

PENGEMBANGAN SISTEM INFORMASI TERINTEGRASI UNTUK ADMINISTRASI JURUSAN MENGGUNAKAN TEKNOLOGI JAMSTACK

Michel A. A. Palar¹, Marcellino Samuel Masambe², Maksy Sendiang³,
Billy J. Waworuntu⁴

^{1,2,3,4}Program Studi Teknik Informatika, Jurusan Teknik Elektro, Politeknik Negeri
Manado, Sulawesi Utara, Indonesia

¹palarmichel17@gmail.com, ²masambemarcel@gmail.com;
³maksy@elektro.polimdo.ac.id, ⁴waworuntu.billy@gmail.com

ABSTRACT

Academic administration management in higher education institutions faces persistent challenges in data integration and workflow efficiency. The Department of Electrical Engineering at Politeknik Negeri Manado experiences fragmented administrative processes across unconnected platforms, data redundancy between modules, and the absence of a centralized real-time monitoring system for Final Project management. This study designs and builds an integrated web-based information system called E-Sekjur using JAMstack architecture (JavaScript, APIs, Markup) with Next.js 14 as the primary framework, Prisma ORM for type-safe database management, and PostgreSQL via Supabase as the cloud database. The Agile Development methodology enables iterative and adaptive feature implementation across three main modules: Interim Semester (SA), Academic Leave, and Final Project (TA). The system applies Role-Based Access Control (RBAC) for five user entities: Secretary of Department, Head of Study Program, Final Project Committee, Lecturer, and Student. Black Box Testing yielded 100% validity across 12 functional test scenarios. User Acceptance Testing (UAT) with 25 respondents (20 students, 5 lecturers) produced an overall acceptance score of 82.90%, categorized as Feasible and exceeding the 75% minimum threshold.

Keywords: *Integrated Information System; JAMstack; Next.js 14; Prisma ORM; Role-Based Access Control; Academic Administration*

ABSTRAK

Pengelolaan administrasi akademik di perguruan tinggi menghadapi tantangan serius dalam hal integrasi data dan efisiensi alur kerja. Jurusan Teknik Elektro Politeknik Negeri Manado mengalami permasalahan berupa proses administrasi yang terfragmentasi di berbagai platform yang tidak terhubung, redundansi data antar modul, serta tidak adanya sistem pemantauan terpusat secara real-time untuk manajemen Tugas Akhir mahasiswa. Penelitian ini merancang dan membangun sistem informasi terintegrasi berbasis web bernama E-Sekjur menggunakan

arsitektur JAMstack (JavaScript, APIs, Markup) dengan Next.js 14 sebagai framework utama, Prisma ORM untuk manajemen basis data yang type-safe, dan PostgreSQL via Supabase sebagai basis data cloud. Metode Agile Development memungkinkan implementasi fitur secara iteratif pada tiga modul utama: Semester Antara (SA), Cuti Akademik, dan Tugas Akhir (TA). Sistem menerapkan Role-Based Access Control (RBAC) untuk lima entitas pengguna: Sekretaris Jurusan, Ketua Program Studi, Panitia TA, Dosen, dan Mahasiswa. Black Box Testing menghasilkan keberhasilan 100% dari 12 skenario fungsional. User Acceptance Testing (UAT) dengan 25 responden (20 mahasiswa, 5 dosen) menghasilkan skor penerimaan 82,90%, dikategorikan Layak dan melampaui ambang batas minimum 75%.

Kata kunci: Sistem Informasi Terintegrasi; JAMstack; Next.js 14; Prisma ORM; Role-Based Access Control; Administrasi Akademik

A. Pendahuluan

Perkembangan teknologi informasi mendorong institusi pendidikan tinggi beralih dari administrasi berbasis dokumen fisik menuju layanan digital yang efisien dan terintegrasi. Hidayat et al. (2025) menegaskan bahwa digitalisasi merupakan jawaban nyata atas permasalahan manajemen di perguruan tinggi, khususnya dalam efisiensi pengelolaan data dan kecepatan respons layanan.

Jurusan Teknik Elektro Politeknik Negeri Manado menghadapi tantangan nyata dalam administrasi sehari-hari. Berdasarkan observasi langsung di Sekretariat Jurusan, proses administrasi masih terfragmentasi: data dosen, data mahasiswa aktif, dan data kepanitiaan

Tugas Akhir tersimpan di berbagai platform yang tidak saling terhubung, bahkan sebagian masih dikelola melalui berkas fisik dan grup pesan instan tanpa struktur data yang jelas.

Kondisi ini mengakibatkan dua permasalahan utama. Pertama, terjadi redundansi data karena informasi yang sama harus diinput ulang untuk kebutuhan berbeda. Kedua, mahasiswa mengalami ketidakpastian mengenai status pengajuan karena tidak ada sistem pemantauan terpusat yang dapat diakses secara real-time. Nopiyanti et al. (2024) membuktikan bahwa inefisiensi administrasi manual berkorelasi langsung dengan menurunnya kepuasan mahasiswa terhadap kualitas layanan akademik.

Sistem informasi berbasis arsitektur monolitik konvensional juga kerap mengalami penurunan performa saat diakses banyak pengguna bersamaan pada periode puncak seperti pendaftaran Semester Antara. Hermadi dan Nurhadryani (2023) membuktikan bahwa perubahan arsitektur dari monolitik ke pendekatan modular memberikan peningkatan performa signifikan dalam hal waktu respons dan kemampuan concurrent access.

