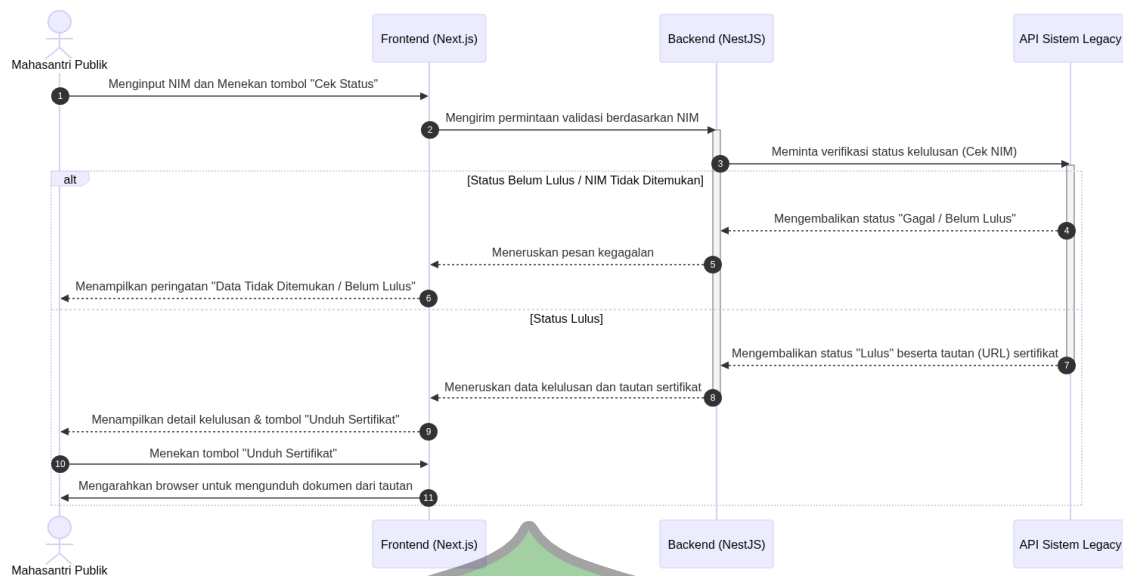
Diagram ini menggambarkan proses transaksional terpadu ketika Ustadz mengelola kelas yang diampunya. Karena antarmuka dirancang menggunakan sistem tabulasi (Tabs), Ustadz dapat melakukan input presensi kehadiran maupun komponen nilai dalam satu alur kerja. Diagram ini menggunakan blok alternatif (alt fragment) untuk memisahkan logika backend, di mana khusus untuk input nilai, NestJS akan mengeksekusi fungsi komputasi nilai_akhir secara otomatis sebelum menyimpannya ke dalam basis data guna mencegah anomali data.



Gambar 3. 11 Sequence Diagram Pengelolaan Data Akademik

D. Sequence Diagram Validasi Kelulusan Publik

Diagram ini memodelkan alur layanan mandiri (*self-service*) tanpa proses autentikasi. Untuk menjamin akurasi dan prinsip *Single Source of Truth*, sistem tidak memvalidasi kelulusan dari basis data lokal, melainkan meneruskan kueri Nomor Induk Mahasiswa (NIM) secara seketika (*real-time*) ke API sistem Ma'had lama. Jika divalidasi "Lulus", sistem lama akan mengembalikan tautan (*URL/Link*) dokumen PDF, yang kemudian diteruskan oleh NestJS ke antarmuka Next.js agar mahasiswa dapat langsung mengunduh sertifikatnya.



Gambar 3. 12 Sequence Diagram Validasi Kelulusan

3.1.4.5 Perancangan Basis Data

Sistem Informasi Ma'had Al-Jami'ah ini dirancang menggunakan pendekatan *Relational Database Management System* (RDBMS) berbasis PostgreSQL. Mengingat sistem ini beroperasi berdampingan dengan basis data akademik Ma'had yang sudah ada (*legacy system*), perancangan arsitektur data mengadopsi prinsip *Single Source of Truth*.

Struktur basis data ini terdiri dari beberapa entitas utama yang saling berelasi untuk mendukung proses pengelolaan kelas, presensi, serta penilaian.

Hasil dari identifikasi entitas beserta atribut yang digunakan dalam sistem (Kamus Data) dapat dilihat pada Tabel 3.5 sampai Tabel 3.13.

Tabel 3.5 Entitas Periode Akademik

Atribut	Tipe Data	Keterangan
id_periode	INT	Primary Key
nama_periode	VARCHAR	Nama periode akademik
is_active	BOOLEAN	Status periode aktif

Tabel 3.6 Entitas Mata Pelajaran

Atribut	Tipe Data	Keterangan
id_mapel	INT	Primary Key
nama_mapel	VARCHAR	Nama mata pelajaran

Tabel 3.7 Entitas Lokasi Gedung

Atribut	Tipe Data	Keterangan
id_lokasi	INT	Primary Key
nama_lokasi	VARCHAR	Nama gedung/lokasi
peruntukan_gender	ENUM	Peruntukan (Ikhwan/Akhwat)

Tabel 3.8 Entitas Ustadz

Atribut	Tipe Data	Keterangan
id_ustadz	VARCHAR	Primary Key
nama_ustadz	VARCHAR	Nama ustadz
username	VARCHAR	Username login
password_hash	VARCHAR	Password terenkripsi

Tabel 3.9 Entitas Kelas Unit

Atribut	Tipe Data	Keterangan
id_kelas	INT	Primary Key
nama_unit	VARCHAR	Nama kelas
kapasitas	INT	Kapasitas kelas
id_periode	INT	Foreign Key: Periode Akademik Id
id_mapel	INT	Foreign Key: Mata Pelajaran Id
id_lokasi	INT	Foreign Key: Lokasi Gedung Id
id_ustadz	VARCHAR	Foreign Key: Ustadz Id

Tabel 3. 10 Entitas Mahasantri

Atribut	Tipe Data	Keterangan
nim	VARCHAR	Primary Key
nama_lengkap	VARCHAR	Nama mahasiswa
jenis_kelamin	ENUM	L/P
prodi	VARCHAR	Prodi Mahasiswa
email	VARCHAR	Email Mahasiswa
angkatan	INT	Angkatan Mahasiswa
gelombang	VARCHAR	Gelombang Aktif Mahasiswa

Tabel 3. 11 Entitas Peserta Kelas

Atribut	Tipe Data	Keterangan
id_peserta	INT	Primary Key
nim	VARCHAR	Foreign Key: Mahasantri Id
id_kelas	INT	Foreign Key: Kelas Unit Id
status_lulus	ENUM	Lulus / Tidak Lulus

Tabel 3.12 Entitas Presensi

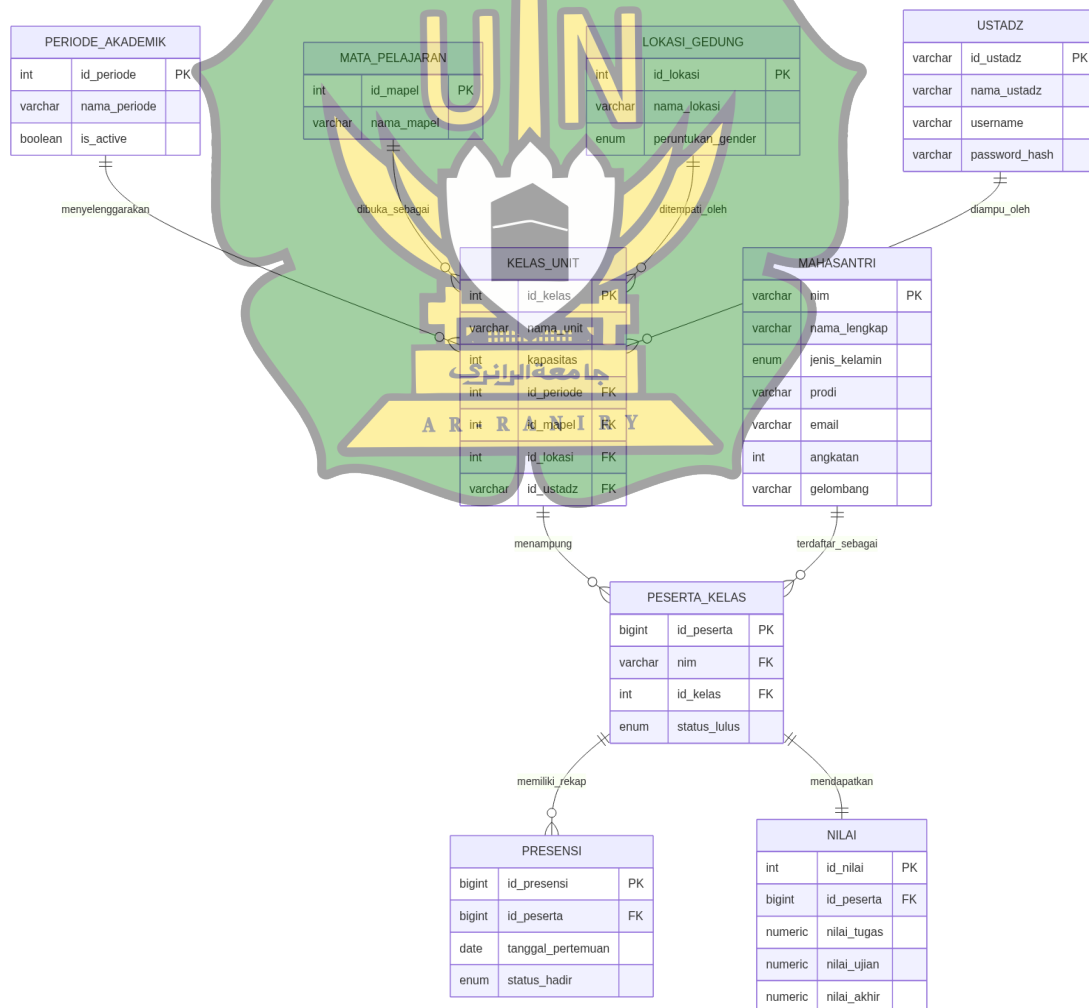
Atribut	Tipe Data	Keterangan
id_presensi	INT	Primary Key
id_peserta	INT	Foreign Key: Peserta Kelas Id
tanggal_pertemuan	DATE	Tanggal pertemuan
status_hadir	ENUM	Hadir/Izin/Alfa

Tabel 3. 13 Entitas Nilai

Atribut	Tipe Data	Keterangan
id_nilai	INT	Primary Key
id_peserta	INT	Foreign Key: Peserta Kelas Id
nilai_tugas	NUMERIC	Nilai tugas
nilai_ujian	NUMERIC	Nilai ujian
nilai_akhir	NUMERIC	Nilai akhir

Berdasarkan hasil identifikasi entitas dan pendefinisian atribut yang telah diuraikan secara terstruktur pada kamus data di atas, tahapan perancangan selanjutnya adalah merumuskan keterkaitan logis antar entitas tersebut. Untuk memodelkan struktur konseptual secara utuh serta memetakan kardinalitas relasi seperti batasan *one-to-many* dan relasi *many-to-many* antar entitas di dalam basis data, digunakanlah representasi visual berupa *Entity Relationship Diagram (ERD)*.

ERD ini dirancang sebagai bentuk translasi visual dari tabel-tabel fisik yang telah didefinisikan sebelumnya. Tujuannya adalah untuk memvalidasi integritas referensial sistem, guna memastikan bahwa seluruh aliran data operasional, mulai dari integrasi data mahasiswa, pembagian unit kelas (*uniting*), hingga rekapitulasi presensi dan nilai, terhubung dengan prinsip normalisasi yang tepat tanpa adanya anomali data. Keseluruhan pemodelan arsitektur data yang komprehensif ini dapat dilihat secara detail pada Gambar 3.13.



Gambar 3.13 Entity Relationship Diagram

3.1.4.6 Perancangan Antarmuka (*User Interface Design*)

Perancangan antarmuka merupakan tahapan visualisasi dari kebutuhan fungsional dan logika sistem yang telah didefinisikan sebelumnya. Pada tahap ini, rancangan layar dibuat dalam bentuk *wireframe* (kerangka garis) dengan fidelitas rendah hingga menengah (*low-to-mid fidelity*). Tujuannya adalah untuk memetakan tata letak (*layout*) informasi, navigasi, dan hierarki visual sebelum antarmuka benar-benar diimplementasikan ke dalam kode pemrograman.

Dalam penelitian ini, perancangan antarmuka difokuskan pada tiga jenis pengguna utama, yaitu Koordinator, Ustadz, dan Mahasiswa (layanan publik).

A. Perancangan Halaman Login

Halaman login berfungsi sebagai gerbang autentikasi utama bagi pengguna sebelum mengakses sistem. Antarmuka halaman ini dirancang menggunakan tata letak layar terbagi (*split-screen layout*) bergaya minimalis. Sisi kiri difungsikan sebagai area informasi statis, sedangkan sisi kanan berfokus pada panel interaksi autentikasi. Halaman ini mengakomodasi 2 model autentikasi, memberikan fleksibilitas bagi pengguna untuk masuk menggunakan kredensial kampus maupun kredensial lokal.



Gambar 3.14 Halaman Login – Wireframe

Berdasarkan Gambar 3.13 yang telah diberi penomoran, berikut merupakan penjelasan komponen-komponen sesuai dengan nomor yang tertera pada gambar.

1. **Area Informasi Statis dan Identitas Sistem:** Terletak pada sisi panel kiri, area ini memuat elemen visual statis seperti logo institusi, judul atau pesan sambutan utama, serta deskripsi singkat mengenai Sistem Informasi Ma'had guna memberikan konteks kepada pengguna.
2. **Area Login SSO UIN (Kredensial Kampus):** Terletak pada bagian atas panel kanan, area ini menyediakan tombol akses cepat bagi Koordinator atau pengguna yang terintegrasi dengan layanan Single Sign-On (SSO) milik universitas.
3. **Area Login Lokal:** Terletak pada bagian bawah panel kanan, area ini difungsikan untuk pengguna dengan kredensial sistem lokal (seperti Ustadz).

B. Perancangan Halaman Dashboard Koordinator

Dashboard Koordinator merupakan halaman utama yang merangkum keseluruhan aktivitas operasional dan manajerial di dalam sistem informasi Ma'had. Halaman ini berfungsi sebagai pusat kendali bagi administrator untuk memantau metrik penting melalui deretan kartu statistik ringkas, serta meninjau data terbaru melalui tabel informasi yang komprehensif. Desain tata letak dirancang menggunakan pendekatan antarmuka dashboard modern dengan memisahkan menu navigasi di sisi panel kiri (*sidebar*) dan area kerja di sisi kanan, sehingga memungkinkan koordinator untuk menganalisis data dan melakukan navigasi antarfitur dengan cepat, intuitif, dan efisien.



Gambar 3.15 Halaman Dashboard Koordinator – Wireframe

Berdasarkan Gambar 3.14 yang telah diberi penomoran, berikut merupakan penjelasan komponen-komponen sesuai dengan nomor yang tertera pada gambar.

1. **Area Menu Navigasi Sistem:** Terletak pada panel sisi kiri (*sidebar*), area ini memuat daftar menu utama yang memungkinkan Koordinator untuk berpindah dari satu modul ke modul lainnya (seperti manajemen kelas, data mahasiswa, dll) secara terpusat.
2. **Area Informasi Akun:** Berada di sudut kiri bawah pada panel navigasi, area ini menampilkan identitas atau profil singkat pengguna yang sedang aktif (login), yang umumnya dilengkapi dengan akses cepat menuju pengaturan atau tombol keluar (logout).
3. **Area Header:** Terletak membentang pada bagian atas area kerja, elemen ini menyediakan fungsionalitas pendukung operasional, seperti bilah pencarian cepat (*global search bar*) dan ikon panel notifikasi sistem.
4. **Area Statistik Metrik Utama:** Terletak di bagian atas area kerja utama, komponen ini menyajikan ringkasan data kuantitatif secara *real-time* melalui

deretan kartu (*cards*). Area ini memberikan wawasan instan kepada Koordinator mengenai metrik krusial, seperti total mahasantri aktif, jumlah pengajar (ustadz), maupun total ruang kelas.

5. **Area Tabel Informasi Aktivitas:** Mendominasi bagian tengah hingga bawah area kerja, tabel komprehensif ini berfungsi untuk menampilkan pratinjau data operasional atau rekam jejak aktivitas terbaru (misalnya riwayat pembuatan akun ustadz, penambahan gedung, dll) agar Koordinator dapat memantau pembaruan sistem tanpa harus masuk ke menu spesifik.

C. Perancangan Halaman Uniting Mahasantri

Halaman Uniting Mahasantri merupakan fitur inti (*core feature*) dari sistem ini yang difungsikan untuk memetakan dan mendistribusikan mahasantri ke dalam unit-unit kelas. Antarmuka halaman ini dirancang untuk menangani kompleksitas data dalam jumlah besar dengan tetap mempertahankan kemudahan navigasi. Berdasarkan desain *wireframe*, halaman ini menampilkan area aksi (*action banner*) yang menonjol untuk memicu proses uniting otomatis secara massal, diikuti dengan deretan kartu ringkasan status kapasitas, dan tabel data unit kelas yang komprehensif. Tabel ini dilengkapi dengan indikator visual berupa progress bar untuk memantau keterisian kapasitas masing-masing kelas, serta tombol aksi spesifik untuk pengelolaan kelas secara manual (CRUD parsial).



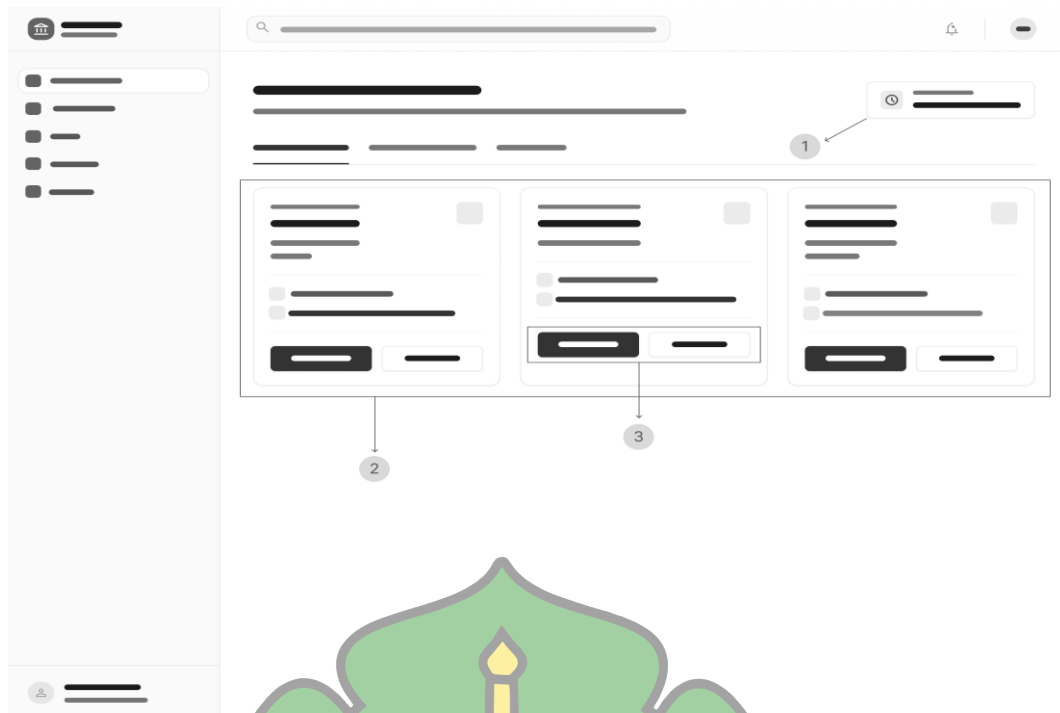
Gambar 3.16 Halaman Uniting Mahasantri – Wireframe

Berdasarkan Gambar 3.15 yang telah diberi penomoran, berikut merupakan penjelasan komponen-komponen sesuai dengan nomor yang tertera pada gambar.

1. **Area Tombol Aksi Tambahan:** Terletak di sudut kanan atas, area ini memuat tombol fungsional untuk membuat atau menambah data unit kelas baru secara manual, serta tombol opsi untuk mengeksport (*export*) data kelas yang sudah ada ke dalam format dokumen.
2. **Area Aksi Massal (*Action Banner*):** Merupakan panel area yang menonjol pada bagian atas, difungsikan khusus sebagai pemicu (*trigger*) utama untuk mengeksekusi algoritma uniting (pembagian kelas) otomatis secara massal oleh sistem.
3. **Area Kartu Statistik Ringkasan:** Menampilkan deretan kartu informasi yang merangkum metrik operasional secara real-time, seperti jumlah unit kelas yang sedang aktif dan ringkasan kapasitas keterisian kelas secara keseluruhan.
4. **Tabel Data Unit Kelas:** Merupakan area penyajian data utama yang menampilkan daftar seluruh kelas secara komprehensif, terstruktur, dan mudah dibaca oleh Koordinator.
5. **Indikator Visual (*Progress Bar*):** Terintegrasi langsung di dalam baris data tabel, indikator visual ini berfungsi untuk memantau persentase tingkat keterisian atau batas kuota kapasitas dari masing-masing unit kelas secara cepat dan intuitif.
6. **Tombol Aksi Parsial (*Action Menu*):** Berupa ikon opsi (titik tiga) yang terdapat pada setiap baris data. Tombol ini memberikan akses cepat kepada pengguna untuk melakukan tindakan pengelolaan pada kelas spesifik (CRUD parsial), seperti melihat detail, memperbarui data kelas, atau memindahkan mahasiswa secara manual.

D. Perancangan Halaman Dashboard Ustadz

Dashboard Ustadz dirancang sebagai portal utama bagi tenaga pengajar untuk mengelola kegiatan akademik dari kelas-kelas yang diampunya. Halaman ini memprioritaskan antarmuka yang bersih dan terfokus, sehingga ustadz dapat langsung mengidentifikasi kelas aktif dan melakukan administrasi akademik harian tanpa perlu melalui banyak tahapan navigasi yang membingungkan.



