

PENERAPAN AGILE SCRUM DAN ALGORITMA ROUND ROBIN PADA SISTEM MANAJEMEN KERJA PRAKTIK TEKNIK INFORMATIKA ITATS

M Sultan Fadli, Dian Puspita Hapsari

Teknik Informatika, Fakultas Teknik Elektro dan Teknologi Informasi,
Institut Teknologi Adhi Tama Surabaya

Email: msfadliiii@gmail.com, dian.puspita@itats.ac.id

Diajukan: 12-08-2026

Direvisi: 07-09-2026

Disetujui: 11-09-2026

Diterbitkan: 30-09-2026

Abstract. *The Internship is a crucial phase that bridges students with the professional world; however, its implementation at the Informatics Engineering Programme at ITATS faces various administrative challenges. These issues include the lack of integration between student and lecturer data and the campus information centre, a supervision process reliant on manual communication via WhatsApp and paper logbooks, a registration fee recapitulation prone to calculation errors, the absence of an automated notification system, and an unstructured supervisor rotation mechanism that risks creating an unequal distribution of supervision workloads among lecturers. This study aims to develop an internship management system integrated with an Academic Application Programming Interface (API) simulation and equipped with a Round Robin algorithm to automate the equitable distribution of supervising lecturers, utilising the Agile Scrum framework to ensure development remains adaptive to changing functional requirements and academic policies. The research methodology begins with observation and interviews with the Internship Coordinator to compile the Product Backlog, followed by system design using UML and database design. The development process passes through four Sprint cycles, with Scrum roles adapted for a solo developer, encompassing Sprint Planning, Execution, Review, and Retrospective. System testing applied Black Box Testing across 19 test scenarios and a User Acceptance Test (UAT) involving 27 respondents. The testing results demonstrate a 100% success rate for Black Box Testing, whilst the UAT yielded a feasibility score of 94.50%, categorised as Highly Feasible. The system is proven capable of automating registration, supervisor allocation, and Internship administration more efficiently, transparently, and adaptively.*

Keywords: Agile Scrum, API Simulation, Internship, Round Robin, Management System

Abstrak. *Kerja Praktik (KP) merupakan tahapan penting yang menghubungkan mahasiswa dengan dunia industri, namun pelaksanaannya di Program Studi Teknik Informatika ITATS masih menghadapi berbagai kendala administratif. Permasalahan tersebut meliputi belum terintegrasinya data mahasiswa dan dosen dengan pusat informasi kampus, proses bimbingan yang masih bergantung pada komunikasi manual melalui WhatsApp dan logbook kertas, rekapitulasi dana pendaftaran yang rawan kesalahan hitung, ketiadaan sistem notifikasi otomatis, serta mekanisme rotasi dosen pembimbing yang belum terstruktur sehingga berpotensi menimbulkan ketimpangan beban bimbingan antar dosen. Penelitian ini bertujuan mengembangkan sistem manajemen Kerja Praktik yang terintegrasi dengan Simulasi Application Programming Interface (API) Akademik serta dilengkapi algoritma Round Robin untuk mengotomatisasi pemerataan distribusi dosen pembimbing, dengan menerapkan kerangka kerja Agile Scrum agar pengembangan tetap adaptif terhadap perubahan kebutuhan fungsional dan kebijakan akademik. Metode penelitian diawali dengan observasi dan wawancara kepada Koordinator KP untuk menyusun Product Backlog, dilanjutkan dengan perancangan sistem berupa UML, serta perancangan basis data. Proses pengembangan dilaksanakan melalui empat siklus Sprint dengan peran Scrum yang diadaptasi untuk pengembang tunggal, mencakup tahapan Sprint Planning, Sprint Execution, Sprint Review,*

dan Sprint Retrospective. Pengujian sistem dilakukan menggunakan Black Box Testing terhadap 19 skenario uji dan User Acceptance Test (UAT) kepada 27 responden. Hasil pengujian menunjukkan tingkat keberhasilan Black Box Testing mencapai 100% dan hasil UAT memperoleh persentase kelayakan sebesar 94,50% dengan kategori Sangat Layak, sehingga sistem terbukti mampu mengotomatisasi pendaftaran, distribusi dosen pembimbing, serta administrasi Kerja Praktik secara lebih efisien, transparan, dan adaptif.
Kata Kunci: Agile Scrum, Kerja Praktik, Round Robin, Simulasi API, Sistem Manajemen