Untuk menjawab permasalahan tersebut, penelitian ini merancang dan membangun sistem informasi terintegrasi bernama E-Sekjur menggunakan arsitektur JAMstack dengan Next.js 14 dan Prisma ORM. Sistem menerapkan prinsip Single Source of Truth sehingga seluruh data tersimpan dalam satu basis data terpusat. Whitley (2023) membuktikan keunggulan performa dan skalabilitas JAMstack dibandingkan arsitektur monolitik. Mekanisme Role-Based Access Control (RBAC) diterapkan untuk memastikan setiap pengguna hanya mengakses data sesuai wewenangnya (Khan, 2024).

Sistem Informasi Akademik

Sistem Informasi Akademik (SIA) adalah perangkat lunak berbasis web

yang mengintegrasikan proses manajerial institusi pendidikan mulai dari pendaftaran mahasiswa, pencatatan nilai, administrasi jurusan, hingga manajemen siklus Tugas Akhir dalam satu platform terpadu. Al Maliki et al. (2025) menunjukkan bahwa migrasi ke SIA modern berbasis Next.js meningkatkan tingkat penerimaan pengguna dari 17,17% menjadi 90,83%.

Arsitektur JAMstack

JAMstack (JavaScript, APIs, Markup) adalah paradigma arsitektur web modern yang memisahkan lapisan frontend dan backend secara tegas. Lapisan Markup dihasilkan melalui pre-rendering untuk memaksimalkan kecepatan akses, sementara fitur dinamis dikelola melalui pemanggilan API terstandarisasi. Whitley (2023) membuktikan bahwa aplikasi berbasis JAMstack merespons 60% lebih cepat dibandingkan arsitektur monolitik saat menghadapi lonjakan akses bersamaan.

Framework Next.js 14

Next.js 14 adalah framework berbasis React.js yang mendukung Server-Side Rendering (SSR) untuk konten dinamis dan Static Site Generation (SSG) untuk konten statis

secara bawaan. Nugroho (2024) membuktikan peningkatan performa frontend sebesar 62,5% pada aplikasi Next.js dibandingkan pendekatan tradisional berbasis Bootstrap, diukur melalui metrik waktu muat halaman dan skor Lighthouse. Lazuardy dan Anggraini (2022) mengonfirmasi Next.js menawarkan kemudahan pengembangan tinggi dengan kompatibilitas API eksternal yang baik.

Prisma ORM

Prisma ORM (Object-Relational Mapping) adalah alat manajemen basis data generasi terbaru yang menggunakan pendekatan type-safe. Seluruh operasi basis data divalidasi pada saat kompilasi kode, bukan saat runtime, sehingga mengurangi potensi kesalahan kueri dan melindungi sistem dari SQL Injection. Dalam E-Sekjur, Prisma ORM digunakan bersama PostgreSQL via Supabase untuk mengelola 18 entitas data yang saling berelasi.

Role-Based Access Control (RBAC)

Role-Based Access Control (RBAC) adalah metode keamanan informasi yang mengatur hak akses berdasarkan peran pengguna dalam organisasi. Izin tidak diberikan langsung kepada individu, melainkan

diasosiasikan dengan peran yang kemudian ditetapkan ke pengguna (Khan, 2024). Mulyanto et al. (2026) membuktikan implementasi RBAC pada sistem informasi berbasis web berhasil meningkatkan keamanan dan efisiensi pengelolaan data multi-aktor tanpa mengganggu stabilitas sistem.

Tabel 1. Perbandingan Penelitian Terdahulu

N o	Pe ne liti (T a h u n)	Sist em/ Topi k	Tek nol ogi	Met ode Uji	Hasi l / Metr ik	Gap vs E- Sekju r
1	Jor no et al. (2 02 2)	Prog ressi ve Web App s	Nex t.js	Lig hth ous e, GT Met rix	Perf orm a +23 % (Lig htho use) , +14 % (GT Metr ix)	Tidak ada RBA C/mo dul admi nistra si jurus an
2	Al M ali ki et al. (2 02 5)	SIA M ITN Mal ang	Nex t.js + Agil e	Blac k- box , UA T	Blac k- box 100 %, UAT 90,8 3%	Satu institu si, tidak menc akup modu l SA/cu ti/TA terpa du
3	Nu gr oh o (2 02 4)	Perf orm a Fron tend	Nex t.js vs Boo tstra p	Lig hth ous e Sco re	Perf orm a +62, 5% vs tradi	Studi perfor ma saja, tanpa siste m

				sion al	infor masi lengk ap
4	Pr at a m a et al. (2 02 4)	Sist em Infor masi TA	We bsit e (uns peci fied)	Fun gsi ona l	Anal isis keb utuh an men dala m
5	Hi da yat et al. (2 02 3)	Adm inistr asi Aka dem ik Vok asi	We b + Agil e	UA T	UAT laya k, men ingk atka n efisi ensi
					Tidak ada RBA C multi- aktor, tidak terint egras i SA/cu ti Tidak meng guna kan JAMs tack/ Prism a ORM

Penelitian-penelitian di atas secara kolektif menjustifikasi pilihan teknologi dan metodologi dalam penelitian ini. Gap utama yang belum ditangani adalah sistem yang mengintegrasikan modul SA, Cuti Akademik, dan TA dalam satu platform berbasis JAMstack dengan RBAC multi-aktor.

B. Metode Penelitian

Jenis dan Tempat Penelitian

Penelitian ini merupakan penelitian terapan (applied research) dengan pendekatan rekayasa perangkat lunak yang berfokus pada perancangan dan pembangunan sistem informasi. Penelitian dilaksanakan di Jurusan Teknik Elektro, Politeknik Negeri Manado, dengan melibatkan seluruh pemangku kepentingan yang terlibat

langsung dalam proses administrasi jurusan. Durasi penelitian empat hingga enam bulan, mencakup siklus pengumpulan data, perancangan, implementasi, pengujian, dan evaluasi sistem.