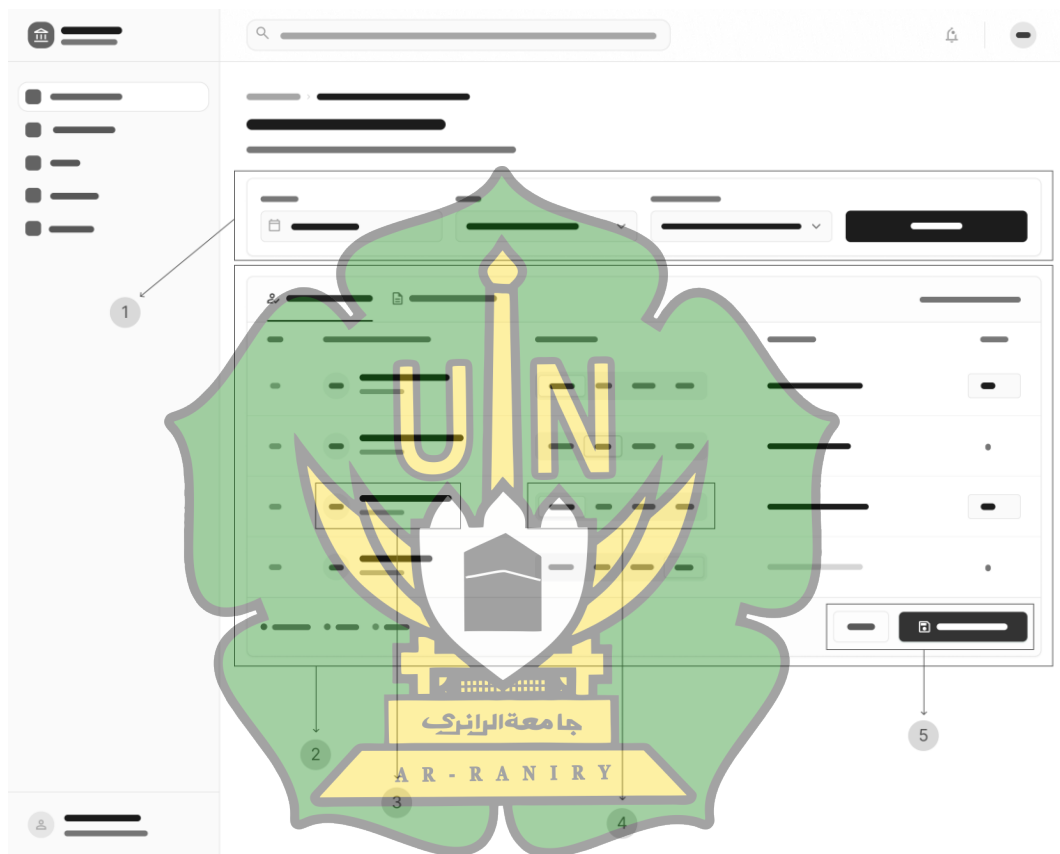
Gambar 3.17 Halaman Dashboard Ustadz – Wireframe

Berdasarkan Gambar 3.16 yang telah diberi penomoran, berikut merupakan penjelasan komponen-komponen sesuai dengan nomor yang tertera pada gambar.

1. **Informasi Waktu Saat Ini:** Terletak pada sudut kanan atas area kerja, elemen ini menampilkan informasi waktu dan tanggal aktual. Fitur ini berfungsi sebagai acuan langsung bagi ustadz untuk menyesuaikan ritme kerja dengan jadwal kegiatan kelas yang sedang atau akan berlangsung.
2. **Area Kartu Unit Kelas (*Grid Layout*):** Mendominasi bagian tengah layar, area ini menyajikan daftar seluruh kelas yang secara khusus ditugaskan kepada ustadz yang sedang login. Antarmuka dirancang menggunakan tata letak berbasis grid dengan komponen kartu (*class cards*), di mana setiap kartu merangkum informasi esensial kelas (seperti nama mata pelajaran, jumlah mahasiswa, dan indikator progres jadwal) secara padat dan terstruktur.
3. **Tombol Aksi Cepat (Absensi & Nilai):** Terletak secara ergonomis pada bagian bawah setiap kartu kelas, area ini menyediakan tombol pintasan (*shortcut*) fungsional. Tombol ini mengarahkan ustadz secara spesifik ke formulir input presensi kehadiran maupun input komponen penilaian untuk kelas tersebut dengan satu kali klik.

E. Perancangan Halaman Pengelolaan Absensi & Nilai

Halaman Pengelolaan Absensi dan Nilai dirancang sebagai antarmuka input data transaksional utama bagi ustadz. Untuk meningkatkan efisiensi alur kerja, halaman ini mengintegrasikan fungsi presensi dan penilaian ke dalam satu antarmuka yang sama menggunakan sistem tabulasi (*tabs*). Antarmuka dirancang sedemikian rupa untuk meminimalisasi potensi kesalahan (*human error*) dan mempercepat proses pendataan administrasi akademik harian.



Gambar 3.18 Halaman Pengelolaan Absensi & Nilai – Wireframe

Berdasarkan Gambar 3.17 yang telah diberi penomoran, berikut merupakan penjelasan komponen-komponen sesuai dengan nomor yang tertera pada gambar.

1. **Area Panel Filter:** Terletak di bagian atas area kerja, panel ini menyediakan parameter penyaringan (seperti pemilihan tanggal, pertemuan ke-, dan unit kelas) yang berfungsi untuk memuat baris data secara lebih spesifik dan relevan.
2. **Area Tabel Utama (Data Transaksional):** Merupakan area kerja sentral yang menampilkan daftar komprehensif seluruh mahasiswa di dalam kelas yang telah difilter, siap untuk dilakukan pengisian data kehadiran maupun nilai.

3. **Informasi Identitas Mahasantri:** Terletak pada kolom awal tabel, area ini menyajikan identitas dasar secara ringkas berupa Nama Lengkap dan Nomor Induk Mahasiswa (NIM) untuk memastikan akurasi penargetan data oleh ustadz.
4. **Komponen *Input Toggle Pill*:** Mengadopsi desain tombol pil (*pill-shaped buttons*), komponen interaktif ini digunakan untuk mengisi status kehadiran secara instan (Hadir, Sakit, Izin, Alfa). Pendekatan UI ini dipilih karena terbukti lebih cepat dan minim kesalahan ketimbang menggunakan menu *dropdown* konvensional.
5. **Area Tombol Aksi (Eksekusi):** Ditempatkan secara strategis pada sudut kanan bawah tabel agar mudah dijangkau setelah pengguna selesai meninjau data dari atas ke bawah. Area ini memuat tombol eksekusi untuk menyimpan seluruh perubahan (*bulk save*) ke dalam pangkalan data, serta tombol opsi untuk membatalkannya.

F. Perancangan Halaman Validasi Sertifikat (Publik)

Halaman Validasi Sertifikat dirancang sebagai antarmuka layanan publik yang dapat diakses secara terbuka oleh mahasantri maupun pihak luar yang berkepentingan, tanpa memerlukan proses autentikasi (*login*). Halaman ini mengusung pendekatan tata letak terpusat (*centered layout*) bergaya *landing page* minimalis guna meminimalisasi distraksi visual dan memfokuskan perhatian pengguna pada satu tugas utama.



Gambar 3.19 Halaman Validasi Sertifikat (Publik) – Wireframe

Berdasarkan Gambar 3.18 yang telah diberi penomoran, berikut merupakan penjelasan komponen-komponen sesuai dengan nomor yang tertera pada gambar.

1. **Kolom Input Pencarian:** Merupakan bidang teks (*text field*) utama yang dilengkapi dengan ikon kaca pembesar. Kolom ini difungsikan bagi pengguna untuk memasukkan Nomor Induk Mahasiswa (NIM) atau nomor identitas akademik lainnya secara spesifik.
2. **Tombol Eksekusi (Cek Status):** Ditempatkan tepat di bawah kolom pencarian, tombol tunggal ini berfungsi sebagai pemicu (*trigger*) untuk mengeksekusi proses validasi NIM ke dalam sistem. Jika data tervalidasi sebagai “Lulus”, sistem akan merespons dengan menampilkan tautan untuk mengunduh dokumen sertifikat digital secara mandiri.

3.1.5 Tahap Implementasi Sistem

Tahap implementasi sistem merupakan fase krusial dalam model pengembangan *Waterfall*, di mana seluruh hasil analisis kebutuhan dan perancangan logis meliputi desain arsitektur, *Use Case Diagram*, *Activity Diagram*,

struktur basis data, hingga *design Wireframe* diimplementasikan ke dalam bentuk kode program (*source code*) yang fungsional.

Pengembangan sistem informasi ini mengadopsi arsitektur *Client-Server* terpisah (*decoupled architecture*) untuk menjamin modularitas, skalabilitas, dan kemudahan pemeliharaan (*maintainability*). Sisi klien (*frontend*) akan direalisasikan menggunakan *framework* Next.js guna menyajikan antarmuka pengguna yang reaktif. Sementara itu, sisi server (*backend*) akan dibangun menggunakan *framework* NestJS yang dikonfigurasi untuk memproses komputasi algoritma dan logika bisnis. Komunikasi data antara kedua sisi tersebut dijumpai oleh arsitektur RESTful API, dengan Prisma ORM bertindak sebagai layer abstraksi untuk mengelola transaksi data menuju sistem manajemen basis data relasional PostgreSQL.

Secara spesifik, tahapan implementasi fungsionalitas sistem dibagi ke dalam empat modul utama.

3.1.5.1 Implementasi Autentikasi dan Otorisasi

Keamanan akses sistem diimplementasikan melalui penyediaan dua jalur autentikasi serta otorisasi berlapis berbasis peran. Implementasi pada modul ini difokuskan pada pengelolaan sesi pengguna dan pembatasan hak akses (*endpoint protection*):

1. **Dua Jalur Autentikasi (SSO dan Lokal):** Logika *backend* (NestJS) diprogram untuk mengakomodasi dua metode *login* yang berbeda berdasarkan peran. Untuk pengguna dengan peran Koordinator, autentikasi divalidasi melalui integrasi API *Single Sign-On* (SSO) milik UIN Ar-Raniry. Sedangkan untuk peran Ustadz, autentikasi menggunakan basis data lokal. Seluruh data kredensial kata sandi Ustadz tidak disimpan dalam bentuk teks langsung (*plaintext*), melainkan diproses menggunakan algoritma *hashing* kriptografis sehingga menghasilkan nilai hash yang kemudian disimpan ke dalam basis data PostgreSQL.
2. **Manajemen Sesi dan Otorisasi:** Pasca-autentikasi yang berhasil, NestJS akan men-generate token sesi berupa *JSON Web Token* (JWT) yang disisipkan ke dalam *header* peramban pengguna. Sistem mengimplementasikan protokol

Role-Based Access Control (RBAC) menggunakan *Middleware* dan *Guards* pada level *backend*. Mekanisme ini memastikan bahwa fungsi sistem diisolasi secara ketat berdasarkan peran pengguna (*role claim*). Sebagai contoh, *endpoint* API untuk manipulasi data master dan *uniting* kelas dikunci secara eksklusif dan hanya dapat diakses oleh token dengan klaim peran Koordinator, mencegah terjadinya eskalasi hak akses (*privilege escalation*) dari pengguna lain.

3.1.5.2 Implementasi Modul Dashboard Koordinator

Modul ini diimplementasikan sebagai pusat kendali administratif (*administrative backend*) yang ditujukan khusus bagi pengguna dengan peran Koordinator. Implementasi logika bisnis pada modul ini berfokus pada:

1. **Implementasi Sinkronisasi Data (API Pulling):** Sistem mengimplementasikan modul penarikan data otomatis untuk menghindari entri data mahasiswa secara manual. Logika *backend* (NestJS) diprogram untuk mengeksekusi *HTTP GET Request* menuju *endpoint* API milik sistem Ma'had lama (*legacy system*). Setelah menerima *payload* balasan dalam format JSON, sistem melakukan pemetaan struktur data menggunakan *Data Transfer Object* (DTO) untuk memvalidasi integritas nilai (seperti keunikan NIM), lalu menyimpannya secara *bulk insert* ke dalam tabel MAHASANTRI di basis data PostgreSQL lokal melalui perantara Prisma ORM.
2. **Manajemen Data Master:** Implementasi fungsi *Create, Read, Update, Delete* (CRUD) untuk mengelola entitas ruang, mata pelajaran, ustadz, dan pembentukan unit kelas baru.
3. **Algoritma Uniting:** Fungsionalitas pemetaan mahasiswa ke dalam kelas (*uniting*) diimplementasikan dalam dua mode eksekusi yaitu manual dan otomatis. Pada mode otomatis (*massal*), NestJS mengeksekusi algoritma perulangan (*looping*) yang membaca sisa kuota atribut kapasitas pada tabel KELAS_UNIT, lalu mendistribusikan daftar NIM mahasiswa yang belum memiliki kelas secara sekuensial. Untuk mencegah terjadinya anomali data (seperti kapasitas kelas yang *overload*), proses penyimpanan relasi ke entitas PESERTA_KELAS ini dibungkus di dalam sebuah *Database Transaction*. Pendekatan ini memastikan kepatuhan terhadap standar ACID (*Atomicity*,

Consistency, Isolation, Durability), di mana seluruh pemindahan ratusan data akan dibatalkan (*rollback*) jika terjadi satu saja kegagalan di tengah proses.

4. **Pemantauan Distribusi:** Penyajian antarmuka rekapitulasi untuk memonitor rasio keterisian kelas berdasarkan periode akademik yang aktif.

3.1.5.3 Implementasi Modul Dashboard Ustadz

Modul ini direalisasikan sebagai portal akademik operasional harian bagi tenaga pengajar (Ustadz). Implementasi pada tingkat antarmuka dan logika peladen difokuskan pada pemrosesan transaksi data akademik yang cepat, aman, dan meminimalkan beban kognitif pengguna:

1. **Implementasi Antarmuka Kelas:** Pada sisi klien (Next.js), *dashboard* utama diimplementasikan menggunakan tata letak berbasis *grid* yang merender komponen kartu (*class cards*) secara dinamis berdasarkan relasi data pengguna yang sedang melakukan *login*. Untuk efisiensi ruang kerja, halaman pengelolaan diimplementasikan menggunakan arsitektur komponen *Tabs*. Pendekatan *Client-Side Routing* ini memungkinkan Ustadz untuk berpindah secara mulus antara antarmuka *Input Presensi* dan *Input Nilai* tanpa memuat ulang halaman web (*page reload*).
2. **Pencatatan Presensi Terintegrasi:** Implementasi formulir presensi antarmuka menggunakan komponen *toggle group* berbentuk pil (*pill-shaped buttons*) untuk meminimalisasi potensi kesalahan klik (*missclick*). *Payload* data presensi kemudian dikirimkan ke peladen melalui metode *HTTP POST*. Backend NestJS akan menggunakan *Pipes* untuk melakukan validasi kerangka data (*payload validation*), guna memastikan parameter yang dikirimkan valid sebelum direkam secara persisten ke dalam tabel PRESENSI.
3. **Komputasi Nilai Akhir:** Sistem memuat *business logic* di sisi *backend* untuk mengalkulasi bobot nilai akhir secara otomatis sebelum proses penyimpanan (*commit*) ke dalam basis data.

3.1.5.4 Implementasi Layanan Publik Validasi Kelulusan

Sebagai bentuk transparansi informasi dan peningkatan efisiensi birokrasi administratif, sistem mengimplementasikan portal layanan publik (*public-facing portal*) yang dapat diakses melalui peramban web tanpa memerlukan sesi otentikasi

standar (*login*). Mengingat validitas status kelulusan mutlak bersumber dari basis data sistem lama, implementasi fitur mandiri (*self-service*) ini dirancang untuk berintegrasi langsung dengan sistem lama (*legacy system*) secara seketika (*real-time*):

1. **Antarmuka Pencarian Terpusat:** Pada sisi *frontend* (Next.js), halaman ini direalisasikan menggunakan tata letak terpusat (*centered layout*) bergaya *landing page* minimalis untuk memfokuskan interaksi pengguna. Formulir antarmuka diprogram khusus untuk menerima parameter *input* berupa Nomor Induk Mahasiswa (NIM) sebagai kata kunci pencarian identitas.
2. **Validasi *Real-time* melalui Integrasi API Eksternal:** Berbeda dengan modul *dashboard* yang mengakses basis data lokal (PostgreSQL), pencarian status kelulusan publik tidak membebani basis data sistem informasi Ma'had ini. Saat mahasiswa menekan tombol validasi, Next.js akan mengirimkan permintaan ke *backend* NestJS. Lapisan *backend* kemudian bertindak sebagai perantara yang langsung menembakkan kueri *HTTP GET Request* menuju *endpoint* API milik sistem Ma'had lama (*legacy system*). Apabila API sistem lama mengembalikan respons bahwa NIM tidak ditemukan atau status belum lulus, *endpoint* NestJS akan meneruskan kode status kesalahan tersebut untuk dirender oleh Next.js menjadi pesan peringatan pada layar pengguna.
3. **Penyajian Tautan Unduhan ~~Sertifikat~~ (*URL Forwarding*):** Berdasarkan arsitektur integrasi yang diterapkan, proses komputasi untuk merender fisik dokumen sertifikat (PDF) dikelola sepenuhnya oleh administrator pada sistem Ma'had lama. Oleh karena itu, fungsionalitas *backend* (NestJS) pada sistem baru difokuskan sebagai penerus data (*data forwarder*) yang efisien. Jika kueri validasi berhasil, *endpoint* API sistem lama akan mengembalikan *payload* JSON yang berisi detail kelulusan mahasiswa beserta sebuah atribut tautan (*URL/Link*) unik spesifik untuk dokumen sertifikat tersebut. NestJS kemudian meneruskan tautan ini ke antarmuka Next.js. Pada sisi klien, sistem menyajikan tombol aksi yang, saat ditekan oleh mahasiswa, akan mengarahkan peramban (*browser*) untuk langsung mengunduh berkas PDF tersebut dari peladen sistem lama berdasarkan tautan unik yang telah didapatkan.

3.1.6 Tahap Pengujian Sistem

Tahap pengujian sistem merupakan fase akhir dalam siklus pengembangan perangkat lunak model *Waterfall* sebelum sistem siap diimplementasikan atau diserahkan kepada pengguna akhir (*deployment*). Tahap pengujian dan evaluasi sistem merupakan fase untuk menilai kesesuaian fungsi utama sistem dengan spesifikasi kebutuhan serta mengevaluasi tingkat kebergunaan sistem berdasarkan pengalaman pengguna. Pada penelitian ini, evaluasi dilakukan menggunakan *Black-Box Testing* terhadap fungsi vital sistem dan *System Usability Scale* (SUS) terhadap persepsi kebergunaan sistem.

Untuk memastikan bahwa Sistem Informasi Ma'had yang dibangun telah memenuhi spesifikasi kebutuhan yang didefinisikan pada tahap analisis, Penelitian ini menerapkan dua pendekatan evaluasi, yaitu pengujian fungsional menggunakan *Black-Box Testing* dan evaluasi kebergunaan menggunakan *System Usability Scale* (SUS).

3.1.6.1 Pengujian Fungsional (*Black Box Testing*)

Pengujian fungsional dilakukan menggunakan metode *Black-Box Testing*. Metode ini merupakan teknik pengujian yang menyusun dan memilih kasus uji berdasarkan spesifikasi fungsional sistem tanpa mempertimbangkan struktur internal maupun kode sumber perangkat lunak. Dengan demikian, pengujian difokuskan pada kesesuaian antara data masukan, proses yang dijalankan, dan keluaran yang dihasilkan oleh sistem (International Software Testing Qualifications Board, 2024).

Skenario pengujian dipilih berdasarkan fungsi vital yang merepresentasikan alur operasional utama pada setiap modul dan memiliki risiko kegagalan yang signifikan. Pengujian diringkas menjadi sembilan skenario utama yang mencakup autentikasi dan otorisasi koordinator, sinkronisasi data, pengelolaan data master, otomasi uniting kelas, autentikasi dan akses kelas ustadz, pengelolaan presensi, pengelolaan nilai, pengecekan status kelulusan berdasarkan NIM, serta pengunduhan sertifikat. Setiap skenario dapat memuat beberapa variasi kondisi, seperti masukan valid, masukan tidak valid, pelanggaran hak akses, batasan aturan bisnis, dan kegagalan layanan eksternal.

Kasus uji disusun berdasarkan kebutuhan fungsional sistem dengan mempertimbangkan kondisi masukan valid, masukan tidak valid, batasan aturan bisnis, pembatasan hak akses, serta kegagalan komunikasi dengan layanan eksternal. Setiap kasus uji memiliki kondisi awal, data masukan, tindakan pengujian, dan hasil yang diharapkan. Hasil aktual yang ditampilkan oleh sistem kemudian dibandingkan dengan hasil yang diharapkan untuk menentukan status pengujian.

Tahapan pengujian fungsional dilakukan sebagai berikut:

1. Mengidentifikasi fungsi yang akan diuji berdasarkan kebutuhan fungsional sistem.
2. Menentukan kondisi awal dan data uji untuk setiap fungsi.
3. Menjalankan skenario melalui antarmuka sistem dan mengamati pertukaran request dan response pada tab Network di browser DevTools.
4. Mencatat hasil aktual yang diberikan oleh sistem.
5. Membandingkan hasil aktual dengan hasil yang diharapkan.
6. Menetapkan status lulus atau gagal pada setiap kasus uji.
7. Memperbaiki fungsi yang gagal dan melakukan pengujian ulang.

Suatu skenario dinyatakan lulus apabila seluruh variasi pengujian di dalam skenario tersebut menghasilkan keluaran yang sesuai dengan spesifikasi. Skenario dinyatakan gagal apabila satu atau lebih variasi menghasilkan keluaran yang tidak sesuai, menampilkan kesalahan yang tidak direncanakan, atau tidak dapat dijalankan. Bukti pengujian diperoleh melalui browser DevTools dengan memperhatikan endpoint, metode HTTP, status response, dan bagian response yang relevan. Informasi sensitif seperti kata sandi, token autentikasi, cookie, dan identitas pengguna disamarkan pada dokumentasi pengujian. Rincian instrumen dan skenario pengujian fungsional disajikan pada Subbab 3.3.2.