1. Pendahuluan

Kerja Praktik (KP) memegang peran vital sebagai jembatan antara dunia akademik dan industri, bukan sekadar syarat kelulusan. Pada fase ini mahasiswa diuji kemampuannya dalam beradaptasi terhadap budaya profesional di dunia industri (Risal et al., 2022). Dalam pelaksanaannya di Program Studi Teknik Informatika ITATS, kompleksitas proses administratif masih menjadi tantangan tersendiri sehingga mahasiswa harus mengalokasikan waktu ekstra untuk mengurus dokumen, yang berpotensi menurunkan optimalisasi capaian pembelajaran.

Sistem pendaftaran digital yang telah dikembangkan sebelumnya menggunakan metode *Rapid Application Development* (RAD) baru mampu menangani fungsionalitas administrasi dasar, seperti manajemen login dan registrasi, pendaftaran kelompok, pengunduhan dokumen pendukung, pengisian logbook, penilaian, dan pengumpulan dokumen akhir. Namun demikian, sistem tersebut masih berdiri sendiri (belum terintegrasi dengan pusat informasi kampus SIM ITATS), sehingga mahasiswa dan dosen masih harus menggunakan akun terpisah untuk login, dan data anggota kelompok masih dimasukkan secara manual. Ketiadaan rekam jejak digital pada sistem bimbingan juga menyulitkan institusi dalam memantau riwayat progres akademik (Kurniasari et al., 2025).

Permasalahan lain yang teridentifikasi mencakup proses bimbingan yang masih bergantung pada komunikasi melalui WhatsApp dan logbook kertas, rekapitulasi dana pendaftaran secara manual yang rentan kesalahan hitung, ketiadaan sistem notifikasi otomatis, serta mekanisme rotasi dosen pembimbing yang belum terstruktur sebagai model distribusi komputasional yang terdokumentasi. Rotasi tersebut masih bergantung pada pengecekan histori penugasan terakhir secara manual sehingga rentan kehilangan jejak urutan dan sering kembali ke urutan awal saat berganti periode KP, serta belum memiliki penanganan dinamis untuk mengabaikan antrean dosen yang tiba-tiba berstatus tidak aktif.

Untuk mengatasi distribusi dosen pembimbing, penelitian ini menggunakan algoritma Round Robin, yaitu metode penjadwalan yang dirancang untuk mendistribusikan alokasi antrean dengan fokus pada keadilan (*fairness*) (Fadhilah & Siregar, 2025). Sementara itu, untuk memfasilitasi pengembangan fungsionalitas yang bersifat dinamis, penelitian ini menggunakan kerangka kerja Scrum, yang terbukti mampu memberikan tingkat fleksibilitas dan kecepatan adaptasi tinggi dalam tahap pembangunan sistem (Eryc, 2021) melalui pembagian proses pengembangan ke dalam siklus waktu pendek atau Sprint (Kustanto et al., 2021).

2. Tinjauan Pustaka

2.1. Sistem Informasi Manajemen Akademik dan Administrasi KP

Sistem Informasi Manajemen (SIM) merupakan sekumpulan subsistem yang saling bekerja sama untuk mengolah data mentah menjadi informasi akurat sebagai landasan pengambilan keputusan strategis (Rusdiana & Irfan, 2014). Implementasi SIM terbukti mampu meningkatkan kualitas manajemen data secara signifikan, mencakup aspek efisiensi data, aksesibilitas, pelaporan, dan komunikasi (Hamurdani & Kholik, 2023). Di tingkat perguruan tinggi, penerapan SIM dikenal sebagai Sistem Informasi Akademik (SIKAD), yang berperan mengintegrasikan seluruh manajemen data mahasiswa, dosen, dan kegiatan operasional kampus (Rofi'i, 2024).