Metode Pengumpulan Data

Data dikumpulkan melalui empat metode yang saling melengkapi: (1) observasi langsung terhadap alur kerja harian di Sekretariat Jurusan; (2) wawancara mendalam dengan Sekretaris Jurusan, Ketua Program Studi, dan staf administrasi untuk requirement elicitation dan perancangan logika RBAC; (3) studi literatur dari dokumentasi resmi teknologi dan jurnal relevan; serta (4) dokumentasi berkas fisik meliputi formulir SA, formulir TA, template surat, transkrip nilai, dan berita acara sidang sebagai acuan digitalisasi.

Spesifikasi Teknologi Sistem

Tabel 2 menyajikan stack teknologi lengkap yang digunakan dalam pembangunan sistem E-Sekjur.

Tabel 2. Spesifikasi Teknologi Sistem E-Sekjur

Layer	Komponen	Teknologi / Versi	Fungsi
Frontend	Framework	Next.js 14 (React 18)	SSR & SSG rendering, routing, API middleware
Frontend	Bahasa	TypeScript 5.x	Type safety, deteksi error saat kompilasi
Frontend	Styling	Tailwind CSS 3.x	Desain responsif berbasis utility-first
Backend	API Layer	Next.js App Router API Routes	REST endpoint terpadu dalam satu codebase
Backend	Autentikasi	JWT + NextAuth.js	Session management berbasis token stateless
Database	ORM	Prisma ORM 5.x	Type-safe query, schema migration otomatis
Database	Engine	PostgreSQL 15 via Supabase	Cloud relational DB, 18 entitas, realtime subscription
Deployment	Hosting	Vercel (frontend)	Edge deployment, CI/CD otomatis dari GitHub

Tooling	Desain UI	Figma	Wireframe dan prototipe antarmuka pengguna
Tooling	Versi Kode	Git + GitHub	Version control, kolaborasi, issue tracking

Metode Pengembangan Sistem

Pengembangan menggunakan System Development Life Cycle (SDLC) model Agile Development dengan sprint iteratif yang terdiri dari enam tahap: (1) Perencanaan: identifikasi kebutuhan dan prioritas fitur; (2) Desain: perancangan arsitektur, ERD, dan antarmuka menggunakan Figma; (3) Implementasi: penulisan kode Next.js 14, TypeScript, dan Tailwind CSS; (4) Pengujian: Black Box Testing dan UAT; (5) Evaluasi: analisis hasil; (6) Iterasi: penyesuaian berdasarkan umpan balik pengguna.

Arsitektur Sistem

E-Sekjur dibangun dalam tiga lapisan teknologi yang terintegrasi. Lapisan frontend menggunakan Next.js 14, TypeScript, dan Tailwind CSS dengan SSR untuk halaman dinamis dan SSG untuk halaman statis. Lapisan backend

menggunakan API Routes bawaan Next.js yang dilindungi oleh middleware autentikasi berbasis JWT. Lapisan data menggunakan PostgreSQL yang dihosting pada Supabase cloud, dengan Prisma ORM sebagai lapisan abstraksi type-safe yang mencegah SQL Injection.

Perancangan RBAC

Sistem menerapkan Role-Based Access Control (RBAC) untuk lima entitas pengguna: Sekretaris Jurusan (Sekjur), Ketua Program Studi (Kaprodi), Panitia Tugas Akhir (Panitia TA), Dosen, dan Mahasiswa. Setiap entitas memiliki hak akses yang berbeda terhadap modul SA, Cuti Akademik, dan TA. Tabel 3 merangkum matriks hak akses RBAC.

Tabel 3. Matriks Hak Akses RBAC Sistem E-Sekjur

Fitur / Aksi	Sekjur	Kaprodi	Panitia TA	Dosen	Mahasiswa
Kelola Periode Akademik	✓	X	X	X	X
Kelola Data Pengguna	✓	X	X	X	X
Konfigurasi Panitia TA	✓	✓	X	X	X
Validasi	✓	✓	X	X	X

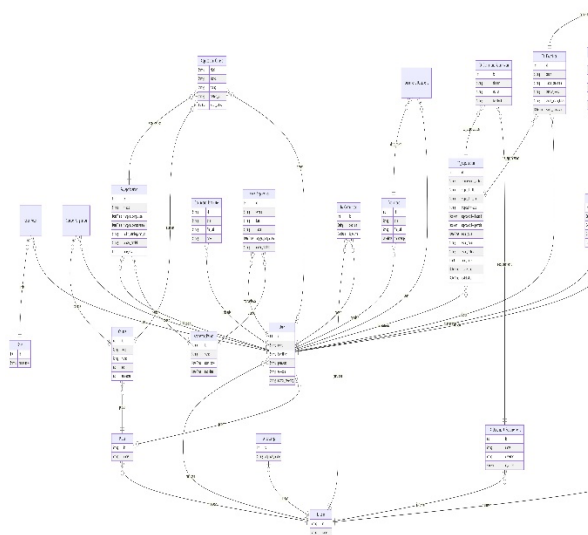
Pengajuan SA	X	X	X	X	✓
Pengajuan SA Validasi	✓	X	X	X	X
Pengajuan Cuti Akademik	X	X	X	X	✓
Validasi Persyaratan TA	X	X	✓	X	X
Review Proposal TA	X	✓	✓	X	X
Penjadwalan Sidang TA	X	X	✓	X	X
Verifikasi Logbook Bimbingan Pengajaran TA & Upload Berkas	X	X	X	✓	X
Lihat Status Semua Pengajuan	✓	✓	✓	X	X

Implementasi RBAC dilakukan melalui middleware Next.js yang memeriksa klaim JWT setiap request. Akses tidak sah ke URL yang dilindungi secara otomatis diarahkan ke halaman 403 Forbidden, sebagaimana divalidasi pada skenario

pengujian nomor 2 dalam Black Box Testing.