3.1.6.2 Pengujian Penerimaan Pengguna (*System Usability Scale*)

Setelah fungsi utama sistem dinyatakan dapat berjalan, penelitian dilanjutkan dengan evaluasi usability menggunakan System Usability Scale (SUS).

SUS merupakan instrumen yang terdiri atas sepuluh pernyataan untuk memperoleh penilaian subjektif pengguna terhadap usability sistem secara keseluruhan (Brooke, 1996). Penelitian ini menggunakan SUS versi Bahasa Indonesia yang diadaptasi dan diuji reliabilitasnya oleh Sharfina & Santoso (2016).

Evaluasi dilakukan terhadap tiga kelompok pengguna, yaitu koordinator, ustadz, dan mahasantri. Ketiga kelompok tersebut dilibatkan karena masing-masing berinteraksi dengan antarmuka dan fungsi yang berbeda. Koordinator menggunakan dashboard untuk melakukan sinkronisasi data, mengelola data operasional, menjalankan proses uniting, dan melihat rekapitulasi distribusi kelas. Ustadz menggunakan dashboard untuk melihat kelas yang diampu serta mengelola presensi dan nilai. Mahasantri menggunakan layanan publik untuk mengecek status kelulusan dan mengunduh sertifikat berdasarkan NIM.

Sebelum mengisi kuesioner SUS, responden diminta menjalankan skenario penggunaan sesuai dengan perannya. Setelah seluruh tugas selesai, responden langsung mengisi sepuluh pernyataan SUS sebelum dilakukan diskusi atau wawancara lanjutan. Setiap pernyataan dinilai menggunakan skala Likert lima tingkat, yaitu 1 untuk Sangat Tidak Setuju, 2 untuk Tidak Setuju, 3 untuk Netral, 4 untuk Setuju, dan 5 untuk Sangat Setuju.

Hasil SUS dianalisis untuk setiap kelompok pengguna. Pemisahan tersebut diperlukan karena koordinator, ustadz, dan mahasantri tidak menggunakan fungsi dan antarmuka yang sama. Nilai gabungan seluruh responden hanya digunakan sebagai informasi tambahan dan tidak dijadikan satu-satunya dasar penarikan kesimpulan. Skenario penggunaan dan kuesioner SUS disajikan pada Subbab 3.3.3, sedangkan teknik perhitungannya dijelaskan pada Subbab 3.4.2.

Dalam penelitian ini, istilah penerimaan pada hasil SUS dibatasi pada penerimaan terhadap usability atau persepsi kemudahan penggunaan sistem. Hasil tersebut tidak diartikan sebagai penerimaan teknologi secara luas sebagaimana diukur menggunakan Technology Acceptance Model atau Unified Theory of Acceptance and Use of Technology.

3.2 Populasi dan Sampel

Penelitian ini melibatkan informan pada tahap analisis kebutuhan dan responden pada tahap evaluasi kebergunaan sistem. Informan penelitian merupakan pihak yang memahami proses pengelolaan akademik dan administrasi di Ma'had Al-Jami'ah UIN Ar-Raniry Banda Aceh, terutama proses pemetaan kelas (*uniting*), pengelolaan presensi dan nilai, serta pelayanan informasi kelulusan dan sertifikat. Informan dipilih secara purposif berdasarkan keterlibatan dan pengetahuannya terhadap proses bisnis yang diteliti.

Populasi pengguna internal yang akan menggunakan dashboard sistem berjumlah 11 orang, terdiri atas 2 koordinator dan 9 ustadz/ustadzah. Selain pengguna internal tersebut, evaluasi SUS juga melibatkan mahasiswa sebagai pengguna sasaran layanan publik. Kelompok pengguna yang dievaluasi adalah sebagai berikut:

1. Koordinator Ma'had sebanyak 2 orang sebagai pengguna dashboard koordinator untuk melakukan sinkronisasi data, mengelola data operasional, melakukan *uniting* kelas, dan melihat rekapitulasi distribusi kelas.
2. Ustadz/ustadzah sebanyak 9 orang sebagai pengguna dashboard ustadz untuk melihat kelas yang diampu serta mengelola data presensi dan nilai mahasiswa.
3. Mahasiswa sebagai pengguna layanan publik untuk melakukan pengecekan status kelulusan dan pengunduhan sertifikat berdasarkan Nomor Induk Mahasiswa (NIM).

Penentuan responden dibedakan berdasarkan karakteristik populasinya. Untuk pengguna internal Ma'had digunakan total sampling atau sampel jenuh karena seluruh populasi yang berjumlah 11 orang dilibatkan, yaitu 2 koordinator dan 9 ustadz/ustadzah. Untuk kelompok mahasiswa digunakan *non-probability sampling* dengan pendekatan *purposive sampling* karena responden harus memenuhi kriteria yang sesuai dengan konteks penggunaan layanan pengecekan kelulusan dan sertifikat.

Kriteria responden dalam penelitian ini adalah sebagai berikut:

1. Koordinator yang terlibat dalam pengelolaan akademik atau administratif Ma'had.
2. Ustadz/ustadzah yang aktif mengajar dan memiliki tanggung jawab terhadap pengelolaan presensi atau nilai mahasiswa.
3. Mahasiswa yang memiliki NIM dan pernah mengikuti program Ma'had sehingga memahami konteks layanan pengecekan status kelulusan dan pengunduhan sertifikat.
4. Bersedia mengikuti skenario penggunaan sistem sesuai dengan perannya.
5. Menyelesaikan seluruh tugas pada skenario penggunaan sistem sesuai dengan perannya sebelum mengisi kuesioner *System Usability Scale* (SUS).
6. Mengisi seluruh pernyataan pada kuesioner SUS secara lengkap.

Responden evaluasi SUS direncanakan berjumlah 31 orang, terdiri atas seluruh 2 koordinator dan 9 ustadz/ustadzah sebagai pengguna internal, serta 20 mahasiswa yang dipilih secara purposif. Setiap kelompok diberikan skenario penggunaan yang berbeda sesuai dengan hak akses dan fungsi sistem yang digunakannya.

Penentuan 20 mahasiswa sebagai responden dilakukan untuk mengevaluasi usability modul layanan publik setelah diimplementasikan dalam lingkup penelitian dan sebelum diterapkan secara operasional secara penuh. Faulkner (2003) menunjukkan bahwa dalam konteks pengujian usability terstruktur untuk menemukan permasalahan penggunaan, kelompok berukuran 20 pengguna menemukan sedikitnya 95% dan rata-rata 98,40% dari keseluruhan masalah usability yang telah diketahui. Sementara itu, Tullis & Stetson (2004) menemukan bahwa dalam konteks perbandingan dua website, penggunaan SUS dengan ukuran sampel 12 menghasilkan kesimpulan yang sama dengan keseluruhan data pada sedikitnya 90% pengambilan sampel ulang. Temuan tersebut digunakan sebagai referensi metodologis pendukung dalam menetapkan 20 responden untuk evaluasi usability pada penelitian ini, bukan sebagai perhitungan statistik untuk menggeneralisasikan hasil terhadap seluruh populasi mahasiswa. Responden

mahasantri dipilih secara purposive berdasarkan kesesuaiannya dengan karakteristik pengguna sasaran modul layanan publik.

Pengujian fungsional menggunakan Black-Box Testing tidak menggunakan populasi dan sampel responden. Unit yang diperiksa dalam pengujian tersebut adalah skenario atau kasus uji berdasarkan kebutuhan fungsional sistem.

3.3 Instrumen Penelitian

Instrumen penelitian digunakan untuk memperoleh data secara sistematis pada tahap analisis kebutuhan, pengujian fungsional, dan evaluasi kebergunaan sistem. Instrumen yang digunakan meliputi pedoman observasi, pedoman wawancara, lembar pengujian *Black-Box*, skenario penggunaan sistem, dan kuesioner *System Usability Scale* (SUS).

3.3.1 Pedoman Observasi dan Wawancara

Pedoman observasi digunakan untuk mencatat kondisi dan proses operasional yang sedang berlangsung di lingkungan Ma'had Al-Jami'ah. Aspek yang diamati meliputi:

1. Proses penarikan dan pengelolaan data identitas mahasantri.
2. Proses pembentukan dan pemetaan mahasantri ke dalam kelas.
3. Proses pencatatan presensi dan nilai oleh Ustadz.
4. Proses rekapitulasi data akademik.
5. Proses pengecekan status kelulusan dan pelayanan sertifikat.
6. Kendala, pengulangan pekerjaan, serta potensi kesalahan pada proses yang sedang berjalan.

Pedoman wawancara disusun dalam bentuk pertanyaan semi-terstruktur. Pertanyaan diarahkan untuk memperoleh informasi mengenai proses bisnis yang berjalan, permasalahan yang dihadapi, kebutuhan setiap pengguna, pembagian hak akses, kebutuhan integrasi dengan sistem lama, dan hasil yang diharapkan dari sistem baru. Bentuk semi-terstruktur memungkinkan peneliti mengembangkan pertanyaan lanjutan sesuai dengan jawaban informan.

3.3.2 Lembar Pengujian Fungsional

Instrumen pengujian fungsional berupa lembar pengujian Black-Box yang digunakan untuk mencatat pelaksanaan pengujian terhadap fungsi vital sistem. Pada tahap perencanaan, lembar pengujian memuat kode skenario, fungsi yang diuji, variasi atau kondisi pengujian, dan hasil yang diharapkan. Pada tahap pelaksanaan, lembar tersebut dilengkapi dengan hasil aktual, status pengujian, dan referensi bukti pengujian. Tabel 3.14 menyajikan sembilan skenario fungsi utama yang digunakan sebagai dasar pelaksanaan pengujian.

Bukti pengujian didokumentasikan melalui tab Network pada browser DevTools. Informasi yang dicatat meliputi endpoint yang diakses, metode HTTP, status response, dan bagian response yang relevan dengan hasil pengujian. DevTools hanya digunakan sebagai alat untuk mengamati keluaran eksternal sistem dan tidak digunakan untuk memeriksa struktur internal atau kode sumber aplikasi.

Tabel 3. 14 Skenario Pengujian Fungsionalitas Menggunakan Black-Box Testing

Kode	Fungsi yang diuji	Variasi/kondisi pengujian	Hasil yang diharapkan
BB-K01	Autentikasi dan otorisasi koordinator	Akun SSO koordinator valid serta akun tanpa peran koordinator	Akun yang valid berhasil membentuk sesi dan mengakses dashboard koordinator, sedangkan akun yang tidak memiliki hak akses ditolak.
BB-K02	Sinkronisasi data mahasiswa	Data tersedia, sinkronisasi berulang, dan layanan sumber data tidak tersedia	Data berhasil ditarik secara read-only, tidak terjadi duplikasi, dan kegagalan layanan ditangani dengan pesan yang sesuai.
BB-K03	Pengelolaan data master	Data valid, data tidak lengkap, dan data duplikat	Data valid berhasil disimpan atau diperbarui, sedangkan data yang tidak memenuhi ketentuan ditolak.
BB-K04	Otomasi uniting kelas	Peserta memenuhi ketentuan serta kapasitas kelas terbatas	Peserta dialokasikan sesuai aturan dan sistem tidak menempatkan peserta melebihi kapasitas kelas.
BB-U01	Autentikasi dan akses kelas ustadz	Akun ustadz aktif, akun tidak aktif, dan	Ustadz aktif dapat mengakses kelas yang diampu, sedangkan akun

Kode	Fungsi yang diuji	Variasi/kondisi pengujian	Hasil yang diharapkan
		akses terhadap kelas ustadz lain	tidak aktif dan akses terhadap kelas lain ditolak.
BB-U02	Pengelolaan presensi	Data presensi valid dan data tidak lengkap/tidak sesuai	Presensi valid berhasil disimpan dan ditampilkan kembali, sedangkan data yang tidak valid ditolak.
BB-U03	Pengelolaan nilai	Nilai dalam rentang yang diizinkan dan nilai di luar rentang	Nilai valid berhasil disimpan, sedangkan nilai di luar ketentuan ditolak.
BB-M01	Pengecekan status kelulusan berdasarkan NIM	NIM lulus, belum lulus/tidak ditemukan, format NIM tidak valid, dan layanan sumber data tidak tersedia	Sistem menampilkan status yang sesuai tanpa mengungkapkan data yang tidak diperlukan serta menampilkan pesan kegagalan yang sesuai ketika layanan tidak tersedia.
BB-M02	Pengunduhan sertifikat	NIM dinyatakan lulus dan sertifikat tersedia serta sertifikat tidak tersedia	Sertifikat yang tersedia dapat diunduh, sedangkan permintaan yang tidak memenuhi ketentuan ditolak dengan pesan yang sesuai.

3.3.3 Skenario Penggunaan dan Kuesioner SUS

Sebelum mengisi kuesioner SUS, seluruh 2 koordinator, 9 ustadz/ustadzah, dan 20 mahasiswa menjalankan skenario penggunaan sesuai dengan perannya. Skenario tersebut digunakan untuk memastikan bahwa penilaian diberikan setelah responden memperoleh pengalaman langsung menggunakan fungsi yang relevan.

Tabel 3. 15 Skenario Penggunaan Koordinator

Kode	Tugas
SK-01	Masuk ke dalam sistem menggunakan akun koordinator yang disediakan
SK-02	Membuka dashboard dan memeriksa ringkasan data operasional
SK-03	Menjalankan sinkronisasi data mahasiswa dan membaca ringkasan hasilnya
SK-04	Mengelola salah satu data master yang diperlukan
SK-05	Menjalankan proses uniting dan memeriksa peserta yang berhasil dialokasikan atau dilewati
SK-06	Membuka rekap distribusi kelas berdasarkan filter yang ditentukan

Tabel 3. 16 Skenario Penggunaan Ustadz

Kode	Tugas
SU-01	Masuk ke dalam sistem menggunakan akun ustadz
SU-02	Membuka dashboard dan memilih salah satu kelas yang diampu
SU-03	Membuka pertemuan dan mengisi presensi mahasiswa
SU-04	Membuka halaman penilaian dan memasukkan nilai sesuai komponen yang tersedia
SU-05	Memeriksa kembali data presensi dan nilai yang telah disimpan

Tabel 3. 17 Skenario Penggunaan Mahasiswa

Kode	Tugas
SM-01	Membuka halaman pengecekan status kelulusan
SM-02	Memasukkan NIM uji yang telah disediakan
SM-03	Menjalankan pengecekan dan membaca informasi status kelulusan
SM-04	Mengunduh sertifikat apabila data dinyatakan lulus dan sertifikat tersedia

Data NIM yang digunakan dalam skenario dapat berupa data uji atau data milik responden sendiri atas persetujuan yang bersangkutan. Responden tidak diberikan data pribadi milik pengguna lain. Untuk skenario utama mahasiswa, peneliti menyiapkan NIM uji yang berstatus lulus dan memiliki tautan sertifikat agar seluruh alur utama dapat dijalankan.

Selama pelaksanaan skenario, peneliti memberikan petunjuk awal mengenai tujuan pengujian, tetapi tidak mengarahkan setiap langkah secara berlebihan. Bantuan yang diberikan selama pengujian dicatat karena kebutuhan bantuan dapat memengaruhi persepsi responden terhadap kemudahan penggunaan sistem.

Setelah menyelesaikan skenario, responden mengisi kuesioner SUS versi Bahasa Indonesia yang diadaptasi oleh Sharfina dan Santoso (2016). Pilihan jawaban terdiri atas:

1. Sangat Tidak Setuju,
2. Tidak Setuju,
3. Netral,
4. Setuju, dan
5. Sangat Setuju.

Untuk detail lengkap mengenai instrumen *System Usability Scale*(SUS) yang digunakan bisa dilihat pada tabel 3.18.

Tabel 3. 18 Instrumen System Usability Scale

No.	Pernyataan	Jenis
1	Saya berpikir akan menggunakan sistem ini lagi.	Positif
2	Saya merasa sistem ini rumit untuk digunakan.	Negatif
3	Saya merasa sistem ini mudah digunakan.	Positif
4	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini.	Negatif
5	Saya merasa fitur-fitur sistem ini berjalan dengan semestinya.	Positif
6	Saya merasa ada banyak hal yang tidak konsisten atau tidak serasi pada sistem ini.	Negatif
7	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat.	Positif
8	Saya merasa sistem ini membingungkan.	Negatif
9	Saya merasa tidak ada hambatan dalam menggunakan sistem ini.	Positif
10	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.	Negatif

Istilah “sistem” pada setiap pernyataan merujuk pada modul yang digunakan oleh responden sesuai dengan perannya. Pernyataan ganjil merupakan pernyataan positif, sedangkan pernyataan genap merupakan pernyataan negatif. Seluruh butir harus dijawab. Apabila responden tidak dapat menentukan jawaban, responden dapat memilih nilai tengah atau Netral.

3.4 Teknik Analisis Data

Data penelitian dianalisis menggunakan pendekatan deskriptif kualitatif dan deskriptif kuantitatif. Analisis deskriptif kualitatif diterapkan terhadap data observasi, wawancara, catatan pelaksanaan skenario, dan tanggapan terbuka responden. Analisis deskriptif kuantitatif diterapkan terhadap skor kuesioner SUS.

Data observasi dan wawancara direduksi dengan memilih informasi yang berhubungan dengan permasalahan penelitian. Informasi tersebut dikelompokkan berdasarkan proses sinkronisasi data, uniting kelas, pengelolaan presensi dan nilai, pengecekan kelulusan, kebutuhan pengguna, kendala operasional, dan batasan integrasi. Temuan observasi dibandingkan dengan hasil wawancara untuk memeriksa kesesuaian informasi. Hasil yang telah dikonfirmasi digunakan sebagai dasar penyusunan kebutuhan sistem.

3.4.1 Analisis Hasil Black-Box Testing

Hasil Black-Box Testing dianalisis dengan membandingkan hasil aktual dengan hasil yang diharapkan pada setiap variasi pengujian. Status akhir diberikan pada tingkat skenario fungsi utama. Suatu skenario dinyatakan lulus apabila seluruh variasi di dalamnya memperoleh hasil yang sesuai dengan spesifikasi.

1. **Lulus**, apabila seluruh variasi pengujian dalam satu skenario menghasilkan keluaran yang sesuai dengan hasil yang diharapkan.
2. **Gagal**, apabila satu atau lebih variasi pengujian menghasilkan keluaran yang tidak sesuai, mengalami kesalahan yang tidak direncanakan, atau tidak dapat dijalankan.

Tingkat kesesuaian fungsional dapat dirangkum menggunakan rumus:

$$\text{Tingkat Kesesuaian Fungsi Utama} = \frac{\text{Jumlah Skenario Lulus}}{\text{Jumlah Seluruh Skenario}} \times 100\%$$

Persentase tersebut hanya digunakan sebagai ringkasan keterlaksanaan kasus uji dan tidak digunakan sebagai ukuran usability atau tingkat kepuasan pengguna. Apabila terdapat kasus uji berstatus gagal, fungsi terkait diperbaiki dan diuji kembali. Sistem dinyatakan memenuhi spesifikasi fungsional setelah seluruh fungsi utama dan kasus uji kritis menghasilkan keluaran yang sesuai dengan spesifikasi.

3.4.2 Analisis System Usability Scale

Jawaban kuesioner SUS dihitung untuk setiap responden berdasarkan prosedur yang dikembangkan Brooke (1995). Setiap jawaban memiliki nilai antara 1 sampai 5. Nilai kontribusi setiap pernyataan dihitung menggunakan ketentuan berikut:

1. Untuk pernyataan ganjil, yaitu nomor 1, 3, 5, 7, dan 9, nilai kontribusi diperoleh dengan mengurangi nilai jawaban dengan 1.
2. Untuk pernyataan genap, yaitu nomor 2, 4, 6, 8, dan 10, nilai kontribusi diperoleh dengan mengurangi angka 5 dengan nilai jawaban.
3. Seluruh nilai kontribusi dijumlahkan.

4. Jumlah nilai kontribusi dikalikan dengan 2,5 untuk menghasilkan skor SUS pada rentang 0 sampai 100.

Nilai kontribusi jawaban dapat dirumuskan sebagai berikut:

$C_{ij} = X_{ij} - 1$, untuk pernyataan ganjil

$C_{ij} = 5 - X_{ij}$, untuk pernyataan genap

Skor SUS setiap responden dihitung menggunakan rumus:

$$SUS_i = 2,5 \times \sum C_{ij}$$

Keterangan:

- SUS_i adalah skor SUS responden ke-i.
- C_{ij} adalah nilai kontribusi jawaban responden ke-i pada pernyataan ke-j.
- X_{ij} adalah nilai jawaban responden ke-i pada pernyataan ke-j.
- j adalah pernyataan ke-1 sampai ke-10.

Nilai rata-rata SUS setiap kelompok pengguna dihitung menggunakan rumus:

$$\text{Rata-rata SUS kelompok} = \sum \frac{SUS_i}{n}$$

Keterangan:

- SUS_i adalah skor SUS setiap responden dalam kelompok.
- n adalah jumlah responden dalam kelompok pengguna.

Skor SUS berada pada rentang 0 sampai 100, tetapi skor tersebut bukan persentase. Skor yang lebih tinggi menunjukkan persepsi usability yang lebih positif. Skor setiap butir juga tidak diinterpretasikan secara terpisah karena SUS menghasilkan ukuran komposit terhadap usability sistem secara keseluruhan.

Interpretasi skor rata-rata menggunakan acceptability ranges yang dikembangkan oleh Bangor, Kortum, dan Miller (2008).

Tabel 3. 19 Interpretasi Acceptability Ranges Skor SUS

Rentang skor	Interpretasi
Kurang dari 50	Not Acceptable
50 sampai kurang dari 70	Marginal
70 sampai 100	Acceptable

Hasil analisis disajikan secara terpisah untuk kelompok koordinator, ustadz, dan mahasantri. Untuk setiap kelompok disajikan jumlah responden, nilai rata-rata, median, nilai minimum, dan nilai maksimum. Pemisahan hasil dilakukan karena setiap kelompok mengevaluasi antarmuka dan fungsi yang berbeda. Nilai rata-rata seluruh responden dapat disajikan sebagai informasi tambahan setelah hasil setiap kelompok dijelaskan.