Kerja Praktik merupakan instrumen utama perguruan tinggi untuk mengasah kesiapan kerja dan *soft-skill* mahasiswa (Prakasa & Chusairi, 2022), yang menuntut koordinasi sistematis mulai dari

pendaftaran, pengelolaan kontrak kerja sama, pemantauan aktivitas, hingga penilaian akhir (Direktorat Jenderal Pendidikan Tinggi, 2020). Sistem pencatatan dan pelaporan dana administrasi merupakan bagian dari sistem informasi akuntansi yang berfungsi mengumpulkan, mencatat, dan memproses data transaksi menjadi informasi manajerial (Romney et al., 2021), penerapan sistem digital pada aspek ini terbukti meningkatkan akurasi pencatatan, transparansi, dan efisiensi waktu (Adzkia & Anastasya, 2024). Manajemen jadwal dan rekam jejak bimbingan merupakan wujud fungsi pengendalian akademik untuk memantau interaksi dosen dan mahasiswa (Rusdiana & Irfan, 2014), yang dokumentasinya penting untuk mencegah kesalahpahaman dan untuk evaluasi (Romney et al., 2021).

2.2 Metode Pengembangan Sistem (Agile) dan Kerangka Kerja Scrum

Software Development Life Cycle (SDLC) merupakan serangkaian aktivitas analisis, perancangan, dan implementasi untuk membangun sistem (Nova et al., 2022). Agile Development hadir sebagai pendekatan iteratif yang menguraikan pengembangan sistem menjadi bagian-bagian kecil yang dikerjakan bertahap (Ramadhan et al., 2023), meliputi tahapan analisis kebutuhan, perancangan, pengembangan, pengujian, peluncuran, dan evaluasi (Fagarasan et al., 2021).

Scrum merupakan kerangka kerja manajemen untuk pengembangan produk kompleks secara adaptif, berlandaskan tiga pilar empirisme: transparansi, inspeksi, dan adaptasi (Rubin, 2012). Scrum menerapkan *Empirical Process Control* karena kebutuhan sistem dianggap terus berkembang sehingga perencanaan sempurna di awal proyek dinilai mustahil (Schwaber, 2004). Tiga peran dalam Scrum meliputi *Product Owner* (mengelola Product Backlog dan menentukan prioritas), *ScrumMaster* (fasilitator yang menjaga jalannya aturan Scrum), dan *Development Team* (bertanggung jawab mengubah Product Backlog menjadi inkremen fungsionalitas) (Rubin, 2012; Schwaber, 2004). Artefak Scrum terdiri atas *Product Backlog*, *Sprint Backlog*, dan *Increment* yang diukur melalui *Definition of Done* (Rubin, 2012; Schwaber, 2004). Aktivitas Scrum mencakup *Sprint*, *Sprint Planning*, *Daily Scrum*, *Sprint Review*, dan *Sprint Retrospective* (Rubin, 2012; Schwaber, 2004). Fleksibilitas Product Backlog terbukti dapat mengakomodasi perubahan, terutama regulasi di lingkungan akademik (Eryc, 2021).

2.3 Algoritma Round Robin

Round Robin merupakan salah satu algoritma penjadwalan tertua dan dinilai paling adil (*fairest*) dalam sistem komputasi, dengan konsep dasar pemberian interval waktu (*quantum*) kepada setiap proses secara bergiliran (Tanenbaum & Bos, 2015). Jika *quantum* habis namun proses belum selesai, sistem memindahkan proses tersebut ke posisi akhir antrian; sebaliknya jika proses selesai lebih awal, giliran dialihkan ke proses berikutnya (Tanenbaum & Bos, 2015). Secara prosedural, alur kerja Round Robin dimulai dari inisialisasi sistem dan penentuan nilai *quantum* (Widiarto et al., 2024). Dalam penelitian ini, satu *quantum* direpresentasikan secara konseptual sebagai alokasi satu kelompok mahasiswa untuk setiap dosen dalam satu putaran.