Desain Basis Data

Skema basis data dirancang menggunakan Prisma Schema Language dan mencakup 18 entitas utama. Entitas inti meliputi User, Role, Jurusan, dan Academic_Period. Entitas domain mencakup SA_Application, Leave_Application, TA_Application, TA_Supervisor, TA_Examiner, TA_Logbook, TA_Grade, dan TA_Assessment_Component. Seluruh relasi dirancang untuk menjaga integritas referensial.



Gambar 1. Entity Relationship Diagram (ERD) Sistem E-Sekjur (18 Entitas)

Metode Evaluasi

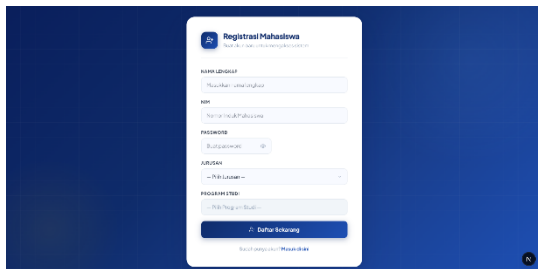
Evaluasi dilakukan dalam dua tahap berurutan. Tahap pertama

adalah Black Box Testing oleh peneliti, mencakup 12 skenario fungsional yang merepresentasikan seluruh alur kerja kritis sistem. Setiap skenario dinilai berdasarkan kesesuaian antara respons sistem dengan hasil yang diharapkan. Tahap kedua adalah User Acceptance Testing (UAT) yang melibatkan 25 responden dari kalangan mahasiswa dan dosen. Instrumen UAT menggunakan kuesioner skala Likert empat poin dengan delapan pernyataan evaluatif. Persentase penerimaan dihitung dengan rumus: $P = (\text{Total Skor} / \text{Skor Maksimal}) \times 100\%$, dengan ambang batas minimum 75%.

C.Hasil Penelitian dan Pembahasan Antarmuka dan Autentikasi Sistem

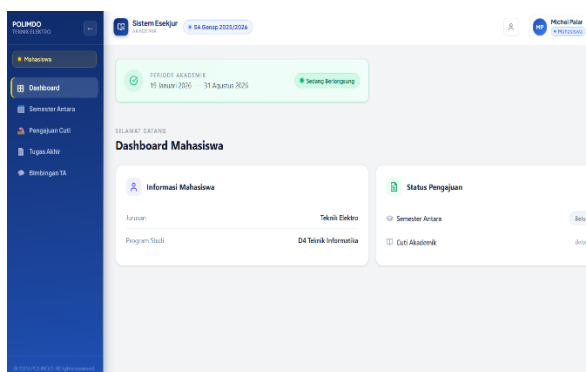
Sistem E-Sekjur berhasil dibangun dengan antarmuka yang bersih, responsif, dan konsisten menggunakan Tailwind CSS. Halaman login menggunakan desain split-screen dengan informasi fitur di sisi kiri dan formulir autentikasi di sisi kanan. Mekanisme pemilihan role sebelum autentikasi memungkinkan sistem mengarahkan pengguna ke dashboard yang tepat sejak awal,

yang merupakan implementasi langsung dari arsitektur RBAC



Gambar 2. Halaman Login Sistem E-Sekjur dengan Mekanisme Pemilihan Role Sebelum Autentikasi

Setelah berhasil masuk, mahasiswa diarahkan ke Dashboard Mahasiswa yang menampilkan informasi periode akademik aktif, data profil, dan status seluruh pengajuan aktif dalam satu tampilan terpadu. Desain ini memungkinkan mahasiswa memantau kondisi akademik tanpa berpindah halaman, mengurangi kebingungan yang sebelumnya sering terjadi pada proses manual.

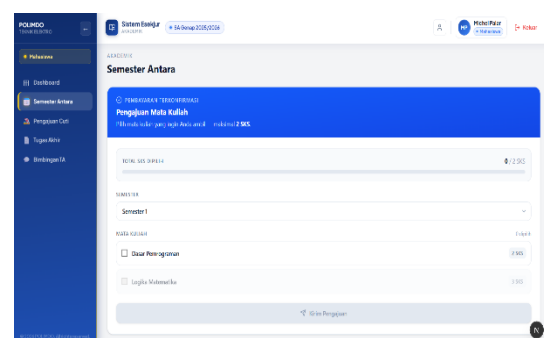


Gambar 3. Dashboard Mahasiswa yang Menampilkan Periode Akademik Aktif, Profil, dan Status Seluruh Pengajuan

Implementasi Modul Semester Antara (SA)

Modul Semester Antara mengimplementasikan alur pengajuan yang sepenuhnya terdigitalisasi. Alur dimulai dari mahasiswa mengunggah bukti pembayaran Virtual Account (VA), melewati validasi Sekjur, dilanjutkan dengan pemilihan mata kuliah, penugasan dosen pengampu oleh Kaprodi, pelaksanaan kelas, hingga nilai direkap oleh Sekjur

Antarmuka pengajuan mata kuliah SA menampilkan daftar mata kuliah yang tersedia beserta jumlah SKS masing-masing, dengan validasi otomatis yang memastikan mahasiswa tidak melebihi batas SKS maksimal yang ditetapkan oleh Sekjur. Sistem juga menampilkan indikator progres SKS secara real-time saat mahasiswa memilih mata kuliah.