Hasil SUS dianalisis secara deskriptif dan tidak digunakan untuk melakukan generalisasi statistik terhadap seluruh populasi. Apabila suatu kelompok memperoleh skor rendah, tanggapan terbuka dan catatan pelaksanaan skenario digunakan untuk mengidentifikasi penyebabnya. Tanggapan tersebut dikelompokkan ke dalam tema, seperti kemudahan navigasi, kejelasan informasi, konsistensi antarmuka, pesan kesalahan, kebutuhan bantuan, dan kendala pelaksanaan tugas.



BAB 4

HASIL DAN PEMBAHASAN

Bab ini memaparkan hasil pengembangan Sistem Informasi Operasional Ma'had berbasis web dan pembahasannya berdasarkan ruang lingkup penelitian. Hasil pengembangan mencakup dashboard koordinator, dashboard ustadz, serta layanan publik untuk pengecekan status kelulusan dan pengunduhan sertifikat berdasarkan Nomor Induk Mahasiswa (NIM). Penyajian hasil difokuskan pada keterkaitan antara kebutuhan sistem, rancangan pada Bab III, dan fungsi yang telah diimplementasikan. Landing page serta Content Management System (CMS) tidak dibahas karena keduanya berada di luar batas penelitian ini.

Bab ini juga menyediakan struktur pelaporan pengujian fungsional menggunakan Black-Box Testing dan evaluasi kebergunaan menggunakan System Usability Scale (SUS). Pada saat penyusunan draf ini, hasil aktual pengujian lapangan dan jawaban kuesioner SUS belum tersedia secara final. Oleh karena itu, bagian yang memerlukan data empiris diberi penanda khusus dan tidak diisi dengan nilai asumsi. Penanda tersebut harus diganti menggunakan bukti pengujian serta data responden yang sebenarnya sebelum naskah dinyatakan final.

4.1 Gambaran Umum Hasil Pengembangan

Hasil penelitian berupa aplikasi web operasional yang menghubungkan pengguna dengan layanan akademik sesuai kewenangannya. Koordinator memperoleh fasilitas pengelolaan data dan kelas, ustadz memperoleh fasilitas untuk mengelola aktivitas akademik pada kelas yang diampu, sedangkan mahasiswa memperoleh layanan mandiri untuk memeriksa kelulusan dan mengakses sertifikat. Sistem baru tidak menggantikan Sistem Informasi Ma'had yang lama. Data identitas awal dan sertifikat tetap bersumber dari sistem lama, sedangkan sistem baru menyimpan keadaan operasional kelas, peserta, presensi, dan nilai pada basis data tersendiri.

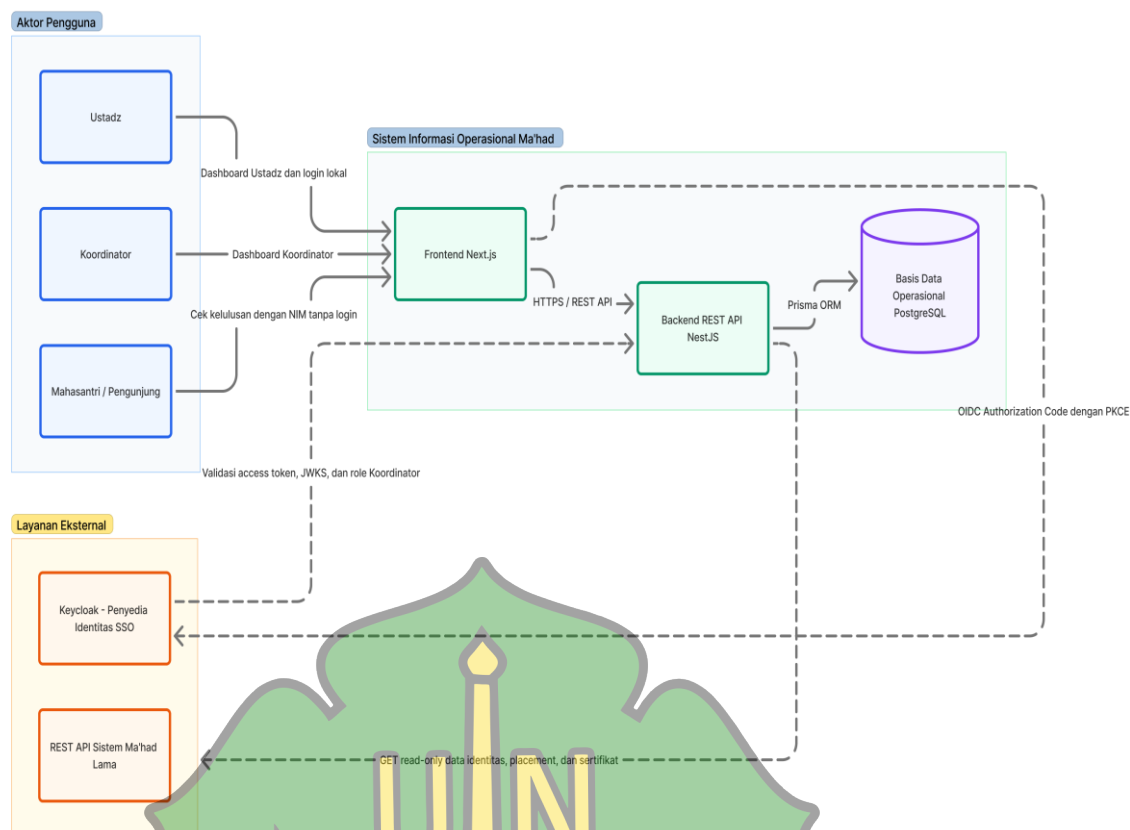
Tabel 4.1 Ruang Lingkup Hasil Implementasi

No.	Modul	Pengguna	Hasil Utama
1	Dashboard koordinator	Koordinator	Sinkronisasi data, pengelolaan data operasional, pembentukan kelas, uniting manual dan otomatis, serta rekap distribusi.
2	Dashboard ustadz	Ustadz	Daftar kelas yang diampu, pengisian presensi per pertemuan, pengisian nilai, dan peninjauan rekap kelas.
3	Layanan kelulusan publik	Mahasantri/pe ngunjung	Pengecekan status kelulusan dan akses unduh sertifikat berdasarkan NIM tanpa login.

4.1.1 Arsitektur Implementasi Sistem

Sistem menerapkan arsitektur client-server dengan pemisahan antara antarmuka, layanan aplikasi, basis data operasional, dan layanan eksternal. Antarmuka dibangun menggunakan Next.js dan berkomunikasi dengan backend melalui REST API. Backend dibangun menggunakan NestJS dengan adapter Fastify, sedangkan akses basis data dilakukan melalui Prisma ORM ke PostgreSQL. Pemisahan ini menempatkan backend sebagai batas keamanan dan pusat aturan bisnis, validasi pada antarmuka hanya berfungsi membantu pengalaman pengguna dan tidak menggantikan validasi di sisi server.

Dua layanan eksternal terlibat dalam alur sistem. Pertama, REST API Sistem Ma'had lama menjadi sumber data identitas, booking/check-in, placement, serta sertifikat. Akses terhadap sistem lama bersifat penarikan data dan tidak memberikan hak kepada sistem baru untuk mengubah data master sensitif pada sistem lama. Kedua, Keycloak digunakan sebagai penyedia identitas untuk autentikasi koordinator menggunakan OpenID Connect. Backend tetap menerbitkan JWT aplikasi setelah identitas dan peran koordinator dinyatakan valid. Susunan komponen dan aliran komunikasinya disajikan pada Gambar 4.1.



Gambar 4.1 Arsitektur Implementasi Sistem Informasi Operasional Ma’had

Pada implementasi ini, beberapa teknologi utama yang digunakan bisa dilihat pada tabel 4.2.

Tabel 4.2 Teknologi Utama pada Implementasi Sistem

Lapisan	Teknologi	Peran dalam Sistem
Frontend	Next.js 16, React 19, React Query 5, Tailwind CSS 4	Menyediakan antarmuka publik dan dashboard sesuai peran pengguna.
Backend	NestJS 11 dan Fastify 5	Menangani autentikasi, otorisasi, validasi, aturan bisnis, dan integrasi eksternal.
Akses data	Prisma ORM 6	Menghubungkan layanan backend dengan basis data relasional.
Basis data	PostgreSQL	Menyimpan data operasional kelas, peserta, presensi, nilai, dan log sinkronisasi.
Integrasi	REST API	Menarik data yang diperlukan dari Sistem Ma’had lama secara read-only.

Lapisan	Teknologi	Peran dalam Sistem
Identitas	Keycloak	Memvalidasi identitas koordinator sebelum pertukaran ke JWT aplikasi.

4.1.2 Struktur Direktori Implementasi

Struktur direktori implementasi digunakan untuk menunjukkan pemisahan tanggung jawab antara frontend dan backend serta pembagian modul berdasarkan fungsi bisnis. Struktur yang disajikan hanya mencakup direktori yang berhubungan langsung dengan ruang lingkup penelitian, yaitu dashboard koordinator, dashboard ustadz, sinkronisasi data, pengelolaan kelas, presensi, nilai, serta layanan publik kelulusan. Struktur direktori bisa dilihat pada gambar 4.2



Gambar 4.2 Struktur Direktori Modul Sistem dalam Lingkup Penelitian

Pada sisi backend, modul disusun berdasarkan fungsi bisnis, seperti autentikasi, akademik, mahasantri, kelas, presensi, nilai, sinkronisasi, dan kelulusan. Pada sisi frontend, halaman dipisahkan berdasarkan kelompok pengguna melalui direktori koordinator, ustadz, dan cek-kelulusan. Pemisahan tersebut membantu menjaga tanggung jawab setiap modul agar implementasi lebih mudah dipelihara, diuji, dan dikembangkan tanpa mencampurkan fungsi di luar batas penelitian.

4.1.3 Batas Implementasi Penelitian

Walaupun repositori pengembangan memuat komponen lain untuk kebutuhan situs Ma'had, unit analisis dan hasil yang dilaporkan pada penelitian ini dibatasi pada tiga modul pada Tabel 4.1.

Implementasi juga hanya berbentuk aplikasi web responsif dan tidak menghasilkan aplikasi native Android atau iOS. Ketergantungan pada layanan lama dipertahankan untuk menghindari duplikasi sumber kebenaran. Sistem baru menyimpan salinan operasional data yang diperlukan untuk proses kelas, tetapi status sertifikat tetap ditentukan oleh layanan lama.

4.2 Hasil Implementasi Sistem

Implementasi dilakukan dengan menerjemahkan kebutuhan fungsional menjadi halaman, endpoint, aturan otorisasi, dan model data. Uraian berikut menjelaskan hasil implementasi berdasarkan aktor dan alur kerja utama. Tampilan aktual pada setiap bagian perlu dilengkapi dengan tangkapan layar dari versi sistem yang digunakan pada saat pengujian agar bukti visual konsisten dengan versi aplikasi yang dievaluasi.

Selain tampilan antarmuka, bagian ini menyajikan potongan kode yang mewakili empat fungsi inti, yaitu sinkronisasi data, auto-uniting, pengolahan nilai, serta layanan kelulusan dan sertifikat. Pemilihan kode dibatasi sesuai dengan ruang lingkup penelitian.

4.2.1 Implementasi Autentikasi dan Otorisasi

Keamanan akses pada Sistem Informasi Operasional Ma'had diimplementasikan melalui dua mekanisme autentikasi yang disesuaikan dengan jenis pengguna. Koordinator menggunakan mekanisme Single Sign-On (SSO)

berbasis OpenID Connect (OIDC), sedangkan ustadz/ustadzah menggunakan autentikasi lokal berupa nama pengguna dan kata sandi. Setelah proses autentikasi berhasil, sistem menerapkan otorisasi berbasis peran atau *Role-Based Access Control* (RBAC) untuk membatasi akses pengguna terhadap fungsi dan sumber daya sistem.

Pada jalur autentikasi koordinator, Keycloak digunakan sebagai *Identity Provider* pada lingkungan pengujian integrasi SSO. Proses autentikasi menggunakan *Authorization Code Flow* dengan *Proof Key for Code Exchange* (PKCE) S256. Ketika koordinator memilih tombol masuk menggunakan SSO, frontend Next.js mengarahkan pengguna menuju halaman autentikasi Keycloak. Setelah identitas pengguna berhasil diverifikasi, Keycloak mengembalikan *authorization code* melalui alamat *callback* yang telah ditentukan. Frontend kemudian menukarkan kode tersebut untuk memperoleh *access token* dari Keycloak.

Access token Keycloak selanjutnya dikirimkan oleh frontend ke endpoint POST /api/auth/sso/login pada backend NestJS. Backend memvalidasi tanda tangan token menggunakan kunci publik dari *JSON Web Key Set* (JWKS), serta memeriksa *issuer*, masa berlaku token, identitas aplikasi atau *client ID*, dan peran pengguna. Sistem hanya menerima token yang memiliki peran KOORDINATOR. Apabila token tidak valid atau akun tidak mempunyai peran yang dipersyaratkan, proses login ditolak dan sistem mengembalikan respons tidak terotorisasi.

Setelah token Keycloak dinyatakan valid, backend mencocokkan identitas pengguna dengan data koordinator pada basis data lokal menggunakan identitas unik dari Keycloak. Data nama dan alamat surel koordinator dapat disimpan atau diperbarui melalui proses ini. Backend kemudian menerbitkan *JSON Web Token* (JWT) aplikasi yang digunakan sebagai sesi untuk mengakses dashboard koordinator. Dengan demikian, Keycloak bertanggung jawab memverifikasi identitas dan peran koordinator, sedangkan backend sistem bertanggung jawab membentuk sesi aplikasi serta membatasi akses terhadap fungsi operasional.

Untuk ustadz/ustadzah, autentikasi dilakukan menggunakan nama pengguna dan kata sandi yang tersimpan pada basis data operasional. Kata sandi tidak disimpan dalam bentuk teks biasa, tetapi dalam bentuk *hash* yang diverifikasi

ketika pengguna melakukan login. Sistem juga memeriksa status keaktifan akun sehingga akun yang telah dinonaktifkan tidak dapat digunakan untuk mengakses dashboard ustadz/ustadzah.

Otorisasi pada backend diimplementasikan menggunakan *guards* untuk memeriksa JWT dan peran pengguna pada setiap permintaan terhadap endpoint yang dilindungi. Endpoint pengelolaan data master, sinkronisasi data, dan proses *uniting* hanya dapat diakses oleh koordinator. Sementara itu, ustadz/ustadzah hanya dapat mengakses kelas yang menjadi tanggung jawabnya serta menjalankan fungsi pengelolaan presensi dan nilai. Mekanisme tersebut mencegah pengguna mengakses fungsi yang tidak sesuai dengan kewenangannya.

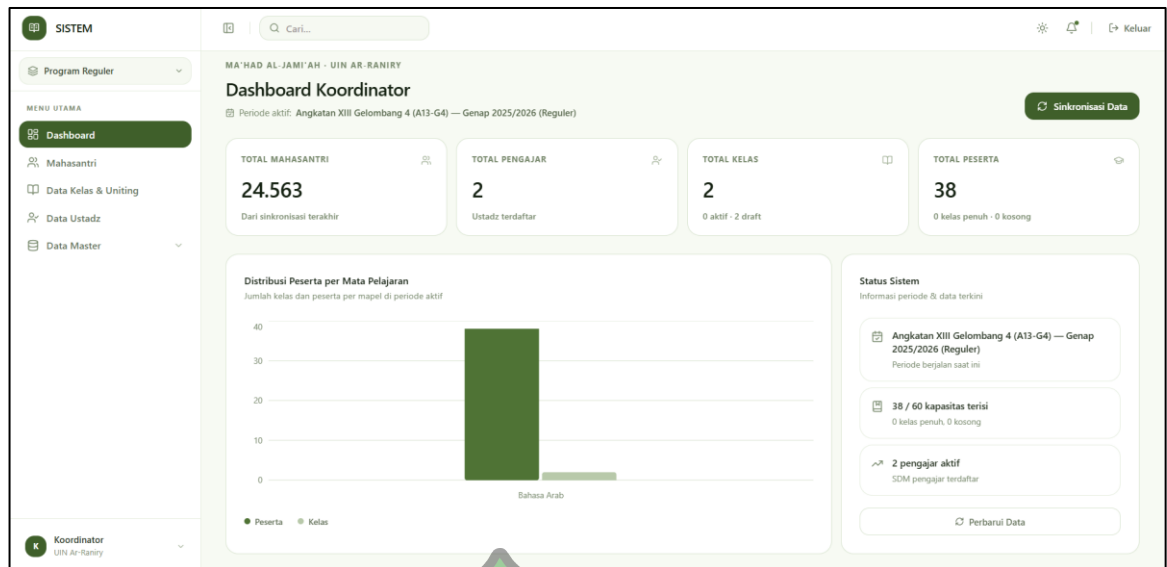
Dalam penelitian ini, layanan Keycloak digunakan untuk memvalidasi implementasi integrasi SSO koordinator dan bukan dinyatakan sebagai pengganti final layanan SSO resmi UIN Ar-Raniry. Konfigurasi penyedia identitas dapat diarahkan kepada layanan SSO resmi setelah informasi *realm*, *client*, peran, dan alamat *callback* diperoleh dari pihak pengelola layanan UIN Ar-Raniry. Tampilan jalur login koordinator dan ustadz disajikan pada Gambar 4.3.

The image shows a login interface for 'Ma'had UIN Ar-Raniry'. It is divided into two main sections. The left section, titled 'Masuk ke layanan Ma'had', prompts the user to 'Pilih jenis akun untuk melanjutkan ke halaman yang sesuai.' (Choose account type to continue to the appropriate page). It features three buttons: 'Mahasiswa' (Student), 'Pengelola' (Manager), and 'Ustadz' (Teacher). Below these, there are two login options: 'Masuk sebagai Admin' (Login as Admin) and 'Masuk sebagai Koordinator' (Login as Coordinator). A note at the bottom states: 'Admin menuju dashboard utama Ma'had. Koordinator tetap masuk ke sistem pengelolaan ini.' (Admin goes to the main Ma'had dashboard. Coordinator remains in this management system). The right section, also titled 'Masuk ke layanan Ma'had', prompts the user to 'Pilih jenis akun untuk melanjutkan ke halaman yang sesuai.' (Choose account type to continue to the appropriate page). It features three buttons: 'Mahasiswa' (Student), 'Pengelola' (Manager), and 'Ustadz' (Teacher). Below these, there are two login options: 'Masuk sebagai Admin' (Login as Admin) and 'Masuk sebagai Koordinator' (Login as Coordinator). A note at the bottom states: 'Admin menuju dashboard utama Ma'had. Koordinator tetap masuk ke sistem pengelolaan ini.' (Admin goes to the main Ma'had dashboard. Coordinator remains in this management system). The interface includes fields for 'USERNAME / NIPU' and 'PASSWORD', and a 'Masuk Sistem' (Login System) button.

Gambar 4.3 Halaman Login Koordinator dan Ustadz

4.2.2 Implementasi Dashboard Koordinator

Dashboard koordinator berfungsi sebagai pusat pengelolaan operasional. Halaman ringkasan menampilkan agregasi data pada periode aktif, sedangkan menu lainnya menyediakan pengelolaan mahasantri, ustadz, periode, mata pelajaran, lokasi, kelas, peserta, presensi, dan nilai sesuai kewenangan koordinator. Statistik kelas mengikuti filter gelombang yang sedang aktif agar angka yang ditampilkan tidak bercampur dengan data di luar konteks yang sedang diperiksa. Tampilan ringkasan dashboard koordinator disajikan pada Gambar 4.4.



Gambar 4.4 Dashboard Ringkasan Koordinator

4.2.2.1 Sinkronisasi data mahasantri.

Koordinator dapat menjalankan sinkronisasi untuk menarik data booking, santri, dan placement dari Sistem Ma'had lama. Backend menormalisasi NIM, menggabungkan sumber data, memvalidasi struktur, lalu melakukan upsert ke tabel Mahasantri. Data yang sudah memiliki NIM yang sama diperbarui dan tidak dibuat sebagai baris duplikat. Hasil proses dicatat pada SyncLog dengan ringkasan jumlah data berhasil dan gagal serta status SUCCESS, PARTIAL, atau FAILED. Sinkronisasi batch tidak selalu bersifat all-or-nothing, bagian data yang valid dapat dipertahankan ketika sumber data yang digunakan gagal. Sistem juga tidak otomatis menghapus data lokal hanya karena data tersebut tidak muncul pada satu proses sinkronisasi. Tampilan data mahasantri dan ringkasan sinkronisasi disajikan pada Gambar 4.5.