2.4 Application Programming Interface (API)

API merupakan lapisan antarmuka yang memungkinkan interaksi antarsistem secara terprogram, dengan Web API memanfaatkan protokol HTTP untuk mendistribusikan data melalui jaringan internet (Haro Peralta, 2023). *Representational State Transfer* (REST) adalah gaya arsitektur yang dirancang membangun sistem *highly scalable* dan *loosely coupled*, dengan dua prinsip dasar yaitu pemisahan *Client-Server* dan *Statelessness* (Haro Peralta, 2023). Metode HTTP dalam RESTful API memodelkan operasi CRUD (*Create, Read, Update, Delete*) melalui lima metode utama: GET, POST, PUT, PATCH, dan DELETE (Haro Peralta, 2023). *JavaScript Object Notation* (JSON) digunakan sebagai format representasi data berbasis pasangan kunci-nilai eksplisit mendefinisikan tipe data (Haro Peralta, 2023).

2.5 Pengujian Perangkat Lunak

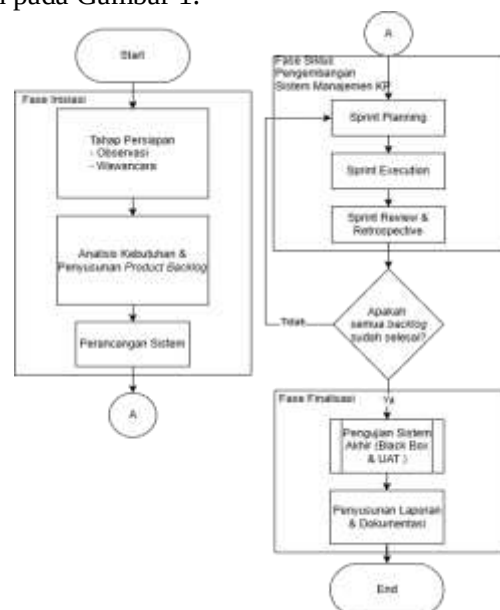
Pengujian perangkat lunak adalah aktivitas sistematis menjalankan program untuk menemukan kesalahan atau cacat (Chauhan, 2010). *Black Box Testing* berfokus pada persyaratan fungsional tanpa memperhatikan struktur internal program, dengan empat teknik utama: *Equivalence Partitioning*, *Boundary Value Analysis*, *Decision Table-Based Testing*, dan *State-Based Testing* (Chauhan, 2010). *User Acceptance Test* (UAT) merupakan implementasi praktis dari *Acceptance Testing* yang

memberikan bukti objektif bahwa perangkat lunak memenuhi kriteria penerimaan (Institute of Electrical and Electronics Engineers (IEEE), 2017), yang dapat dieksekusi menggunakan kuesioner berbasis Skala Likert (Firdaus et al., 2022).

3. Metode Penelitian

3.1 Alur Penelitian

Penelitian ini merupakan pengembangan lanjutan dari sistem pendaftaran KP sebelumnya yang dibangun menggunakan metode RAD, dengan fokus pengembangan pada penambahan fitur manajerial dan mekanisme distribusi dosen pembimbing otomatis. Alur penelitian diadaptasi menggunakan kerangka kerja Scrum yang bersifat iteratif dan inkremental, berlandaskan tiga pilar empirisme: transparansi, inspeksi, dan adaptasi (Schwaber, 2004). Secara keseluruhan, tahapan metodologi penelitian ini divisualisasikan pada Gambar 1.



Gambar 1. Alur Penelitian

Berdasarkan Gambar 1, tahapan penelitian terbagi menjadi tiga fase: (1) Fase Inisiasi (persiapan, observasi dan wawancara, analisis kebutuhan dan penyusunan Product Backlog, perancangan sistem); (2) Fase Siklus Pengembangan (empat siklus Sprint yang masing-masing melalui *Sprint Planning*, *Sprint Execution*, serta *Sprint Review & Retrospective*); dan (3) Fase Finalisasi (pengujian sistem menggunakan Black Box Testing dan UAT, dilanjutkan penyusunan laporan).