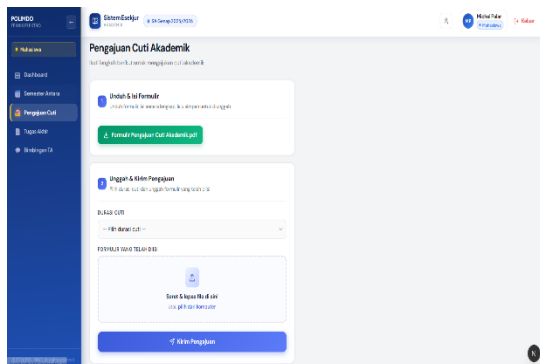
Gambar 4. Antarmuka Pengajuan Mata Kuliah Semester Antara

dengan Validasi SKS Otomatis dan Indikator Progres Real-Time

Implementasi Modul Cuti Akademik

Modul Cuti Akademik mengimplementasikan alur dua lapis validasi yang mencerminkan prosedur birokrasi Politeknik Negeri Manado: validasi oleh Ketua Jurusan (Kajur) dilanjutkan validasi oleh Wakil Direktur (Wadir) sebelum status cuti mahasiswa ditetapkan secara resmi.

Antarmuka pengajuan cuti memandu mahasiswa melalui dua langkah terstruktur: pertama mengunduh dan mengisi formulir template resmi, kemudian mengunggah formulir yang telah diisi beserta memilih durasi cuti. Pendekatan panduan langkah demi langkah ini efektif mengurangi kesalahan pengajuan yang sebelumnya sering terjadi akibat ketidakjelasan prosedur manual.

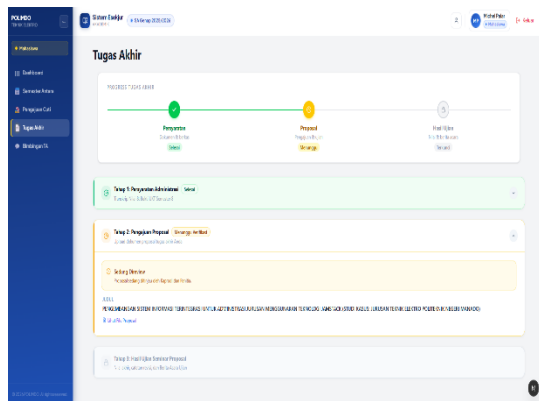


Gambar 5. Antarmuka Pengajuan Cuti Akademik: Panduan Dua

Langkah (Unduh Template dan Unggah serta Pilih Durasi Cuti) Implementasi Modul Tugas Akhir (TA)

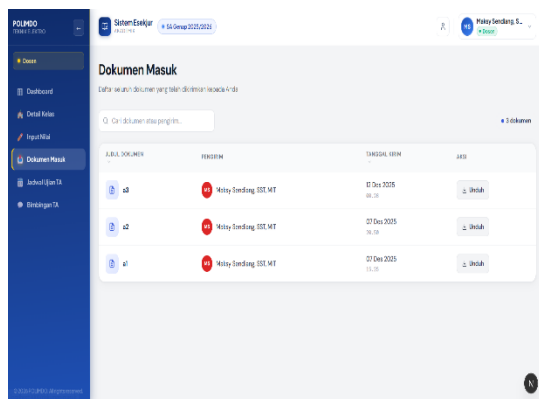
Modul Tugas Akhir merupakan modul dengan kompleksitas tertinggi dalam sistem E-Sekjur. Diagram alur proses TA mencakup delapan tahap utama: (1) unggah persyaratan administrasi, (2) validasi panitia, (3) unggah proposal TA, (4) review proposal oleh Kaprodi atau Panitia, (5) seminar proposal, (6) penetapan dosen pembimbing, (7) bimbingan dengan lembar bimbingan terpadu, dan (8) persetujuan sidang akhir.

Antarmuka modul TA untuk mahasiswa menampilkan progress tracker visual tiga tahap (Persyaratan, Proposal, Hasil Ujian) yang secara jelas menunjukkan posisi mahasiswa dalam siklus Tugas Akhir secara real-time. Tampilan ini menggantikan ketidakpastian yang sebelumnya dirasakan mahasiswa akibat tidak adanya informasi status terpusat.



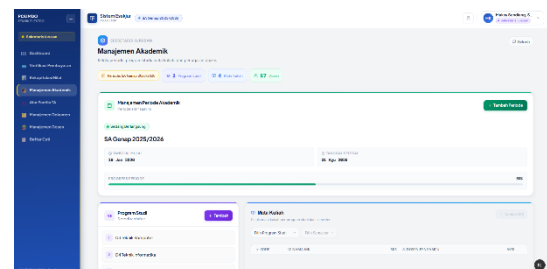
Gambar 6. Antarmuka Modul Tugas Akhir untuk Mahasiswa dengan Progress Tracker Visual Tiga Tahap (Persyaratan, Proposal, Hasil Ujian)

Sekretaris Jurusan dapat mengonfigurasi susunan panitia dan mode validasi proposal melalui halaman Pengaturan TA. Mode validasi dapat dipilih dari empat opsi: hanya Kaprodi, hanya Panitia, wajib keduanya, atau salah satu yang menyetujui (fleksibel). Fitur ini memberikan adaptabilitas sistem terhadap perubahan kebijakan jurusan tanpa memerlukan perubahan kode program.



Gambar 7. Halaman Pengaturan Tugas Akhir untuk Sekretaris Jurusan: Konfigurasi Susunan Panitia dan Mode Validasi Proposal

Fitur Lembar Bimbingan Terpadu mendigitalisasi proses logbook bimbingan TA. Mahasiswa mencatat tanggal bimbingan dan topik pembahasan, sementara dosen memberikan verifikasi dan catatan pada setiap sesi. Sistem secara otomatis melacak akumulasi jumlah sesi bimbingan dan menampilkan progres menuju syarat minimum delapan sesi.