#	MAHASANTRI	PROGRAM STUDI	ANGKATAN	PROGRAM
01	A'ARYS TURISNA JS 250901827	Psikologi Penempatan	2025	Non-Regular
02	A ABDUL FATTAH ISRA 200301846	Ilmu Al-Qur'an dan Tafsir Laki-laki	2020	Non-Regular
03	A'ARIFA NACIA 200793001	Biologi Penempatan	2020	Regular
04	A'BDAN SYAKURA 230040006	Ilmu Ekonomi Laki-laki	2023	Non-Regular
05	A'TSYAH YULITA 250213095	Bimbingan Konseling Penempatan	2025	Regular

Gambar 4.5 Halaman Data Mahasantri dan Ringkasan Sinkronisasi

Implementasi inti sinkronisasi ditunjukkan melalui proses pengindeksan data berdasarkan NIM, penggabungan data booking dengan data santri dan placement, validasi setiap record, serta penyimpanan menggunakan operasi upsert. Potongan kode yang menunjukkan proses tersebut disajikan pada Gambar 4.6.

```
for (const booking of bookings) {
  const nim = booking.nim.trim();
  const placement = placementMap.get(nim);
  const santri = santriMap.get(nim);
  const alreadyExists = existingNims.has(nim);

  const validation = this.mergeAndValidate(
    booking,
    placement,
    santri,
    alreadyExists,
  );

  if (!validation.valid || !validation.entity) {
    errors.push({ nim, reasons: validation.error! });
    continue;
  }

  validRecords.push(validation.entity);

  if (placementAvailable && !placement && alreadyExists) {
    retainedMissingPlacement += 1;
  }
  if (santriAvailable && !santri && alreadyExists) {
    retainedMissingSantri += 1;
  }
  if (santri && !validation.hasSantriEnrichment && alreadyExists) {
    retainedInvalidSantri += 1;
  }
}

this.addRetentionWarnings(warnings, {
  retainedMissingPlacement,
  retainedMissingSantri,
  retainedInvalidSantri,
});

this.logger.log(
  'Merge complete: ${validRecords.length} valid, ${errors.length} failed',
);

if (validRecords.length > 0) {
  const chunks = this.chunkArray(
    validRecords,
    SyncMahasantriUseCase.CHUNK_SIZE,
  );
  this.logger.log(
    'Upserting ${validRecords.length} records in ${chunks.length} chunks...',
  );

  for (let index = 0; index < chunks.length; index += 1) {
    const chunkResult = await this.mahasantriRepo.upsertMany(
      chunks[index],
    );
    successCount += chunkResult;
    this.logger.log(
      'Chunk ${index + 1}/${chunks.length}: ${chunkResult} records upserted',
    );
  }
}
```

Gambar 4.6 Implementasi Sinkronisasi dan Upsert Data Mahasantri

Berdasarkan Gambar 4.6, NIM digunakan sebagai kunci untuk mempertemukan data dari beberapa sumber. Setiap record terlebih dahulu diproses melalui fungsi mergeAndValidate. Record yang tidak memenuhi ketentuan dicatat sebagai kesalahan dan tidak disimpan, sedangkan record yang valid dikumpulkan dan diproses dalam beberapa chunk menggunakan upsertMany. Pendekatan upsert memungkinkan data dengan NIM yang sama diperbarui tanpa menghasilkan duplikasi serta tidak memberikan operasi tulis kepada Sistem Ma'had lama.

4.2.2.2 Pengelolaan data operasional.

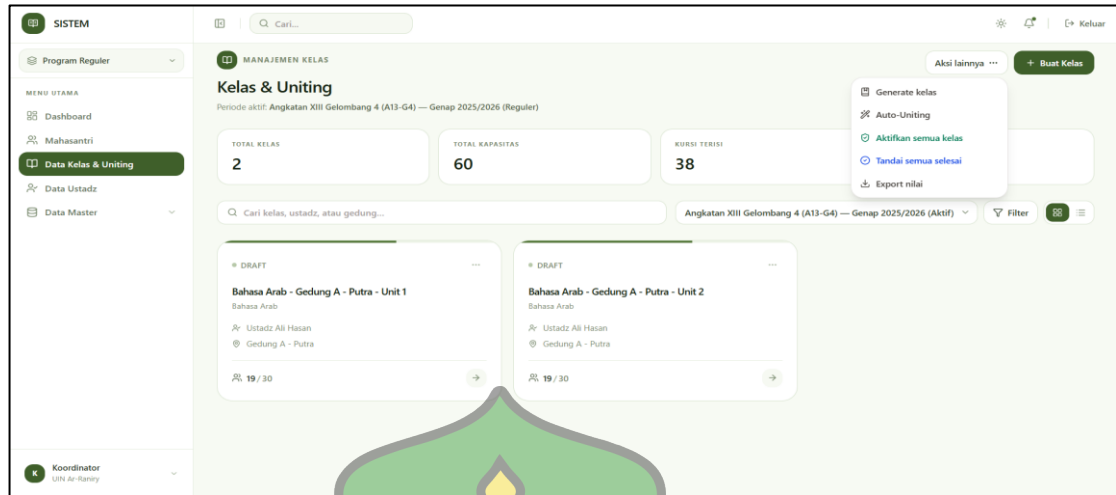
Koordinator dapat mengelola akun ustadz, periode akademik, mata pelajaran, aturan placement, lokasi, dan unit kelas. Satu periode aktif diperbolehkan untuk setiap jenis program, sehingga periode reguler dan non-reguler dapat berjalan bersamaan. Kelas memiliki lifecycle DRAFT, AKTIF, dan SELESAI. Aktivasi kelas memeriksa periode, jadwal, ustadz aktif, konflik jadwal, kapasitas lokasi, serta konfigurasi komponen nilai. Kelas yang selesai menjadi read-only bagi perubahan peserta serta input nilai dan presensi ustadz.

4.2.2.3 Pembentukan kelas dan uniting

Koordinator dapat membuat satu kelas, menghasilkan rekomendasi penambahan kelas, menempatkan peserta secara manual, atau menjalankan auto-uniting. Kandidat auto-uniting harus aktif, sudah check-in, sesuai gelombang dan program, sesuai gender lokasi, belum terdaftar pada konteks kelas yang dilarang, serta memenuhi rentang placement apabila mata pelajaran menggunakan placement. Algoritma memilih kelas yang memenuhi syarat dengan beban terproyeksi paling kecil dan menggunakan urutan stabil sebagai tie-breaker, sehingga distribusi bersifat deterministik dan lebih seimbang.

Penyimpanan peserta menggunakan transaksi dan penguncian data berurutan untuk menjaga kapasitas serta mencegah peserta ganda pada permintaan paralel. Apabila sebagian kandidat tidak memiliki kelas yang memenuhi syarat atau seluruh kelas yang sesuai telah penuh, sistem mempertahankan alokasi yang berhasil. Respons auto-uniting merangkum jumlah mahasantri yang berhasil dialokasikan, jumlah yang belum teralokasi, serta distribusi penambahan peserta pada setiap kelas. Implementasi saat ini belum memberikan alasan per peserta yang

tidak teralokasi; penelusuran kasus tersebut masih dilakukan dengan meninjau kriteria kandidat dan kapasitas kelas. Tampilan proses kelas dan auto-uniting disajikan pada Gambar 4.7.



Gambar 4.7 Halaman Kelas

Logika pemetaan otomatis tidak hanya membagi mahasantri secara berurutan, tetapi memeriksa kelayakan kandidat terhadap setiap kelas dan memilih kelas dengan beban terproyeksi paling kecil. Implementasi inti pemilihan kelas tersebut disajikan pada Gambar 4.8.

```
const sortedCandidates = [...candidates.values()].sort(compareCandidates);
for (const candidate of sortedCandidates) {
  let targetClass: RoundRobinClassInput | undefined;
  for (const kelas of subjectClasses) {
    const projected = projectedCounts.get(kelas.idKelas) ?? 0;
    if (
      projected >= kelas.kapasitas ||
      !eligibleNimsByClass.get(kelas.idKelas)?.has(candidate.nim)
    ) {
      continue;
    }

    if (!targetClass) {
      targetClass = kelas;
      continue;
    }

    const targetProjected = projectedCounts.get(targetClass.idKelas) ?? 0;
    if (
      projected < targetProjected ||
      (projected === targetProjected &&
        compareClasses(kelas, targetClass) < 0)
    ) {
      targetClass = kelas;
    }
  }

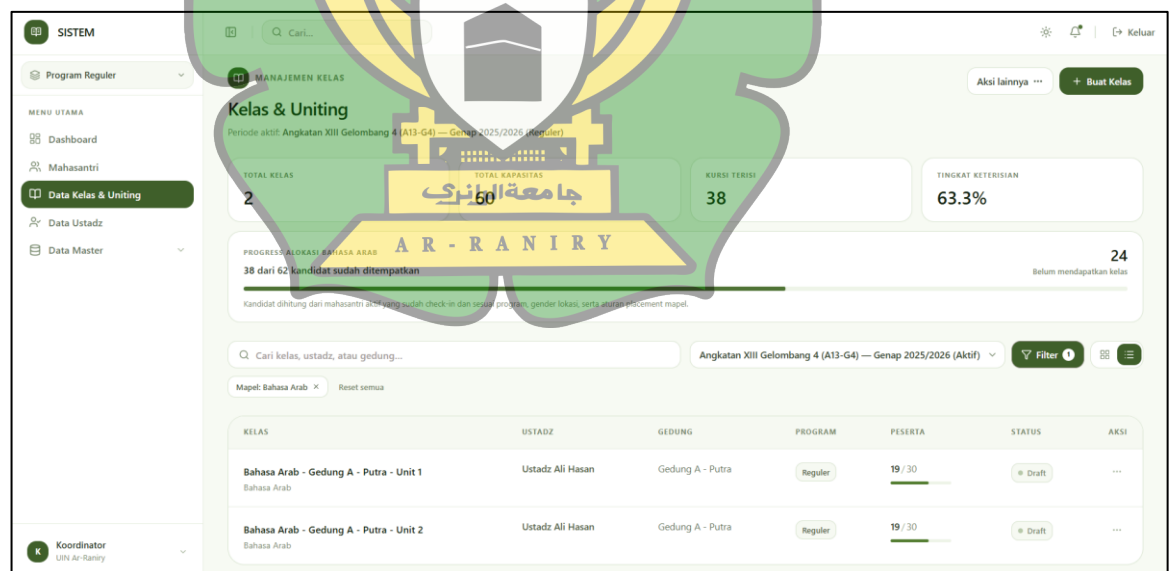
  if (!targetClass) continue;
  const classAssignments = assignments.get(targetClass.idKelas) ?? [];
  classAssignments.push(candidate.nim);
  assignments.set(targetClass.idKelas, classAssignments);
  projectedCounts.set(
    targetClass.idKelas,
    (projectedCounts.get(targetClass.idKelas) ?? 0) + 1,
  );
}
```

Gambar 4.8 Implementasi Algoritma Auto-Uniting

Berdasarkan Gambar 4.8, kandidat diproses dalam urutan yang stabil. Kelas yang telah mencapai kapasitas atau tidak memuat kandidat dalam daftar eligible dilewati. Dari kelas yang masih memenuhi syarat, sistem memilih kelas dengan jumlah peserta terproyeksi paling kecil. Apabila terdapat nilai yang sama, fungsi pembandingan kelas digunakan sebagai tie-breaker agar hasil distribusi tetap deterministik. Setelah kandidat ditetapkan, jumlah peserta terproyeksi diperbarui sebelum kandidat berikutnya diproses. Rencana distribusi selanjutnya disimpan melalui operasi atomik pada use case uniting untuk menjaga kapasitas dan mencegah peserta ganda.

4.2.2.4 Rekap distribusi.

Koordinator dapat meninjau total kelas, kapasitas, kursi terisi, tingkat okupansi, serta progres kandidat yang sudah dan belum dialokasikan. Rekap dapat dipersempit berdasarkan periode, program, lokasi, mata pelajaran, dan status kelas. Pemisahan kandidat per mata pelajaran mencegah jumlah peserta pada satu kelas ditafsirkan sebagai penyelesaian seluruh kebutuhan program. Tampilan rekap distribusi disajikan pada Gambar 4.9.

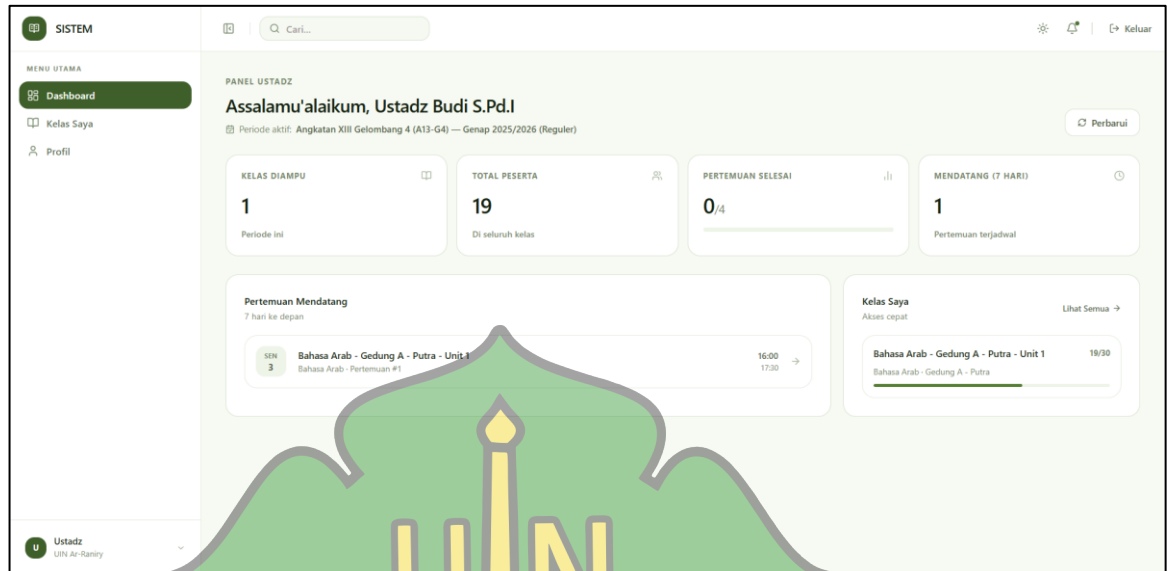


Gambar 4.9 Rekap Distribusi dan Statistik Kelas

4.2.3 Implementasi Dashboard Ustadz

Dashboard ustadz menampilkan ringkasan dan daftar kelas yang benar-benar ditugaskan kepada akun yang sedang masuk. Pembatasan ini tidak hanya diterapkan pada tampilan, tetapi juga diperiksa kembali oleh backend. Apabila

ustadz mencoba membuka alamat kelas milik ustadz lain secara langsung, permintaan ditolak dan data kelas tidak diberikan. Akun ustadz yang tidak aktif juga tidak dapat ditugaskan ke kelas baru. Tampilan dashboard dan daftar kelas ustadz disajikan pada Gambar 4.10.



Gambar 4.10 Dashboard dan Daftar Kelas Ustadz

4.2.3.1 Pengelolaan presensi

Ustadz memilih kelas dan pertemuan, kemudian menetapkan status HADIR, IZIN, SAKIT, atau ALPA untuk setiap peserta. Satu peserta hanya dapat memiliki satu record presensi pada satu pertemuan. Penyimpanan dilakukan melalui transaksi dan memperbarui informasi pertemuan yang telah terlaksana. Indikator progres menggunakan jumlah record pertemuan aktual sehingga jumlah sesi yang ditampilkan sesuai dengan data pertemuan yang benar-benar tersedia. Tampilan formulir presensi disajikan pada Gambar 4.11.

Bahasa Arab - Gedung A - Putra - Unit 1
 Pertemuan #1 - Senin, 3 Agustus 2026 - 16:00 - 17:30

Hadir: 13 Izin: 1 Sakit: 1 Alpha: 4 Total: 19

No	Nama	NIM	Hadir	Izin	Sakit	Alpha
1	ADLIANSYAH PUTRA	238385819	☒	☐	☐	☐
2	AHMAD ZUBAIDI S	258106174	☒	☐	☐	☐
3	ANDIKA SYAHPUTRA	258212888	☒	☐	☐	☐
4	AZRIANSYAH	238785221	☐	☐	☐	☒
5	FAHMI CUT RAJA	208211856	☒	☐	☐	☐
6	IMAM SHUFI	258083149	☐	☐	☒	☐
7	JUANDA AFIT MAULANA	208483868	☒	☐	☐	☐
8	M JOHAN ALAMSYAH	218182157	☐	☐	☐	☐

Gambar 4.11 Formulir Presensi Mahasantri per Pertemuan

4.2.3.2 Pengelolaan nilai

Sistem mendukung mode penilaian PERSENTASE dan SATUAN. Pada mode persentase, komponen aktif memiliki bobot total 100 dan sistem menghitung nilai akhir. Pada mode satuan, setiap komponen mempunyai skor maksimum, skor disimpan sebagai nilai mentah, dan sistem tidak membentuk nilai akhir persentase. Ustadz hanya dapat mengisi nilai ketika kelas aktif dan sesuai dengan kelas yang diampunya, sedangkan kelas selesai ditampilkan dalam keadaan read-only. Tampilan formulir dan rekap nilai disajikan pada Gambar 4.12.

Bahasa Arab - Gedung A - Putra - Unit 1
 Pertemuan #1 - Senin, 3 Agustus 2026 - 16:00 - 17:30

#	NAMA	NIM	TUGAS 40%	UJIAN 60%	AKHIR
1	ADLIANSYAH PUTRA	238385819			
2	AHMAD ZUBAIDI S	258106174			
3	ANDIKA SYAHPUTRA	258212888			
4	AZRIANSYAH	238785221			
5	FAHMI CUT RAJA	208211856			
6	IMAM SHUFI	258083149			
7	JUANDA AFIT MAULANA	208483868			
8	M JOHAN ALAMSYAH	218182157			
9	M. NABIL ZIKRY	258182806			
10	MHD FIRMANSYAH GINTING	258781866			
11	MUHAMMAD ALTAF	258781869			

Gambar 4.12 Formulir Pengisian dan Rekap Nilai Mahasantri

Pada sisi backend, setiap nilai diperiksa berdasarkan komponen yang aktif dan batas skor yang berlaku. Untuk kelas dengan mode persentase, nilai akhir hanya dihitung apabila seluruh komponen telah terisi. Implementasi validasi dan perhitungan tersebut disajikan pada Gambar 4.13.

```
private validateComponentChanges({
  const changes = new Map<number, number | null>();
  for (const item of submitted) {
    const idKomponen = Number(item.idKomponen);
    if (!Number.isSafeInteger(idKomponen) || idKomponen <= 0) {
      throw new BadRequestException('ID komponen nilai tidak valid');
    }
    if (changes.has(idKomponen)) {
      throw new BadRequestException(
        'Komponen nilai duplikat pada peserta ${idPeserta}',
      );
    }

    const config = komponen.find(
      (komponenItem) => komponenItem.idKomponen === idKomponen,
    );
    if (!config) {
      throw new BadRequestException(
        'Komponen dengan ID ${item.idKomponen} tidak aktif pada kelas ini',
      );
    }

    if (item.skor !== null) {
      this.validateScore(
        item.skor,
        modePenilaian === 'SATUAN' ? config.skorMaks : 100,
      );
    }
    changes.set(idKomponen, item.skor);
  }
  return changes;
})

private async evaluationResult({
  idPeserta: number,
  modePenilaian: 'PERSENTASE' | 'SATUAN',
  komponen: Awaitable<
    ReturnTypes[EvaluationUseCase['getActiveKomponenOrDefault']]>,
  scores: Map<number, number>,
}): EvaluationResult {
  const resultKomponen = komponen.map((item) => ({
    idKomponen: item.idKomponen,
    namaKomponen: item.namaKomponen,
    bobot: item.bobot,
    skorMaks: item.skorMaks,
    skor: scores.get(item.idKomponen) ?? null,
  }));
  const isComplete = komponen.every((item) => scores.has(item.idKomponen));
  const nilaiAkhir =
    modePenilaian === 'PERSENTASE' && isComplete
      ? Math.round(
          komponen.reduce(
            (sum, item) =>
              sum + (scores.get(item.idKomponen)! * item.bobot) / 100,
            0,
          ) / 100
        )
      : null;
  return { idPeserta, nilaiAkhir, komponen: resultKomponen };
}
```

Gambar 4.13 Implementasi Validasi dan Perhitungan Nilai Akhir

Gambar 4.13 menunjukkan bahwa sistem menolak komponen tidak valid, komponen duplikat, dan skor yang melampaui batas konfigurasi. Pada mode PERSENTASE, nilai akhir dihitung dari jumlah skor berbobot dan hanya dibentuk apabila seluruh komponen telah terisi. Pada mode SATUAN, skor komponen disimpan sebagai nilai mentah tanpa membentuk nilai akhir persentase.

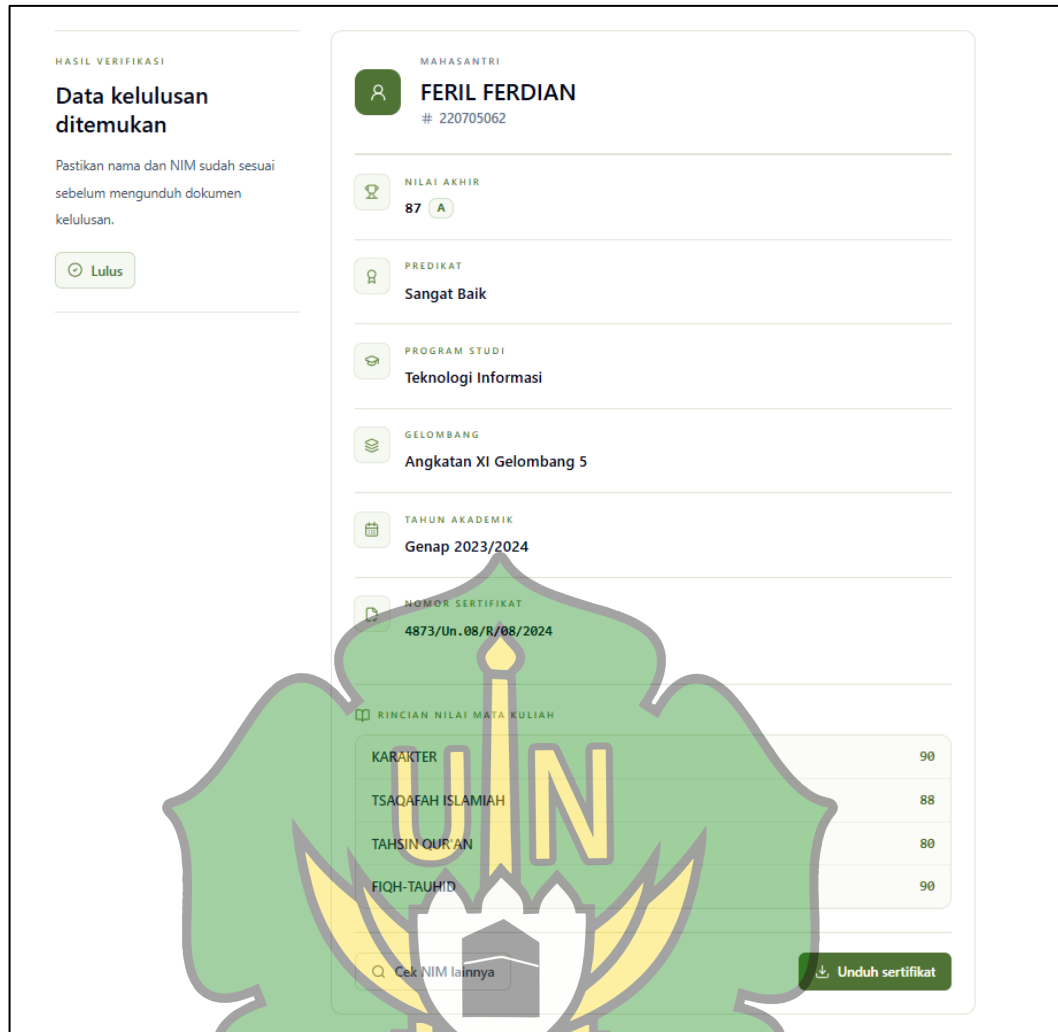
4.2.4 Implementasi Layanan Publik Kelulusan dan Sertifikat

Layanan publik tersedia pada halaman pengecekan kelulusan dan dapat digunakan tanpa login. Pengguna memasukkan NIM, kemudian frontend mengirim permintaan ke endpoint GET `/api/kelulusan/:nim` pada backend. Backend memangkas spasi pada awal dan akhir NIM lalu meneruskan permintaan ke

endpoint sertifikat pada Sistem Ma'had lama. Dengan pola tersebut, sistem baru tidak menentukan kelulusan dan tidak menghasilkan dokumen sertifikat sendiri; status serta tautan sertifikat tetap bersumber dari sistem lama. Halaman input NIM disajikan pada Gambar 4.14.