3.2 Tahap Persiapan

Observasi dan wawancara dilakukan kepada Koordinator Kerja Praktik ITATS untuk menggali kebutuhan spesifik. Hasil wawancara menemukan beberapa poin krusial: mekanisme bimbingan harus bersifat dua arah (mahasiswa mengajukan jadwal, dosen menyetujui/menolak) dengan batas satu kali bimbingan per minggu dan minimal 4 kali bimbingan; jika dosen berhalangan hadir setelah menyetujui, status diubah menjadi "Berhalangan Hadir" dan mahasiswa mengajukan ulang; rekapitulasi dana cukup berupa total uang masuk yang bertambah otomatis saat pendaftaran; laporan keuangan memuat kolom Mahasiswa, Dosen Pembimbing, dan Jenis Pembayaran; fitur jadwal seminar/sidang tidak perlu dibangun karena sudah ditangani website SEMTIK; serta dashboard perlu menampilkan timeline yang diunggah koordinator.

Selanjutnya dilakukan analisis pengembangan fungsionalitas sistem yang membandingkan sistem sebelumnya dengan sistem yang diusulkan pada delapan aspek: autentikasi pengguna (beralih ke integrasi Simulasi API SIM Akademik), alokasi pembimbing (beralih ke Round Robin berbasis beban

kerja), bimbingan KP (penambahan fitur reservasi), notifikasi sistem (otomatisasi), rekapitulasi keuangan (modul otomatis), perpanjangan KP (modul baru), audit log aktivitas (fitur baru), serta dashboard dan arsip terpusat (fitur baru).

3.3 Adaptasi Peran Scrum

Karena pengembangan dilakukan oleh satu orang (*single developer*), peran Scrum diadaptasi tanpa menghilangkan tanggung jawab masing-masing peran. Koordinator Kerja Praktik Prodi Teknik Informatika ITATS berperan sebagai *Product Owner* yang menentukan kebutuhan sistem, memvalidasi fitur, dan menetapkan prioritas Product Backlog. Peneliti berperan ganda sebagai *ScrumMaster* (memastikan tahapan Scrum berjalan disiplin) dan *Development Team* (merancang, mengimplementasikan, dan menguji sistem).

3.4 Perancangan Product Backlog dan Estimasi Sprint

Seluruh kebutuhan fungsional disusun dalam format *User Stories* dengan penentuan prioritas berdasarkan nilai bisnis dan ketergantungan antar-fitur, serta estimasi *effort* menggunakan deret Fibonacci (1, 2, 3, 5, 8, 13). Product Backlog terdiri atas 14 *user story* (US01–US14) sebagaimana dirangkum pada Tabel 2.

Tabel 2. Ringkasan Product Backlog

ID	Fitur	Prioritas	Effort
US01	Login integrasi Simulasi API SIM Akademik	1	13
US02	Pendaftaran kelompok & alokasi Round Robin	2	8
US03	Reservasi bimbingan	3	8
US04	Notifikasi otomatis	4	5
US05	Rekap dana KP otomatis	5	3
US06	Log aktivitas pengguna	6	8
US07	Upload dokumen KP & PPT	7	5
US08	Upload timeline	8	3
US09	Kelola data dosen	9	1
US10	List aplikasi KP	10	3
US11	Dashboard (timeline)	11	2
US12	Perpanjangan KP	12	2
US13	Upload surat balasan mitra	13	5
US14	Approve dosen pembimbing	14	1