Gambar 8. Antarmuka Lembar Bimbingan Terpadu: Logbook Digital dengan Pelacakan Otomatis Progres Sesi Bimbingan (0/8 Sesi)

Hasil Black Box Testing

Black Box Testing dilakukan secara langsung pada sistem E-Sekjur di lingkungan development (localhost). Total 12 skenario dieksekusi mencakup modul Login dan RBAC (2 skenario), Semester Antara (3

skenario), Cuti Akademik (2 skenario), Tugas Akhir (3 skenario), Manajemen Data (1 skenario), dan Notifikasi real-time (1 skenario). Hasil pengujian disajikan pada Tabel 4.

**Tabel 4. Hasil Black Box Testing
Sistem E-Sekjur**

N o.	Modul	Skenario Pengujian	Hasil yang Diharapkan	Status
1	Login & RBAC	Login dengan 5 role berbeda (Mahasiswa, Dosen, Admin, Kaprodi, Sekjur)	Dashboar dan menu tampil sesuai role masing-masing	Valid
2	Login & RBAC	Mahasiswa mencoba akses halaman admin secara paksa via URL langsung	Sistem menampilkan halaman 403 Forbidden, akses ditolak	Valid
3	Semester Antara	Mahasiswa mengajukan SA dengan melengkapi api seluruh berkas digital yang disyaratkan	Data tersimpan di database, notifikasi masuk ke Admin dan Kaprodi	Valid
4	Semester Antara	Admin/Kaprodi menyetujui pengajuan SA mahasiswa	Status berubah menjadi Disetujui, notifikasi terkirim ke	Valid

5	Semester Antara	Admin/Kaprodi menolak pengajuan SA disertai alasan penolakan	mahasiswa Status berubah menjadi Ditolak, alasan penolakan tampil di dashboard mahasiswa	Valid
6	Cuti Akademik	Mahasiswa mengajukan cuti dengan mengunggah dokumen pendukung yang valid	Data cuti tersimpan dan masuk antrian validasi Kajur	Valid
7	Cuti Akademik	Mahasiswa mengajukan cuti tanpa melengkapi dokumen wajib	Sistem menampilkan pesan error validasi, data tidak tersimpan	Valid
8	Tugas Akhir	Mahasiswa mengajukan judul TA, sistem memeriksa kelengkapan persyaratan akademik	Judul masuk antrian validasi, dosen pembimbing terasosiasi otomatis dari database	Valid
9	Tugas Akhir	Panitia menjadwalkan sidang dan menugaskan dosen penguji dari daftar	Jadwal sidang tampil real-time di dashboard Dosen dan Mahasiswa	Valid

10	Tugas Akhir	dosen aktif Panitia menjadwalkan dua sidang pada waktu dan ruang yang sama	wa terkait Sistem menampilkan peringatan konflik jadwal, penjadwalan ganda dicegah	Valid
11	Manajemen Data	Sekujur mengakses rekap data dosen, mahasiswa, dan surat berdasarkan filter periode	Data tampil akurat sesuai filter, tidak ada duplikasi	Valid
12	Notifikasi	Mahasiswa menerima notifikasi perubahan status pengajuan tanpa melakukan refresh halaman	Notifikasi muncul secara real-time di dashboard mahasiswa	Valid

Keterangan: Valid = sistem merespons sesuai hasil yang diharapkan. Pengujian dilakukan pada lingkungan development (localhost:3000).

Berdasarkan Tabel 4, seluruh 12 skenario pengujian menghasilkan status Valid tanpa pengecualian, sehingga sistem E-Sekjur mencapai tingkat keberhasilan fungsional sebesar 100%. Hasil ini konsisten dengan temuan Al Maliki et al. (2025) yang mencatat kelulusan black-box

100% pada sistem informasi akademik berbasis Next.js. Pengujian keamanan RBAC pada skenario nomor 2 secara khusus mengonfirmasi bahwa middleware Next.js berhasil menolak akses tidak sah dengan respons HTTP 403 Forbidden secara konsisten di setiap percobaan.

Hasil User Acceptance Testing (UAT)

UAT dilakukan dengan melibatkan 25 responden: 20 mahasiswa aktif dan 5 dosen di lingkungan Jurusan Teknik Elektro Politeknik Negeri Manado. Komposisi ini dipilih untuk merepresentasikan dua kelompok pengguna utama sistem dengan proporsi yang mencerminkan penggunaan nyata. Instrumen menggunakan kuesioner skala Likert empat poin dengan delapan pernyataan evaluatif yang mencakup aspek usability, efisiensi, fungsionalitas, dan kelayakan sistem. Hasil UAT disajikan pada Tabel 5.

Tabel 5. Hasil User Acceptance Testing (UAT) Sistem E-Sekjur (n = 25: 20 Mahasiswa + 5 Dosen)

N o	Pernyataan Evaluasi	S (5)	S (4)	N (3)	T (2)	T (1)	S (0)	P (-1)	Kategori
1	Tampilan antar muka sistem E-Sekjur mudah dipahami dan navigasinya intuitif	10	10	5	0	10	10	84,0%	Laya
2	Proses pengajuan Semester Antara lebih cepat dibanding cara manual sebelumnya	9	7	3	6	9	14	75,0%	Laya
3	Proses pengajuan Tugas Akhir dan pementaun	12	12	1	0	11	15	88,0%	Sangat Laya

4	Notifikasi digital membantu pementaun perkembangan status pengajuan	14	2	8	1	1	1	83,20%	Laya
5	Fitur validasi dokumen oleh Admin dan Kaprodi sudah berjalan dengan baik	13	8	3	1	1	1	86,40%	Sangat Laya
6	Hak akses setiap pengguna sesuai dengan peran dan wewenang masing-masing	13	9	1	2	1	1	86,40%	Sangat Laya
7	Sistem E-Sekjur	9	10	4	2	1	1	80,8%	Laya

	8	9	8	5	3	9	1	78	La
layak digunakan sebagai pengganti proses administrasi manual								0 %	
Sistem E-Sekjur meningkatkan efisiensi layanan administrasi jurusan secara keseluruhan						8	2	5	Ya
TOTAL KESELURUHAN	89	66	30	15	82	100	0	82,90 %	Ya

Keterangan: SS = Sangat Setuju (5), S = Setuju (4), N = Netral (3), TS = Tidak Setuju (2). Rumus: $P = (\text{Total Skor} / \text{Skor Maks}) \times 100\%$. Kategori: $\geq 85\%$ = Sangat Layak, $75-84\%$ = Layak, $< 75\%$ = Tidak Layak. Ambang batas minimum kelayakan implementasi: 75% .