Apabila data kelulusan dan tautan sertifikat tersedia, sistem menampilkan informasi kelulusan serta tombol untuk membuka dokumen sertifikat. Apabila data kelulusan ditemukan tetapi tautan sertifikat belum tersedia, informasi kelulusan tetap ditampilkan dan antarmuka memberikan pemberitahuan 'Sertifikat belum tersedia'. Apabila NIM tidak ditemukan atau belum lulus, sistem menampilkan pesan generik bahwa data kelulusan tidak ditemukan atau belum tersedia. Pada implementasi saat ini, kegagalan komunikasi atau timeout dari layanan lama juga dipetakan ke respons generik yang sama sehingga penyebabnya belum dapat dibedakan oleh pengguna. Endpoint publik dilindungi rate limit untuk mengurangi penyalahgunaan permintaan. Contoh keluaran layanan dan tombol unduh disajikan pada Gambar 4.15.

Gambar 4.14 Halaman Pengecekan Kelulusan Berdasarkan NIM



Gambar 4.15 Hasil Kelulusan dan Tombol Unduh Sertifikat

Implementasi layanan publik menggunakan backend sebagai perantara untuk memanggil endpoint sertifikat pada Sistem Ma'had lama. Backend memvalidasi struktur respons sebelum mengembalikan data kelulusan dan tautan sertifikat kepada frontend. Potongan kode implementasinya disajikan pada Gambar 4.16.



```

async cekKelulusan(nim: string): Promise<KelulusanResponse> {
  if (!this.baseUrl) {
    this.logger.warn('LEGACY_API_BASE_URL tidak dikonfigurasi.');
```

```

    return {
      success: false,
      message: 'LEGACY_API_BASE_URL tidak dikonfigurasi',
    };
  }

  const url = `${this.baseUrl}/santri/${encodeURIComponent(nim)}/sertifikat`;

  try {
    const response = await firstValueFrom(
      this.httpService.get<unknown>(url, {
        headers: { Accept: 'application/json' },
        timeout: 15000,
      })
    );

    // Some endpoints return { data: {...} }, others return the object directly.
    const responseBody = response.data;
    const raw =
      isLegacyObject(responseBody) && isLegacyObject(responseBody.data)
        ? responseBody.data
        : responseBody;

    if (!isLegacyObject(raw) || !toNullableString(raw.nim)) {
      return {
        success: false,
        message: 'Data kelulusan tidak ditemukan atau belum tersedia',
      };
    }

    return { success: true, data: this.mapToKelulusanData(raw) };
  } catch (error) {
    this.logger.warn(
      {
        errorType:
          error instanceof Error ? error.constructor.name : 'UnknownError',
      },
      'Gagal mengambil data sertifikat dari legacy API',
    );
    return {
      success: false,
      message: 'Data kelulusan tidak ditemukan atau belum tersedia',
    };
  }
}

private mapToKelulusanData(raw: LegacyObject): KelulusanData {
  const idSertifikat = toNullableString(raw.idSertifikat);

  const nilaiMataKuliahList: SertifikatMataKuliah[] = Array.isArray(
    raw.nilaiMataKuliahList,
  )
    ? raw.nilaiMataKuliahList.filter(isLegacyObject).map((mataKuliah) => ({
        nilai: toLegacyString(mataKuliah.nilai),
        idMataKuliah: toLegacyString(mataKuliah.idMataKuliah),
        namaMataKuliah: toLegacyString(mataKuliah.namaMataKuliah).trim(),
      }))
    : [];

  return {
    nim: toLegacyString(raw.nim).trim(),
    nama: toLegacyString(raw.nama).trim(),
    jenisKelamin: toLegacyString(raw.jenisKelamin).trim(),
    rataRata: Number(raw.rataRata ?? 0),
    nilaiHuruf: toLegacyString(raw.nilaiHuruf).trim(),
    statusNilai: toLegacyString(raw.statusNilai).trim(),
    statusLulus: Boolean(raw.statusLulus),
    noSertifikat: toNullableString(raw.noSertifikat),
    idSertifikat,
    prodi: toLegacyString(raw.prodi).trim(),
    isReguler: Boolean(raw.isReguler),
    gelombang: toLegacyString(raw.gelombang).trim(),
    tahunAkademik: toLegacyString(raw.tahunAkademik).trim(),
    nilaiMataKuliahList,
    linkSertifikat: idSertifikat
      ? `${this.sertifikatBaseUrl}/${idSertifikat}`
      : null,
  };
}

```

Gambar 4.16 Implementasi Pengambilan Data Kelulusan dan Sertifikat

Berdasarkan Gambar 4.16, NIM terlebih dahulu dikodekan menggunakan `encodeURIComponent` sebelum ditempatkan pada alamat endpoint. Backend menerima respons JSON, memeriksa keberadaan objek dan NIM, kemudian memetakan atribut yang dibutuhkan ke bentuk data internal. Tautan sertifikat hanya dibentuk apabila `idSertifikat` tersedia. Dengan demikian, sistem baru tidak menetapkan status kelulusan dan tidak membuat dokumen sertifikat, tetapi menyajikan kembali data yang bersumber dari Sistem Ma'had lama melalui akses read-only.

4.2.5 Implementasi Basis Data dan Integrasi

Basis data operasional memisahkan identitas pengguna, data akademik, dan catatan sinkronisasi. Entitas utama meliputi Koordinator, Ustadz, Mahasantri, PeriodeAkademik, MataPelajaran, AturanPlacementMapel, LokasiGedung, KelasUnit, Pertemuan, PesertaKelas, Presensi, KomponenNilai, Nilai, NilaiKomponen, dan SyncLog. Relasi peserta kelas menjadi penghubung antara

Constraint basis data dan validasi aplikasi digunakan secara bersama. NIM



Pengujian fungsional dilaksanakan terhadap sembilan skenario t

dapat dijalankan, skenario diberi status Gagal. Bukti lengkap ditempatkan pada Lampiran 2 dengan menyamakan token, cookie, kata sandi, dan identitas sensitif.

4.3.1 Lingkungan dan Ketentuan Pengujian

Identitas versi sistem, perangkat, peramban, data uji, dan kondisi layanan eksternal dicatat agar pengujian dapat ditelusuri kembali. Rincian lingkungan pengujian disajikan pada Tabel 4.3.

Tabel 4.3 Identitas Pelaksanaan Black-Box Testing

Komponen	Keterangan
Alamat/lingkungan pengujian	Lokal, dengan frontend berada di url localhost:3000, dan backend berada di url localhost:3001
Tanggal pengujian	3 Agustus 2026
Penguji	Feril Ferdian sebagai peneliti dan pelaksana Black-Box Testing.
Perangkat klien	- Laptop MSI GF63 Thin 9SCSR - Prosesor Intel Core i5-9300H 2,40 GHz (4 core/8 logical processor) - RAM 16 GB - Windows 11 Home Single Language 64-bit.
Peramban	Brave Browser 151.1.93.129.
Data uji	Akun koordinator dan ustadz khusus pengujian, data master terkontrol, serta NIM uji yang mewakili kondisi pada Tabel 3.14. Data sensitif disamarkan.
Kondisi API Sistem Ma'had lama	Tersedia dan bisa digunakan

4.3.2 Hasil Pengujian Modul Koordinator

Hasil empat skenario fungsi vital pada modul koordinator dicatat pada Tabel 4.4. Status setiap skenario ditentukan setelah seluruh variasi di dalamnya diperiksa.

Tabel 4.4 Hasil Black-Box Testing Modul Koordinator

Kode	Tindakan/Data Uji	Hasil yang Diharapkan	Hasil Aktual	Status	Bukti
BB-K01	Koordinator masuk menggunakan	Sistem menerima autentikasi dan	Sistem berhasil memvalidasi akun SSO dan	Lulus	Lampiran 2.1

Kode	Tindakan/Data Uji	Hasil yang Diharapkan	Hasil Aktual	Status	Bukti
	akun SSO yang valid dan memiliki peran KOORDINATOR.	memberikan akses ke dashboard koordinator.	menampilkan respons success: true dengan pesan “Login SSO berhasil”. Data pengguna menunjukkan peran KOORDINATOR. Permintaan ke dashboard koordinator juga menghasilkan status 200 OK, sehingga dashboard dapat diakses.		
BB-K02	Koordinator menjalankan fungsi sinkronisasi data dari Sistem Ma’had lama.	Sistem menarik data dari sumber eksternal, memprosesnya, dan menampilkan ringkasan hasil sinkronisasi.	Permintaan POST /api/mahasantri/sync menghasilkan status 201 Created. Sistem menampilkan pesan “Sinkronisasi selesai” dan berhasil menarik 27.288 data. Sebanyak 24.716 data berhasil diproses, sedangkan 2.572 data tidak diproses karena tidak memenuhi	Lulus	Lampiran 2.2

Kode	Tindakan/Data Uji	Hasil yang Diharapkan	Hasil Aktual	Status	Bukti
			validasi data sumber.		
BB-K03	Koordinator mengisi formulir lokasi dengan data yang valid, kemudian menyimpannya.	Sistem menyimpan data lokasi dan menampilkan informasi bahwa data berhasil ditambahkan.	Sistem menampilkan respons success: true dengan pesan “Lokasi berhasil ditambahkan”. Data lokasi bernama “Lokasi Uji BB-K03” tersimpan dengan kapasitas total 20 dan kapasitas setiap unit 10.	Lulus	Lampiran 2.3
BB-K04	Koordinator memilih periode, jenis program, dan peruntukan gender, kemudian menjalankan proses uniting.	Sistem mendistribusikan mahasiswa ke dalam kelas berdasarkan kriteria dan kapasitas kelas yang tersedia.	Sistem menampilkan respons success: true dengan pesan “Uniting otomatis berhasil”. Sebanyak 63 mahasiswa berhasil ditempatkan ke kelas. Sistem juga menampilkan hasil distribusi dan jumlah mahasiswa yang belum memperoleh kelas.	Lulus	Lampiran 2.4

4.3.3 Hasil Pengujian Modul Ustadz

Hasil tiga skenario fungsi vital pada modul ustadz dicatat pada Tabel 4.5.

Variasi pengujian mencakup hak akses serta validasi data presensi dan nilai.

Tabel 4.5 Hasil Black-Box Testing Modul Ustadz

Kode	Tindakan/Data Uji	Hasil yang Diharapkan	Hasil Aktual	Status	Bukti
BB-U01	Ustadz masuk menggunakan akun aktif dan kredensial yang benar.	Sistem menerima autentikasi dan memberikan akses ke dashboard ustadz.	Permintaan login menghasilkan status 200 OK. Sistem berhasil mengidentifikasi ustadz dan menampilkan data dashboard yang meliputi periode aktif, jumlah kelas, jumlah peserta, jumlah pertemuan, dan daftar kelas yang diampu.	Lulus	Lampiran 2.5
BB-U02	Ustadz memilih kelas dan pertemuan, mengisi status kehadiran peserta, kemudian menyimpan presensi.	Sistem menyimpan data presensi seluruh peserta pada pertemuan yang dipilih.	Permintaan POST /api/presensi menghasilkan status 201 Created. Sistem menampilkan respons success: true dengan pesan "Presensi berhasil dicatat". Sebanyak 30 data presensi peserta berhasil direkam.	Lulus	Lampiran 2.6
BB-U03	Ustadz mengisi nilai peserta pada kelas yang diampu, kemudian menyimpan nilai tersebut.	Sistem memvalidasi dan menyimpan nilai seluruh peserta.	Sistem menampilkan respons success: true dengan pesan "Nilai batch berhasil diinput". Sebanyak 30 data nilai peserta berhasil diproses dan disimpan.	Lulus	Lampiran 2.7

4.3.4 Hasil Pengujian Layanan Publik Mahasantri

Hasil dua skenario layanan publik dicatat pada Tabel 4.6, yaitu pengecekan status kelulusan dan pengunduhan sertifikat berdasarkan NIM.

Tabel 4.6 Hasil Black-Box Testing Layanan Publik Mahasantri

Kode	Tindakan/Data Uji	Hasil yang Diharapkan	Hasil Aktual	Status	Bukti
BB-M01	Mahasantri memasukkan NIM yang memiliki data kelulusan, kemudian menekan tombol pengecekan.	Sistem menemukan data dan menampilkan informasi kelulusan mahasantri beserta link sertifikat.	Sistem menampilkan respons success: true dengan pesan "Data kelulusan ditemukan". Informasi kelulusan berhasil ditampilkan, meliputi status lulus, nilai rata-rata, nilai huruf, status nilai, beserta link sertifikat yang bisa diakses.	Lulus	Lampiran 2.8
BB-M02	Pengguna memasukkan NIM yang tidak terdaftar atau belum memiliki data kelulusan.	Sistem menolak pencarian dan menampilkan pemberitahuan, bahwa data belum tersedia tanpa menampilkan data mahasiswa lain.	Sistem menghasilkan status 404 <i>Not Found</i> dan menampilkan pesan "Data kelulusan tidak ditemukan atau belum tersedia". Tidak terdapat data mahasiswa lain yang ditampilkan.	Lulus	Lampiran 2.9

4.3.5 Rekapitulasi Hasil Pengujian Fungsional

Setelah seluruh variasi diperiksa, status sembilan skenario dirangkum berdasarkan kelompok modul pada Tabel 4.7.

Tabel 4.7 Rekapitulasi Black-Box Testing

Kelompok Modul	Jumlah Skenario	Lulus	Gagal	Keterangan
Koordinator	4	4	0	BB-K01 sampai BB-K04

Kelompok Modul	Jumlah Skenario	Lulus	Gagal	Keterangan
Ustadz	3	3	0	BB-U01 sampai BB-U03
Layanan publik	2	2	0	BB-M01 sampai BB-M02
Total	9	9	0	9 Lulus, 0 Gagal

Berdasarkan rekapitulasi tersebut, jumlah skenario yang memperoleh status Lulus adalah 9 dari sembilan skenario, sedangkan 0 skenario memperoleh status Gagal. Tingkat kesesuaian fungsi utama dihitung menggunakan rumus pada Subbab 3.4.1, yaitu

$$\text{Tingkat Kesesuaian Fungsi Utama} = \frac{\text{Jumlah Skenario Lulus}}{\text{Jumlah Seluruh Skenario}} \times 100\%$$

Sehingga:

$$\text{Tingkat Kesesuaian Fungsi Utama} = \frac{9}{9} \times 100\%$$

$$\text{Tingkat Kesesuaian Fungsi Utama} = 1 \times 100\%$$

$$\text{Tingkat Kesesuaian Fungsi Utama} = 100\%$$

Persentase tersebut hanya merupakan ringkasan pelaksanaan pengujian dan bukan ukuran usability atau kualitas perangkat lunak secara keseluruhan. Skenario yang gagal harus diperbaiki dan diuji ulang sebelum kesimpulan kesesuaian fungsional ditetapkan.

4.4 Hasil Evaluasi Kebergunaan Menggunakan System Usability Scale

Evaluasi kebergunaan dilakukan setelah responden menjalankan skenario penggunaan sesuai dengan perannya. Seluruh 2 koordinator menilai dashboard koordinator, seluruh 9 ustadz/ustadzah menilai dashboard ustadz, dan 20 mahasiswa yang dipilih secara purposif menilai layanan publik kelulusan. Hasil dianalisis secara terpisah karena setiap kelompok menggunakan fungsi dan tingkat kompleksitas antarmuka yang berbeda.

4.4.1 Karakteristik dan Realisasi Responden

Rencana responden terdiri atas 11 pengguna internal dan 20 mahasantri, sehingga jumlah keseluruhan responden SUS adalah 31 orang. Rencana dan realisasinya dicatat pada Tabel 4.8.

Tabel 4.8 Rencana dan Realisasi Responden SUS

Kelompok Pengguna	Rencana	Realisasi	Keterangan
Koordinator	2	2	Seluruh 2 koordinator dilibatkan melalui total <i>sampling</i> .
Ustadz/Ustadzah	9	9	Seluruh 9 ustadz/ustadzah dilibatkan melalui total <i>sampling</i> .
Mahasantri	20	20	Dipilih secara purposif dan menjalankan skenario SM-01 sampai SM-04.
Total	31	31	Realisasi akhir harus sama dengan jumlah kuesioner lengkap yang dianalisis.

Seluruh pengguna internal Ma'had dilibatkan melalui total sampling, yaitu 2 koordinator dan 9 ustadz/ustadzah. Sebanyak 20 mahasantri dipilih secara purposif berdasarkan kesesuaiannya dengan konteks layanan publik. Seluruh 31 responden telah menjalankan skenario penggunaan dan mengisi sepuluh pernyataan SUS secara lengkap, sehingga seluruh kuesioner dapat dianalisis.

4.4.2 Hasil Perhitungan Skor *System Usability Scale* (SUS)

Perhitungan SUS dilakukan untuk setiap responden dan disajikan secara terpisah berdasarkan kelompok pengguna. Kontribusi butir ganjil diperoleh dari nilai jawaban dikurangi 1, sedangkan kontribusi butir genap diperoleh dari 5 dikurangi nilai jawaban. Jumlah kontribusi ganjil dan genap menghasilkan total kontribusi setiap responden, kemudian dikalikan 2,5 untuk memperoleh skor SUS. Rincian perhitungan Koordinator, Ustadz/Ustadzah, dan Mahasantri disajikan pada sub bab dibawah.

4.4.2.1 Hasil Perhitungan Skor SUS Koordinator

Pengujian usability pada kelompok koordinator melibatkan seluruh pengguna yang menjalankan fungsi koordinator, yaitu sebanyak 2 responden. Hasil konversi jawaban menjadi skor kontribusi disajikan pada Tabel 4.9.

Tabel 4.9 Distribusi Skor Kontribusi Jawaban Responden Koordinator

Kode Responden	Q 1	Q 2	Q 3	Q 4	Q 5	Q 6	Q 7	Q 8	Q 9	Q10	Total Skor Kontribusi
K01	3	3	3	2	3	4	3	3	3	0	27
K02	3	4	3	2	4	4	2	4	4	1	31
Total											58

Berdasarkan Tabel 4.9, jumlah skor kontribusi dari kedua responden koordinator adalah 58. Rata-rata skor SUS kelompok koordinator dihitung menggunakan rumus:

$$Skor\ SUS = \frac{Total\ Skor\ Mentah}{Jumlah\ Responden} \times 2,5$$

Sehingga diperoleh skor rata-rata SUS sebagai berikut:

$$Skor\ SUS\ Koordinator = \frac{58}{2} \times 2,5 = 72,50$$

Hasil tersebut menunjukkan bahwa kelompok koordinator memperoleh rata-rata skor SUS sebesar 72,50. Berdasarkan kategori penerimaan SUS yang digunakan dalam penelitian ini, skor tersebut berada pada kategori Acceptable.

Secara individual, responden K01 memperoleh skor SUS sebesar 67,50 dan berada pada kategori Marginal, sedangkan responden K02 memperoleh skor sebesar 77,50 dan berada pada kategori Acceptable. Dengan demikian, meskipun rata-rata kelompok telah berada dalam kategori Acceptable, masih terdapat perbedaan penilaian usability di antara kedua responden koordinator.

4.4.2.2 Hasil Perhitungan Skor SUS Ustadz

Pengujian usability pada kelompok ustadz/ustadzah melibatkan seluruh pengguna yang mengelola aktivitas akademik melalui sistem, yaitu sebanyak 9 responden. Hasil konversi jawaban responden menjadi skor kontribusi disajikan pada Tabel 4.10.

Tabel 4.10 Distribusi Skor Kontribusi Jawaban Responden Ustadz/Ustadzah

Kode Responden	Q 1	Q 2	Q 3	Q 4	Q 5	Q 6	Q 7	Q 8	Q 9	Q10	Total Skor Kontribusi
U01	1	2	2	2	2	2	3	2	3	2	21
U02	4	0	4	1	4	4	4	4	4	1	30
U03	3	1	3	1	3	1	3	1	3	1	20
U04	3	2	3	1	3	2	2	4	2	2	24
U05	3	3	3	2	4	4	3	4	4	3	33
U06	4	3	4	3	4	4	4	4	4	3	37
U07	4	4	4	4	4	3	4	4	3	4	38
U08	4	3	4	4	4	4	4	4	4	4	39
U09	2	3	3	2	3	3	3	3	3	1	26
Total											268

Berdasarkan Tabel 4.10, jumlah skor kontribusi dari 9 responden ustadz/ustadzah adalah 268. Rata-rata skor SUS kelompok ustadz/ustadzah dihitung sebagai berikut:

$$\text{Skor SUS Koordinator} = \frac{268}{9} \times 2,5 = 74,44$$

Hasil tersebut menunjukkan bahwa kelompok ustadz/ustadzah memperoleh rata-rata skor SUS sebesar 74,44. Berdasarkan kategori penerimaan SUS yang digunakan, skor tersebut berada dalam kategori Acceptable.