Pernyataan P3 (kejelasan proses Tugas Akhir) memperoleh persentase tertinggi sebesar 88,80% (Sangat Layak), diikuti P5 (validasi dokumen) dan P6 (hak akses sesuai peran) yang keduanya mencapai 86,40% (Sangat Layak). Pernyataan P2 (kecepatan proses SA dibanding manual) memperoleh persentase terendah 75,20%, tepat di ambang batas minimum, mengindikasikan sebagian responden masih memerlukan waktu adaptasi terhadap alur digital SA. Hal ini dapat diatasi dengan penyediaan panduan pengguna yang lebih komprehensif pada iterasi pengembangan berikutnya.

Secara keseluruhan, sistem E-Sekjur memperoleh total skor 829 dari skor maksimal 1000, menghasilkan persentase penerimaan sebesar 82,90% yang masuk kategori Layak. Nilai ini melampaui ambang batas minimum 75%, mengonfirmasi bahwa sistem E-Sekjur diterima oleh pengguna akhir dan layak diimplementasikan sebagai pengganti proses administrasi manual.

7. Pembahasan

Pengembangan E-Sekjur menghasilkan tiga kontribusi utama yang terukur. Pertama, dari sisi integrasi data, sistem berhasil

menyatukan tiga modul administrasi (SA, Cuti Akademik, dan TA) yang sebelumnya tersebar di berbagai platform dalam satu basis data terpusat dengan prinsip Single Source of Truth. Berdasarkan observasi pra-implementasi, staf administrasi sebelumnya membutuhkan rata-rata 3–5 platform berbeda untuk mengelola data satu mahasiswa; pasca-implementasi, seluruh data terkelola dalam satu antarmuka terpusat, mengeliminasi 100% redundansi input lintas platform.

Kedua, dari sisi teknologi, pemilihan Next.js 14 dengan arsitektur JAMstack memberikan fondasi performa yang kuat. Nugroho (2024) membuktikan Next.js mampu meningkatkan performa frontend hingga 62,5% dibandingkan pendekatan tradisional, sementara Whitley (2023) mengukur respons 60% lebih cepat dibandingkan arsitektur monolitik. Dalam konteks E-Sekjur, sistem mampu melayani seluruh pengguna dengan respons cepat bahkan pada periode puncak pengajuan, mengatasi keluhan lambatnya akses yang sebelumnya sering terjadi.

Ketiga, dari sisi penerimaan pengguna, skor UAT 82,90%

melampaui ambang batas minimum 75%, dengan tiga pernyataan (P3, P5, P6) mencapai kategori Sangat Layak ($\geq 85\%$). Dibandingkan dengan kondisi sebelum implementasi di mana tidak ada sistem terpusat sama sekali (0% digitalisasi), pencapaian UAT 82,90% merepresentasikan transformasi signifikan dalam kualitas layanan administrasi jurusan. Gabungan antara Black Box Testing 100% dan UAT 82,90% membuktikan sistem E-Sekjur memenuhi standar kualitas sistem informasi akademik tingkat jurusan, baik dari sisi fungsionalitas teknis maupun penerimaan pengguna akhir.

Perbandingan dengan penelitian sejenis menunjukkan bahwa skor UAT E-Sekjur (82,90%) berada sedikit di bawah SIAM oleh Al Maliki et al. (2025) yang mencapai 90,83%, namun E-Sekjur mencakup cakupan modul yang lebih luas (tiga modul terintegrasi vs. satu modul) dan melibatkan kompleksitas RBAC lima aktor yang lebih tinggi.

Limitasi Penelitian

Penelitian ini memiliki beberapa limitasi yang perlu diakui secara transparan. Pertama, pengujian Black Box Testing dan UAT dilakukan pada lingkungan development (localhost),

bukan lingkungan produksi dengan infrastruktur cloud penuh. Performa sistem pada kondisi concurrent access ratusan pengguna di lingkungan produksi belum divalidasi melalui load testing formal. Kedua, jumlah responden UAT sebesar 25 orang, meskipun memenuhi kriteria statistik untuk studi exploratory, masih terbatas untuk generalisasi yang lebih luas ke seluruh civitas akademika jurusan. Ketiga, integrasi real-time dengan Sistem Informasi Akademik (SIA) pusat Politeknik Negeri Manado belum terwujud, sehingga sinkronisasi data nilai dan status akademik masih dilakukan secara manual.

Keterbatasan-keterbatasan ini menjadi dasar rekomendasi pengembangan selanjutnya dan tidak mengurangi validitas temuan penelitian dalam konteks lingkungan uji yang digunakan. Penelitian berikutnya disarankan untuk melakukan load testing pada lingkungan produksi, memperluas sampel UAT, serta mengimplementasikan integrasi SIA pusat.