Skor individual responden pada kelompok ustadz/ustadzah berada dalam rentang 50,00 sampai 97,50. Sebanyak 5 responden berada dalam kategori Acceptable, sedangkan 4 responden berada dalam kategori Marginal. Hal ini menunjukkan bahwa sistem secara umum dapat diterima oleh kelompok ustadz/ustadzah, tetapi pengalaman penggunaan belum sepenuhnya seragam pada seluruh responden.

4.4.2.3 Hasil Perhitungan Skor SUS Mahasantri

Pengujian usability pada kelompok mahasantri melibatkan 20 responden yang telah menjalankan skenario penggunaan layanan publik, yaitu pengecekan status kelulusan dan pengunduhan sertifikat berdasarkan NIM. Hasil konversi jawaban menjadi skor kontribusi disajikan pada Tabel 4.11.

Tabel 4.11 Distribusi Skor Kontribusi Jawaban Responden Mahasantri

Kode Responden	Q 1	Q 2	Q 3	Q 4	Q 5	Q 6	Q 7	Q 8	Q 9	Q10	Total Skor Kontribusi
M01	4	3	3	3	4	3	4	4	4	1	33
M02	4	4	4	4	4	3	4	0	4	4	35
M03	4	4	4	4	4	4	4	4	4	4	40
M04	3	4	4	4	3	4	2	4	3	4	35
M05	4	3	4	3	4	4	4	4	4	4	38
M06	4	4	4	3	4	3	4	4	3	3	36
M07	4	2	4	4	4	3	4	4	4	2	35
M08	3	4	4	4	4	4	4	4	4	4	39
M09	3	3	3	2	2	2	3	3	2	3	26
M10	4	4	4	4	4	4	4	4	4	4	40
M11	4	4	4	4	2	3	4	4	4	4	37
M12	4	4	4	3	3	3	3	4	4	4	36
M13	3	3	4	3	3	3	3	3	4	3	32
M14	4	4	4	4	4	4	4	4	4	3	39
M15	4	4	4	3	4	3	3	3	4	4	36
M16	4	4	4	4	4	4	4	4	4	4	40

Kode Responden	Q 1	Q 2	Q 3	Q 4	Q 5	Q 6	Q 7	Q 8	Q 9	Q10	Total Skor Kontribusi
M17	4	3	0	3	1	4	1	4	1	4	25
M18	4	4	4	4	3	3	4	4	3	4	37
M19	4	4	4	3	4	3	4	4	4	3	37
M20	4	3	3	3	4	4	4	3	3	4	35
Total											711

Berdasarkan Tabel 4.11, jumlah skor kontribusi dari 20 responden mahasantri adalah 711. Rata-rata skor SUS kelompok mahasantri dihitung sebagai berikut:

$$\text{Skor SUS Koordinator} = \frac{711}{20} \times 2,5 = 88,88$$

Hasil tersebut menunjukkan bahwa kelompok mahasantri memperoleh rata-rata skor SUS sebesar 88,88. Berdasarkan kategori penerimaan SUS yang digunakan, skor tersebut berada dalam kategori Acceptable.

Skor individual responden mahasantri berada dalam rentang 62,50 sampai 100,00. Sebanyak 18 responden berada dalam kategori Acceptable dan 2 responden berada dalam kategori Marginal. Hasil tersebut menunjukkan bahwa layanan publik secara umum dapat diterima dan dinilai mudah digunakan oleh sebagian besar responden mahasantri.

4.4.2.4 Rekapitulasi Hasil SUS

Rekapitulasi hasil pengukuran SUS berdasarkan kelompok pengguna disajikan pada Tabel 4.12.

Tabel 4.12 Rekapitulasi Hasil SUS Berdasarkan Kelompok Pengguna

Kelompok Pengguna	Jumlah Responden	Total Skor Kontribusi	Skor SUS Minimum	Skor SUS Maksimum	Rata - Rata SUS	Interpretasi
Koordinator	2	58	67,50	77,50	72,50	Acceptable
Ustadz/Ustadzah	9	268	50,00	97,50	74,44	Acceptable
Mahasantri	20	711	62,50	100,00	88,88	Acceptable

Berdasarkan Tabel 4.12, ketiga kelompok pengguna memperoleh rata-rata skor SUS dalam kategori Acceptable. Kelompok mahasantri memperoleh rata-rata tertinggi, yaitu 88,88, diikuti oleh kelompok ustadz/ustadzah sebesar 74,44 dan kelompok koordinator sebesar 72,50.

Hasil tersebut menunjukkan bahwa sistem secara umum dapat diterima dari aspek usability oleh ketiga kelompok pengguna. Meskipun demikian, masih terdapat responden dengan skor individual dalam kategori Marginal, khususnya pada kelompok pengguna internal. Oleh karena itu, hasil rata-rata kelompok tidak dimaknai bahwa seluruh responden memberikan penilaian yang sama. Variasi tersebut menunjukkan bahwa masih terdapat ruang perbaikan, terutama untuk meningkatkan kemudahan penggunaan dan konsistensi pengalaman pengguna pada dashboard koordinator dan dashboard ustadz/ustadzah.

Secara keseluruhan, hasil pengujian SUS mendukung bahwa Sistem Informasi Operasional Ma'had yang dikembangkan telah memiliki tingkat penerimaan usability yang memadai untuk digunakan dalam mendukung pengelolaan akademik dan administrasi sesuai dengan ruang lingkup penelitian.

4.5 Pembahasan

Bagian pembahasan mengaitkan hasil implementasi dengan bukti pengujian fungsional dan evaluasi kebergunaan. Pembahasan difokuskan pada kesesuaian kebutuhan sistem, proses operasional, integrasi, keamanan, keterbatasan, serta jawaban terhadap rumusan masalah penelitian.

4.5.1 Kesesuaian Implementasi dengan Kebutuhan Sistem

Kesesuaian implementasi dinilai dengan membandingkan kebutuhan utama pada Bab III terhadap fungsi yang tersedia dan hasil pengujiannya. Status pada Tabel 4.13 didasarkan pada hasil Black-Box Testing serta evaluasi SUS yang telah

dilaksanakan, sehingga tidak hanya menunjukkan keberadaan halaman atau endpoint, tetapi juga bukti bahwa fungsi utama berjalan sesuai keluaran yang diharapkan.

Tabel 4.13 Keterkaitan Kebutuhan dan Hasil Implementasi

Kebutuhan Utama	Hasil Implementasi	Status Pelaporan
Mengurangi entri ulang data mahasantri	Sinkronisasi read-only menggabungkan data booking, santri, dan placement berdasarkan NIM serta mencatat SyncLog.	Terpenuhi. Didukung hasil BB-K02 yang berstatus Lulus.
Mendukung pembentukan dan distribusi kelas	Koordinator dapat membuat kelas, menempatkan peserta secara manual atau otomatis, serta melihat jumlah dan distribusi hasil auto-uniting.	Terpenuhi. Didukung hasil BB-K03 dan BB-K04 yang berstatus Lulus.
Membatasi akses akademik ustadz	Kelas, roster, presensi, dan nilai dibatasi berdasarkan penugasan serta status akun.	Terpenuhi. Didukung hasil BB-U01 sampai BB-U03 yang berstatus Lulus.
Menyediakan layanan kelulusan mandiri	Pengguna mencari berdasarkan NIM; backend membaca status dan tautan sertifikat dari layanan lama.	Terpenuhi. Didukung hasil BB-M01 dan BB-M02 yang berstatus Lulus.
Menilai kebergunaan sistem	Skenario penggunaan dan instrumen SUS telah disiapkan untuk tiga kelompok pengguna.	Terpenuhi. Rata-rata SUS koordinator 72,50, ustadz/ustadzah 74,44, dan mahasantri 88,88, seluruhnya berkategori Acceptable.

4.5.2 Pembahasan terhadap Proses Operasional

Sinkronisasi berbasis NIM dirancang untuk mengurangi kebutuhan memasukkan ulang identitas mahasantri. Keuntungan arsitekturalnya adalah sistem lama tetap menjadi sumber identitas awal, sedangkan sistem baru dapat mengelola state operasional tanpa memperoleh hak tulis ke basis data lama. Namun, manfaat pengurangan waktu kerja belum dapat dinyatakan secara kuantitatif karena penelitian ini tidak mengukur durasi proses sebelum dan sesudah penggunaan sistem.

Pada proses uniting, distribusi terimbang menjawab kebutuhan agar peserta tidak hanya memenuhi kapasitas, tetapi juga tersebar lebih proporsional pada kelas yang memenuhi kriteria. Pemeriksaan program, gelombang, gender, placement, kapasitas, dan duplikasi dilakukan sebelum penyimpanan. Sistem merangkum jumlah mahasantri yang berhasil dan belum berhasil dialokasikan beserta distribusinya per kelas.

Dashboard ustadz memisahkan tanggung jawab pengajaran dari pengelolaan data master. Pembatasan kelas berdasarkan penugasan mengurangi kemungkinan ustadz mengubah data kelas lain. Presensi berbasis pertemuan dan komponen nilai yang fleksibel memberi ruang bagi variasi pelaksanaan akademik.

Layanan kelulusan berbasis NIM menyediakan jalur mandiri tanpa mewajibkan mahasantri membuat akun baru. Arsitektur penerusan data menjaga status kelulusan dan sertifikat pada sumber yang sama dengan layanan lama, sehingga sistem baru tidak membuat salinan sertifikat yang berpotensi tidak sinkron. Meskipun demikian, peningkatan efisiensi layanan baru dapat dinyatakan sebagai dampak yang terbukti setelah didukung data penggunaan, catatan proses, atau evaluasi pengguna.

4.5.3 Pembahasan Integrasi dan Keamanan

Pemisahan frontend, backend, basis data operasional, penyedia identitas, dan Sistem Ma'had lama memberikan batas tanggung jawab yang jelas. Backend menjadi pusat validasi dan otorisasi, sehingga perubahan alamat halaman atau manipulasi antarmuka tidak cukup untuk melewati kebijakan akses. Pemeriksaan akun pada setiap request memungkinkan penonaktifan ustadz berlaku terhadap token lama, sedangkan kebijakan kepemilikan kelas melindungi data akademik dari akses lintas pengajar.

Autentikasi koordinator menggunakan Authorization Code dengan PKCE lebih sesuai untuk aplikasi browser dibandingkan pertukaran kredensial langsung. Walaupun demikian, keberhasilan integrasi pada Keycloak yang tersedia belum sama dengan serah terima integrasi ke penyedia identitas resmi UIN. Migrasi tersebut tetap memerlukan konfigurasi resmi, kesesuaian issuer dan client, redirect URI, serta pemetaan role koordinator dari pihak pengelola identitas UIN.

4.5.4 Keterbatasan Hasil Pengembangan

Hasil pengembangan memiliki beberapa keterbatasan. Pertama, ketersediaan sinkronisasi dan layanan sertifikat bergantung pada kestabilan serta kontrak REST API Sistem Ma'had lama. Gangguan, perubahan struktur respons, atau timeout pada layanan tersebut dapat memengaruhi fungsi sistem baru. Pada layanan kelulusan, kegagalan komunikasi saat ini dipetakan ke pesan generik yang sama dengan data tidak ditemukan atau belum tersedia sehingga penyebab kegagalan belum dapat dibedakan oleh pengguna. Kedua, pengecekan kelulusan hanya memakai NIM dan rate limit. Untuk penggunaan publik dalam skala lebih luas, faktor verifikasi tambahan perlu dipertimbangkan agar pencarian massal atau enumerasi NIM lebih sulit dilakukan.

Ketiga, sesi frontend saat ini masih menggunakan JWT aplikasi yang dapat dibaca JavaScript dari cookie dan belum menerapkan rancangan sesi HttpOnly serta proteksi CSRF final. Implementasi PKCE telah memperkuat alur login koordinator, tetapi tidak menghilangkan kebutuhan penguatan manajemen sesi browser. Keempat, beberapa aturan operasional, seperti konsistensi lock pada seluruh jalur perubahan pertemuan dan definisi akademik untuk status pertemuan tertentu, masih memerlukan penyempurnaan serta pengujian lanjutan.

Kelima, penelitian ini tidak mencakup aplikasi native mobile, atau landing page. Evaluasi SUS pada mahasiswa melibatkan 20 responden yang dipilih secara purposif untuk menilai modul layanan publik. Oleh karena itu, hasilnya digunakan sebagai gambaran kebergunaan pada pengguna sasaran yang berpartisipasi dan bukan untuk menggeneralisasikan kondisi seluruh populasi mahasiswa UIN Ar-Raniry. Penelitian ini juga belum mengukur perubahan waktu kerja administratif sebelum dan sesudah penerapan sistem.

4.5.5 Pembahasan terhadap Rumusan Masalah

Rumusan masalah pertama membahas cara merancang dan mengembangkan sistem operasional yang mendukung uniting oleh koordinator, pengelolaan presensi dan nilai oleh ustadz, serta layanan kelulusan dan sertifikat berdasarkan NIM. Berdasarkan hasil implementasi pada Subbab 4.1 dan 4.2, ketiga kelompok fungsi tersebut telah diterjemahkan menjadi arsitektur web, endpoint, antarmuka, model data, dan aturan akses yang saling terhubung. Sistem

menggunakan integrasi read-only agar data awal dan sertifikat tetap bersumber dari Sistem Ma'had lama, sementara proses akademik operasional dikelola pada basis data baru.

Rumusan masalah kedua membahas hasil evaluasi berdasarkan kesesuaian fungsional dan kebergunaan. Pengujian Black-Box menunjukkan bahwa seluruh 9 skenario fungsi vital memperoleh status Lulus, sehingga tingkat kesesuaian fungsi utama mencapai 100%. Evaluasi SUS menghasilkan rata-rata 72,50 pada koordinator, 74,44 pada ustadz/ustadzah, dan 88,88 pada mahasantri, ketiganya berada pada kategori Acceptable. Hasil tersebut menunjukkan bahwa sistem telah memenuhi skenario fungsi utama dan dapat diterima dari sisi kebergunaan oleh responden penelitian. Namun, persentase Black-Box dan skor SUS tidak digunakan sebagai bukti kuantitatif peningkatan efisiensi administratif karena penelitian tidak membandingkan waktu atau beban kerja sebelum dan sesudah penggunaan sistem.



BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

1. Sistem Informasi Operasional Ma'had berbasis web telah berhasil dirancang dan dikembangkan sesuai ruang lingkup penelitian. Sistem mengintegrasikan frontend Next.js, backend NestJS, basis data PostgreSQL, autentikasi berbasis peran, serta REST API Sistem Ma'had lama secara read-only. Fungsionalitas yang dihasilkan mencakup sinkronisasi data dan uniting kelas pada dashboard koordinator, pengelolaan presensi dan nilai pada dashboard ustadz, serta pengecekan status kelulusan dan pengunduhan sertifikat berdasarkan NIM melalui layanan publik.
2. Pengujian fungsional terhadap sembilan skenario fungsi vital menghasilkan 9 skenario Lulus dan 0 skenario Gagal, sehingga tingkat kesesuaian fungsi utama mencapai 100%.
3. Evaluasi SUS memperoleh rata-rata 72,50 untuk koordinator, 74,44 untuk ustadz/ustadzah, dan 88,88 untuk mahasiswa, seluruhnya termasuk kategori Acceptable. Dengan demikian, sistem memenuhi fungsi utama yang diuji dan dapat diterima dari sisi kebergunaan oleh responden penelitian. Hasil ini bersifat deskriptif untuk konteks pengujian dan tidak digunakan untuk menyatakan peningkatan efisiensi administratif secara kuantitatif.

5.2 Saran

1. Sistem perlu dipersiapkan untuk penerapan pada lingkungan produksi melalui koordinasi dengan pengelola ICT UIN Ar-Raniry, terutama untuk konfigurasi penyedia identitas resmi, pemetaan peran koordinator, alamat layanan, dan pengelolaan variabel lingkungan.
2. Integrasi dengan REST API Sistem Ma'had lama perlu diperkuat melalui pemantauan ketersediaan layanan, mekanisme retry yang terkendali, pemeriksaan perubahan struktur respons, serta pesan kesalahan yang dapat membedakan data tidak ditemukan, sertifikat belum tersedia, dan gangguan komunikasi.

3. Keamanan aplikasi perlu ditingkatkan dengan manajemen sesi berbasis cookie HttpOnly, proteksi CSRF, pembatasan akses yang konsisten pada seluruh endpoint, serta mekanisme pencegahan pencarian NIM secara massal pada layanan publik.
4. Antarmuka untuk pengguna internal perlu dilengkapi panduan singkat dan pembiasaan penggunaan. Proses unting juga dapat dikembangkan agar menampilkan alasan setiap peserta yang belum memperoleh kelas, sedangkan layanan publik perlu menyempurnakan alur dan umpan balik ketika sertifikat tidak dapat diunduh.
5. Penelitian selanjutnya disarankan melakukan evaluasi pada lingkungan produksi dalam periode penggunaan yang lebih panjang, mengukur waktu penyelesaian tugas dan kesalahan kerja sebelum dan sesudah penerapan, serta menambahkan pengujian kinerja dan keamanan. Sampel yang lebih besar atau probability sampling dapat digunakan apabila penelitian berikutnya bertujuan melakukan generalisasi terhadap populasi yang lebih luas.



DAFTAR PUSTAKA

- Amrulloh, A., & Sela, E. I. (2021). Course scheduling optimization using genetic algorithm and tabu search. *Jurnal Teknologi Dan Sistem Komputer*, 9(3), 157–166. <https://doi.org/10.14710/jtsiskom.2021.14137>
- Anggraeni, E. Y. I. R. (2017). *Pengantar Sistem Informasi* (E. Risanto, Ed.; Edisi Pertama). CV. ANDI OFFSET.
- Aprisilia Anastasia, L., Pinontoan, B., & Alexander Latumakulita, L. (2022). *Design Web-Based Information System of Tri Dharma Higher Education for Lecturer*. 3(1), 23–31. <http://iorajournal.org/indx.php/orics/index>
- Brooke, J. (1995). SUS: A quick and dirty usability scale. *Usability Eval. Ind.*, 189.
- Cherny, Boris. (2019). *Programming TypeScript : making your JavaScript applications scale*. O'Reilly Media, Inc.
- Eka, Indah. W., Fatimah, M., Bintang, P., & Tulas, N. (2025). Pengembangan Sistem Informasi Manajemen Sekolah Berbasis Web untuk Meningkatkan Efisiensi Administrasi Pendidikan. *Uranus : Jurnal Ilmiah Teknik Elektro, Sains Dan Informatika*, 3(2), 65–76. <https://doi.org/10.61132/uranus.v3i2.802>
- Fadhli, M. (2025). *Implementasi Metode Halaqah Dalam Pembelajaran Fiqih Santri Di Ma'had Al-Jami'ah Uin Raden Intan Lampung*. <https://repository.radenintan.ac.id/37964/1/MUHAMMAD%20FADHLI%20NP M.%202111010307.pdf>
- Fadilah, A. G., Santi Wahyuni, F., & Achmadi, S. (2022). Implementasi Metode K-Means Clustering Untuk Pengelompokan Kelas Berdasarkan Pemahaman Siswa Pada Bimbingan Belajar Matematika Saschio Banyuwangi. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 6, Number 2).
- Faulkner, L. (2003). Beyond the five-user assumption: Benefits of increased sample sizes in usability testing. *Behavior Research Methods, Instruments, & Computers*, 35(3), 379–383. <https://doi.org/10.3758/BF03195514>
- Gupta, L. (2025, April 1). *What is REST?: REST API Tutorial*. <https://restfulapi.net/>
- IBM. (2025). *Apa itu Kontrol Akses Berbasis Peran (RBAC)? | IBM*. IBM. <https://www.ibm.com/id-id/think/topics/rbac>
- Ikhwan, I. (2026). Pengembangan RESTFul Web Service Untuk Integrasi Data Pada Sistem Informasi Akademik. *RIGGS: Journal of Artificial Intelligence and Digital Business*, 5(1), 3121–3128. <https://doi.org/10.31004/riggs.v5i1.6547>
- Inayah, S., Firdaus, M., Dianawati, E., & Tooy, S. (2025). *Sistem Informasi Manajemen*.

- International Software Testing Qualifications Board. (2024). *Certified Tester Foundation Level Syllabus (Version 4.0.1)*. https://istqb.org/wp-content/uploads/2024/11/ISTQB_CTFL_Syllabus_v4.0.1.pdf
- Ismail, A. (2022). Peran Ma'had Al-Jami'ah Walisongo dalam Mewujudkan Pendidikan Berbasis Islam Rahmatan Lil 'Alamin. *Dimas: Jurnal Pemikiran Agama Untuk Pemberdayaan*, 21(2), 323–339. <https://doi.org/10.21580/dms.2021.212.9363>
- Koripalli, M. (2025). Cross-Platform Data Management: Patterns and Best Practices. *European Journal of Computer Science and Information Technology*, 13(10), 57–72. <https://doi.org/10.37745/ejcsit.2013/vol13n105772>
- Kristanti, T., & Putra, H. R. (2025). Penerapan Sistem Informasi Manajemen di Sekolah untuk Meningkatkan Efisiensi Administrasi dan Pembelajaran. *DIRASAH*, 8(1). <https://ejournal.iaifa.ac.id/index.php/dirasah>
- Kustitskaya, T. A., Esin, R. V., Kytmanov, A. A., & Zykova, T. V. (2023). Designing an Education Database in a Higher Education Institution for the Data-Driven Management of the Educational Process. *Education Sciences*, 13(9). <https://doi.org/10.3390/educsci13090947>
- Laeliyah, F. (2025). Peran Sistem Informasi Manajemen Pendidikan (Simdik) Dalam Meningkatkan Efisiensi Administrasi Sekolah. *Jurnal Ilmiah Pendidikan Dasar*, 10.
- Makmur, F. (2024). *Implementasi Sistem Informasi Manajemen Pendidikan Di Ma'had Syaikh Muhammad Bin Sholeh* [Thesis, Universitas Islam Negeri Sultan Syarif Kasim Riau]. <https://repository.uin-suska.ac.id/id/eprint/76537>
- Mardhatillah Ahmad, & Musleh Wahid. (2024). Pemanfaatan Aplikasi Sistem Informasi Santri di Ma'had Tahfidh Al-Qur'an Putri Al-Amien. *Jurnal Manajemen Dan Pendidikan Agama Islam*, 2(3), 175–179. <https://doi.org/10.61132/jmpai.v2i3.278>
- Nestjs Documentation. (2025). *Documentation | NestJS - A progressive Node.js framework*. Nestjs. <https://docs.nestjs.com/>
- Newman, Sam. (2021). *Building microservices: designing fine-grained systems*. O'Reilly Media.
- Ngcwabe, Y., & Chiliya, N. (2014). An assessment of the effectiveness of self-service technology in a University enviroment: The case of University of the Witwatersrand. *Mediterranean Journal of Social Sciences*, 5(14), 413–427. <https://doi.org/10.5901/mjss.2014.v5n14p413>
- Nurdin, A., Saptadi, N., Kristiawan, H., Nurhaeni, T., Yunianto, I., Arisandi, D., Putra, P., Hanantyo, B., Mahmudi, A., Sodik, F., Syakur, Purwati, N., Afrianto, I., Mutmainah, S., Soleh, O., & Pustaka, S. (2024). *Pengantar Teknologi Informasi*.

- Nurfajrina, A. (2023, November 23). *Ma'had Adalah? Ini Pengertian dan Bedanya dengan Pesantren*. <https://www.detik.com/hikmah/khazanah/d-7051206/mahad-adalah-ini-pengertian-dan-bedanya-dengan-pesantren>
- Postgresql Team. (2023). *PostgreSQL: About*. Postgresql.Com. <https://www.postgresql.org/about/>
- Pratisto, E., Athifah, D., & Purnomo, F. (2024). Pengembangan dan Uji Usability Sistem Informasi Presensi Berbasis Web di Institusi Pendidikan. *Indonesian Journal of Applied Informatics*, 9, 168. <https://doi.org/10.20961/ijai.v9i1.94878>
- Pressman, R. S., & Maxim, B. R. (2019). *Software Engineering: A Practitioners Approach*. McGraw-Hill Education. <https://books.google.co.id/books?id=taIKxAEACAAJ>
- Qo'du, H., Hakim, I., Sayida Bilqis, S., Ramadhani, R., & Vindua, R. (2025). *Peran Sistem Informasi Dalam Meningkatkan Mutu Layanan Pendidikan* (Vol. 6, Number 1).
- Rafid Pratama, M., Umam, J., & Yakok, R. (2024). Usability Testing pada Aplikasi iJateng Menggunakan Metode System Usability Scale. *Manajemen Dan Teknologi Informasi*, 2(1).
- Rahman, F., Anam, M., & Sodik, A. (2023). *Pattern of PTKIN Ma'had Al-Jami'ah Education Management System in East Java and Its Role in Mainstreaming Religious Moderation in Indonesia* (pp. 135–154). https://doi.org/10.2991/978-2-38476-044-2_15
- Rahmi, E. R., Yumami, E., & Hidayasari, N. (2023). Analisis Metode Pengembangan Sistem Informasi Berbasis Website: Systematic Literature Review. *Remik*, 7(1), 821–834. <https://doi.org/10.33395/remik.v7i1.12177>
- Rismawati, R., Kartika, I., & Arifudin, O. (2024). Peran Sistem Informasi Dalam Meningkatkan Mutu Layanan Pendidikan. In *Jurnal Tahsinia* (Vol. 5, Number 7).
- Sahyudi, S., & Voutama, A. (2025). Pengujian Fungsional Black Box SISKAS UNSIKA dengan Equivalence Partitioning untuk Validasi Input dan Output Sistem. *Volume*, 8, 148–155. <https://ojs.trigunadharma.ac.id/index.php/jsk/index>
- Sharfina, Z., & Santoso, H. B. (2016). An Indonesian adaptation of the System Usability Scale (SUS). *2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 145–148. <https://doi.org/10.1109/ICACSIS.2016.7872776>
- Sutabri, T. (2012). *Analisis Sistem Informasi*. Penerbit Andi. <https://books.google.co.id/books?id=ro5eDwAAQBAJ>
- Tullis, T. S., & Stetson, J. N. (2004). *A Comparison of Questionnaires for Assessing Website Usability*. <https://api.semanticscholar.org/CorpusID:9670323>
- Vercel. (2025). *Next.js Docs | Next.js*. Vercel. <https://nextjs.org/docs>

- Wahab, K., Ullah, A., Wang, Y., & Majid, A. (2023). Development of a Web-Based Student Portal System for University Students. *2023 International Conference on Intelligent Education and Intelligent Research (IEIR)*, 1–8. <https://doi.org/10.1109/IEIR59294.2023.10391217>
- Wasilah, N., Putra, Y., Masyaroh, F., & Handayani, N. (2024). Implementasi Sistem Informasi Manajemen dalam Pengelolaan Data Peserta Didik di SDN Jambringin 2. *JIIIP - Jurnal Ilmiah Ilmu Pendidikan*, 7, 2842–2846. <https://doi.org/10.54371/jiip.v7i3.4187>



LAMPIRAN

Lampiran 1 Transkrip Wawancara

Narasumber : Staf Pengelola Ma'had Al-Jami'ah UIN Ar-Raniry

Waktu : 10 February 2026

Metode : Wawancara semi-terstruktur

Peneliti:

Bagaimana efektivitas penggunaan Sistem Informasi Ma'had yang saat ini sedang berjalan? Apakah sudah cukup membantu operasional staf sehari-hari?

Narasumber:

Secara umum memang kita sudah punya sistem informasi, tapi efektivitasnya di lapangan belum begitu optimal. Kendala utama yang paling terasa itu di awal periode saat proses pemetaan kelas atau *uniting* mahasiswa. Saat ini, pembagian kelasnya masih kami kerjakan secara manual menggunakan Microsoft Excel di luar sistem. Akibatnya, pendataan sering lambat, rentan ada data yang tidak sinkron, dan kami sebagai koordinator tidak punya *dashboard* khusus untuk memantau kapasitas distribusi kelas secara *real-time*.

Peneliti:

Lalu untuk proses akademik di dalam kelas sendiri bagaimana? Apakah tenaga pengajar (ustadz) sudah bisa melakukan rekapitulasi nilai dan absensi langsung ke dalam sistem?

Narasumber :

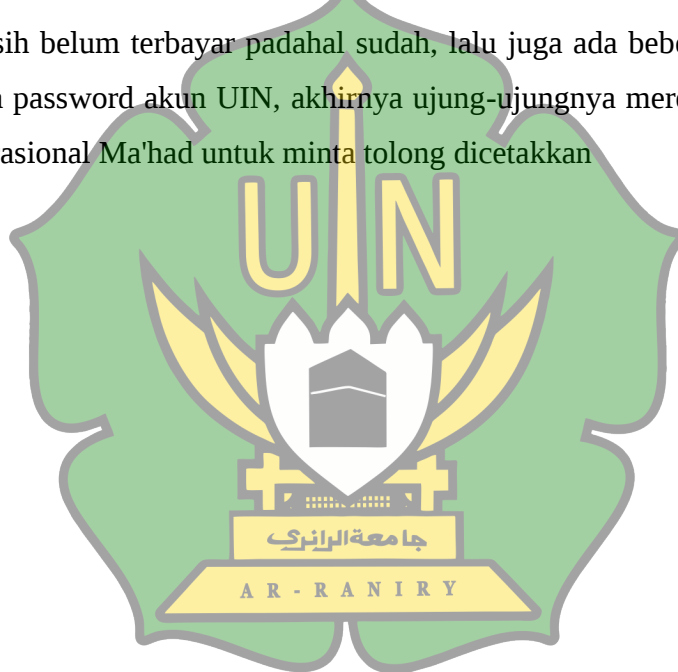
Nah, itu juga menjadi kendala. Saat ini ustadz belum memiliki akses atau portal khusus ke dalam sistem. Jadi, mereka biasanya mencatat absensi dan nilai secara manual, lalu diserahkan ke staf akademik. Nanti staf lagi yang harus merekap dan memindahkannya ke sistem. Proses ini sangat memakan waktu, menimbulkan beban kerja juga, dan kadang membuat publikasi nilai akhir mahasiswa menjadi terlambat.

Peneliti:

Terkait layanan kelulusan mahasiswa, saya mendapat informasi bahwa banyak mahasiswa yang masih datang ke kantor untuk meminta pencetakan sertifikat?

Narasumber :

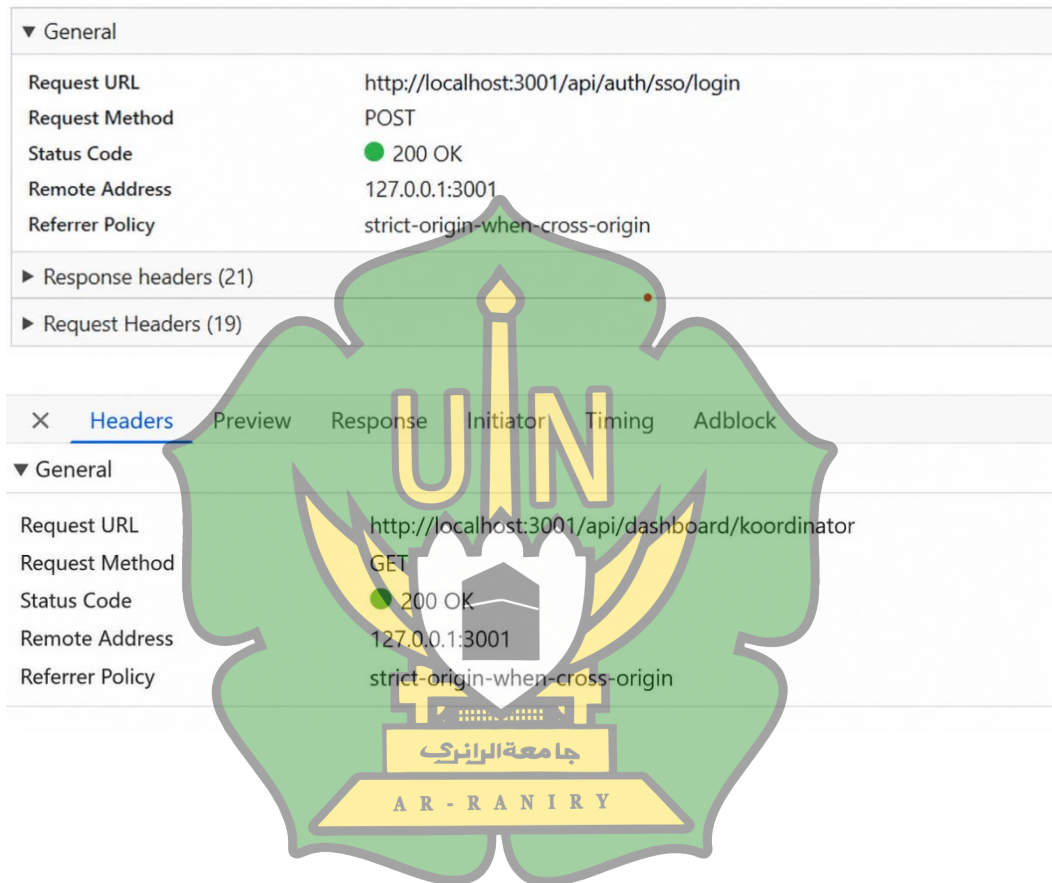
Benar, di sistem yang ada sebenarnya sertifikat bisa dicetak sendiri oleh mahasiswa. Tapi masalahnya ada di kendala sistem yang lama. Karena pada dashboard mereka terdapat kendala dimana sertifikat tidak bisa dicetak karena status pembayaran tertulis masih belum terbayar padahal sudah, lalu juga ada beberapa yang datang karena lupa password akun UIN, akhirnya ujung-ujungnya mereka datang lagi ke kantor operasional Ma'had untuk minta tolong dicetak



Lampiran 2 Bukti Pengujian Black-Box Testing

Lampiran ini memuat bukti lengkap sembilan skenario fungsi vital. Setiap bukti harus memperlihatkan keluaran antarmuka dan informasi eksternal pada tab Network DevTools yang cukup untuk menilai kesesuaian hasil tanpa membuka struktur internal kode.

Lampiran 2.1 BB-K01 Autentikasi dan Otorisasi Koordinator



The screenshot displays the Network tab in Chrome DevTools, showing two API requests. The first request is a POST to the login endpoint, and the second is a GET to the coordinator dashboard endpoint. Both requests resulted in a 200 OK status. A large watermark of the UIN Ar-Raniry logo is overlaid on the image.

Request 1	
Request URL	http://localhost:3001/api/auth/sso/login
Request Method	POST
Status Code	200 OK
Remote Address	127.0.0.1:3001
Referrer Policy	strict-origin-when-cross-origin
▶ Response headers (21)	
▶ Request Headers (19)	

Request 2	
Request URL	http://localhost:3001/api/dashboard/koordinator
Request Method	GET
Status Code	200 OK
Remote Address	127.0.0.1:3001
Referrer Policy	strict-origin-when-cross-origin

Lampiran 2.2 BB-K02 Sinkronisasi Data Mahasantri

✕ Headers Preview Response Initiator Timing Adblock

▼ General

Request URL: http://localhost:3001/api/mahasantri/sync
Request Method: POST
Status Code: 201 Created
Remote Address: 127.0.0.1:3001
Referrer Policy: strict-origin-when-cross-origin

► Response headers (19)

► Request Headers (19)

✕ Headers Preview Response Initiator Timing Adblock

▼ {success: true, message: "Sinkronisasi selesai",...}

▼ data: {totalFetched: 27311, successCount: 24884, failedCount: 2427,...}

► errors: [{nim: "250206005", reason: "Nama/gender tidak lengkap atau tidak valid"},...]

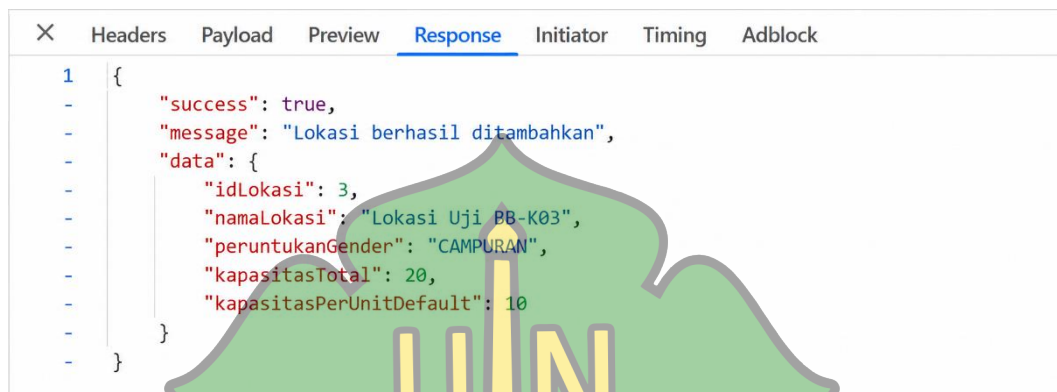
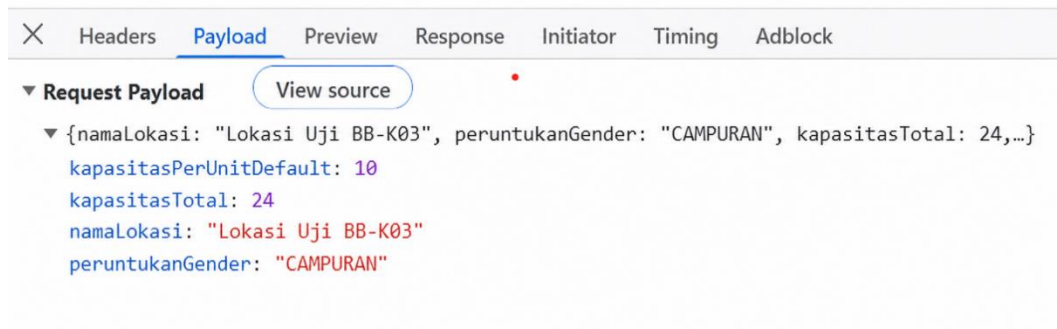
failedCount: 2427
successCount: 24884
totalFetched: 27311

► warnings: [{source: "PLACEMENT",...}]

message: "Sinkronisasi selesai"
success: true



Lampiran 2.3 BB-K03 Pengelolaan Data Master



Lampiran 2.4 BB-K04 Otomasi Uniting Kelas

✕ Headers **Payload** Preview Response Initiator Timing Adblock

▼ **Request Payload** [View source](#)

```
{idPeriode: 37, peruntukanGender: "LAKI_LAKI", jenisProgram: "NON_REGULER"}
  idPeriode: 37
  jenisProgram: "NON_REGULER"
  peruntukanGender: "LAKI_LAKI"
```

✕ Headers Payload **Preview** Response Initiator Timing Adblock

```
{success: true, message: "Uniting otomatis berhasil",...}
  data: {totalMahasantriAssigned: 63, totalMahasantriUnassigned: 68, distribution: [...]}
    distribution: [...]
      0: {idKelas: 25, namaUnit: "Bahasa Arab - Gedung A - Putra - Unit 1", assignedCount:
        assignedCount: 30
        capacityTotal: 30
        capacityUsed: 30
        idKelas: 25
        namaUnit: "Bahasa Arab - Gedung A - Putra - Unit 1"
      1: {idKelas: 26, namaUnit: "Bahasa Arab - Gedung A - Putra - Unit 2", assignedCount:
        assignedCount: 26
        capacityTotal: 26
        capacityUsed: 26
        idKelas: 26
        namaUnit: "Bahasa Arab - Gedung A - Putra - Unit 2"
      2: {idKelas: 24, namaUnit: "Tahsin - Kelompok 0 - Gedung A - Putra - Unit 2", assignedCount:
        assignedCount: 7
        capacityTotal: 30
        capacityUsed: 30
        idKelas: 24
        namaUnit: "Tahsin - Kelompok 0 - Gedung A - Putra - Unit 2"
    totalMahasantriAssigned: 63
    totalMahasantriUnassigned: 68
    message: "Uniting otomatis berhasil"
    success: true
```

Lampiran 2.5 BB-U01 Autentikasi dan Akses Kelas Ustadz

×

Headers

Payload

Preview

Response

Initiator

Timing

Adblock

▼ General

Request URL

http://localhost:3001/api/auth/login

Request Method

POST

Status Code

200 OK

Remote Address

127.0.0.1:3001

Referrer Policy

strict-origin-when-cross-origin

▶ Response headers (21)

▶ Request Headers (19)

×	Headers	Payload	Preview	Response	Initiator	Timing	Adblock
▼ Request Payload View source							
▼ {username: "ustadzbudi", password: "██████████"}							
password: ██████████							
username: "ustadzbudi"							



Lampiran 2.6 BB-U02 Pengelolaan Presensi

▼ General

Request URL	http://localhost:3001/api/presensi
Request Method	POST
Status Code	● 201 Created
Remote Address	127.0.0.1:3001
Referrer Policy	strict-origin-when-cross-origin

✕ Headers Payload Preview Response Initiator Timing Adblock

▼ {success: true, message: "Presensi berhasil dicatat", data: {totalRecorded: 30}}

▶ data: {totalRecorded: 30}

message: "Presensi berhasil dicatat"

success: true



Lampiran 2.7 BB-U03 Pengelolaan Nilai

✕

Headers

Payload

Preview

Response

Initiator

Timing

Adblock

▼ General

Request URL	http://localhost:3001/api/nilai/batch
Request Method	POST
Status Code	● 201 Created
Remote Address	127.0.0.1:3001
Referrer Policy	strict-origin-when-cross-origin

✕

Headers

Payload

Preview

Response

Initiator

Timing

Adblock

▼ {success: true, message: "Nilai batch berhasil diinput",...}

▶ data: {totalProcessed: 1, results: [{idPeserta: 725, nilaiAkhir: 70,...}]}

message: "Nilai batch berhasil diinput"

success: true



Lampiran 2.8 BB-M01 Pengecekan Status Kelulusan

×	Headers	Preview	Response	Initiator	Timing	Adblock
▼ General						
Request URL	http://localhost:3001/api/kelulusan/220705062					
Request Method	GET					
Status Code	200 OK					
Remote Address	127.0.0.1:3001					
Referrer Policy	strict-origin-when-cross-origin					

×	Headers	Preview	Response	Initiator	Timing	Adblock
1 {						
-	"success": true,					
-	"message": "Data kelulusan ditemukan",					
-	"data": {					
-	"nim": "220705062",					
-	"nama": "FERIL FERDIAN",					
-	"jenisKelamin": "L",					
-	"rataRata": 87,					
-	"nilaiHuruf": "A",					
-	"statusNilai": "Sangat Baik",					
-	"statusLulus": true,					
-	"noSertifikat": "4073/Un.08/R/08/2024",					
-	"idSertifikat": "IRZVU38V",					
-	"prodi": "Teknologi Informasi",					
-	"isReguler": false,					
-	"gelombang": "Angkatan XI Gelombang 5",					
-	"tahunAkademik": "Genap 2023/2024",					
-	"nilaiMatakuliahList": [],					
-	"linkSertifikat": "https://serdig.ar-raniry.ac.id/mahad/IRZVU38V"					
-	}					
-	}					



Lampiran 2.9 BB-M02 NIM status kelulusan tidak ditemukan

X Headers Preview Response Initiator Timing Adblock	
▼ General	
Request URL	http://localhost:3001/api/kelulusan/220705001
Request Method	GET
Status Code	● 404 Not Found
Remote Address	127.0.0.1:3001
Referrer Policy	strict-origin-when-cross-origin