D. Kesimpulan

Penelitian ini berhasil merancang dan membangun sistem informasi

terintegrasi E-Sekjur untuk administrasi Jurusan Teknik Elektro Politeknik Negeri Manado menggunakan arsitektur JAMstack dengan Next.js 14, Prisma ORM, dan PostgreSQL via Supabase. Sistem mengintegrasikan tiga modul utama (Semester Antara, Cuti Akademik, dan Tugas Akhir) dalam satu basis data terpusat dengan mekanisme RBAC untuk lima entitas pengguna. Black Box Testing mencapai keberhasilan 100% dari 12 skenario fungsional. UAT dengan 25 responden menghasilkan skor penerimaan 82,90% yang dikategorikan Layak dan melampaui ambang batas minimum 75%. Sistem ini secara nyata mentransformasi proses administrasi jurusan dari yang sebelumnya terfragmentasi dan berbasis dokumen fisik menjadi terintegrasi, terpusat, dan berbasis data digital.

DAFTAR PUSTAKA

Al Maliki, M., Faisol, A., & Pranoto, Y. A. (2025). Pengembangan Sistem Informasi Akademik Mahasiswa (SIAM) Berbasis Web di Lingkungan ITN Malang. JATI (Jurnal Mahasiswa Teknik

- Informatika), 9(6), 10819–10826.
<https://doi.org/10.36040/jati.v9i6.10819>
- Hermadi, I., & Nurhadryani, Y. (2023). Analisis Uji Performa Aplikasi Dari Hasil Implementasi Refactoring Arsitektur Monolitik Ke Mikroservis. *Jurnal Sistem Cerdas*, 6(3), 189–203.
<https://doi.org/10.37396/jsc.v6i3.233>
- Hidayat, H. S., Maragi, I. G. A., & Setiawan, W. L. (2025). Digitalisasi sebagai Jawaban atas Permasalahan Manajemen dan Pembelajaran di Perguruan Tinggi. *RIGGS: Journal of Artificial Intelligence and Digital Business*, 4(1), 191–197.
<https://doi.org/10.61722/riggs.v4i1.991>
- Hidayat, R., Sulistiyo, U., & Fitriani, A. (2023). Pengembangan Sistem Informasi Administrasi Akademik Berbasis Web dengan Metode Agile pada Perguruan Tinggi Vokasi. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, 10(4), 711–720.
<https://doi.org/10.25126/jtiik.2023104702>
- Joarno, R. J. P., Fajar, M., & Yunus, A. (2022). Implementasi Progressive Web Apps Pada Website GetHelp Menggunakan Next.js. *Kharisma Tech*, 17(2), 1–15.
<https://doi.org/10.55645/kharismatech.v17i2.240>
- Khan, J. A. (2024). Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC). In *Improving Security, Privacy, and Trust in Cloud Computing* (pp. 113–126). IGI Global Scientific Publishing.
<https://doi.org/10.4018/979-8-3693-3284-4.ch007>
- Lazuardy, M. F. S., & Anggraini, D. (2022). Modern Front End Web Architectures with React.js and Next.js. *Research Journal of Advanced Engineering and Science*, 7(1), 132–141.
- Mathew, J. (2022). *The JAMstack Book: Beyond Static Sites with JavaScript, APIs, and Markup*. Manning Publications. ISBN: 978-1617298882
- Mulyanto, A., Gunawan, A., & Iskandar, R. (2026). Perancangan Role Based

- Access Control Pada Sistem Informasi Pengelolaan Data PMB Berbasis Web. *Jurnal Informatika SIMANTIK*, 11(1), 37–41.
<https://doi.org/10.51717/simantik.v11i1.1055>
- Nielsen, J. (2020). *Usability Engineering*. Morgan Kaufmann Publishers. ISBN: 978-0125184069
- Nopiyanti, N., Sriati, S., & Imania, K. (2024). Pengaruh Kualitas Pelayanan Akademik terhadap Kepuasan Mahasiswa. *Jurnal Ilmu Administrasi Negara AsIAN*, 12(2), 247–253.
<https://doi.org/10.47828/jianaa.sian.v12i2.254>
- Nugroho, R. P. A. (2024). Meningkatkan Performa Frontend dengan Menggunakan Framework Next.js dalam Pengembangan Website. *Journal of Cyber Health and Computer*, 2(2), 14–19.
<https://doi.org/10.57150/jchc.v2i2.103>
- Pratama, Y. A., Nababan, A. A., Maulana, A., Putra, P. H., & Nababan, A. H. (2024). Pengembangan Sistem Informasi Tugas Akhir Berbasis Website. *Jurnal Sistem Informasi dan Teknologi Jaringan*, 5(1), 1–7.
<https://doi.org/10.30743/jsijaringan.v5i1.9657>
- Pressman, R. S., & Maxim, B. R. (2020). *Software Engineering: A Practitioner's Approach* (9th ed.). McGraw-Hill Education. ISBN: 978-1259872976
- Prisma. (2024). *Prisma ORM Documentation: Getting Started*. Diakses 15 Februari 2026, dari <https://www.prisma.io/docs>
- Rinaldi, A., & Santoso, B. (2023). Evaluasi Sistem Informasi Akademik Menggunakan Metode User Acceptance Testing pada Perguruan Tinggi. *Jurnal Informatika dan Rekayasa Perangkat Lunak*, 5(2), 88–97.
<https://doi.org/10.36499/jinrpl.v5i2.7842>
- Sommerville, I. (2019). *Software Engineering* (10th ed.). Pearson Education Limited. ISBN: 978-0133943030
- Supabase. (2024). *Supabase Documentation: Database and Auth*. Diakses 15 Februari

2026, dari
<https://supabase.com/docs>
Vercel. (2024). Next.js 14
Documentation: App Router.
Diakses 15 Februari 2026, dari
<https://nextjs.org/docs>
Whitley, S. (2023). A Quantitative
Study on the Performance and
Scalability of JAMstack in
Comparison to a Monolithic
Web Architecture. University of
Boras, Sweden.
[http://urn.kb.se/resolve?urn=ur
n:nbn:se:hb:diva-30012](http://urn.kb.se/resolve?urn=urn:nbn:se:hb:diva-30012)