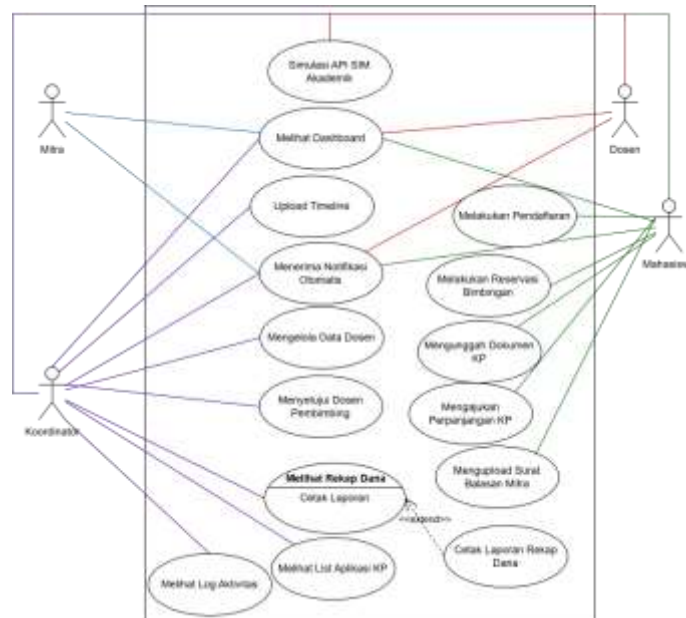
Estimasi awal pembagian sprint disusun dengan durasi masing-masing siklus selama satu minggu, dipantau menggunakan Jira, dengan keputusan akhir urutan pengerjaan tetap ditentukan bersama Koordinator KP pada *Sprint Planning* setiap iterasi.

3.5 Perancangan Sistem

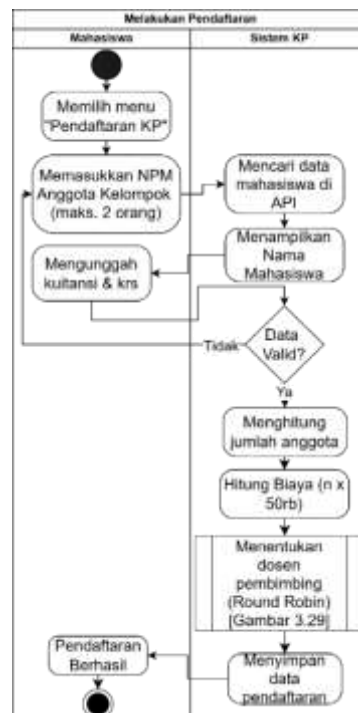
Perancangan sistem menerjemahkan kebutuhan pengguna ke dalam representasi UML sebagai panduan teknis implementasi (Pressman, 2010). Alur sistem terbagi menjadi tiga fase. Fase pertama adalah Pendaftaran dan Persetujuan Mitra, yaitu mahasiswa mendaftar kelompok, sistem menentukan dosen pembimbing melalui *Round Robin*, mahasiswa mengunduh surat survei, dan jika mitra menolak maka data instansi diperbarui. Fase kedua adalah Pelaksanaan dan Bimbingan, yaitu dosen mengunggah surat tugas, kemudian mahasiswa mengajukan bimbingan secara berkala hingga dosen menyatakan KP selesai. Fase ketiga adalah Finalisasi dan Pemberkasan, dosen memperbarui status menjadi "Selesai", penilaian, mahasiswa mengunggah dokumen, dan sistem mengarahkan pengguna ke SEMTIK. Tahapan perancangan ini dievaluasi melalui pemodelan UML berikut.

Use Case Diagram dirancang untuk memberikan gambaran menyeluruh mengenai fungsionalitas sistem dari perspektif pengguna. Sebagaimana ditunjukkan pada Gambar 2, pemodelan ini menempatkan empat aktor utama di luar batas sistem, yaitu Mahasiswa, Dosen, Koordinator, dan Mitra KP. Hak akses fungsionalitas dipetakan berdasarkan *Product Backlog*, dengan penambahan relasi *extend* pada fitur Cetak Laporan sebagai perluasan opsional dari fitur Melihat Rekap Dana. Secara keseluruhan, rancangan operasional sistem memuat 14 *Activity Diagram* dan 11 *Sequence Diagram*. Sebagai representasi fungsionalitas utama sistem, aliran aktivitas untuk skenario Pendaftaran Kelompok divisualisasikan pada Gambar 3. Proses dimulai saat mahasiswa menginputkan NPM, yang kemudian

dicari oleh sistem melalui integrasi API untuk memuat nama secara otomatis. Setelah kuitansi diunggah dan dievaluasi, sistem mengkalkulasi akumulasi biaya pendaftaran ($n \times 50.000$) dan mengeksekusi algoritma Round Robin untuk menentukan dosen pembimbing secara otomatis sebelum data disimpan ke basis data.



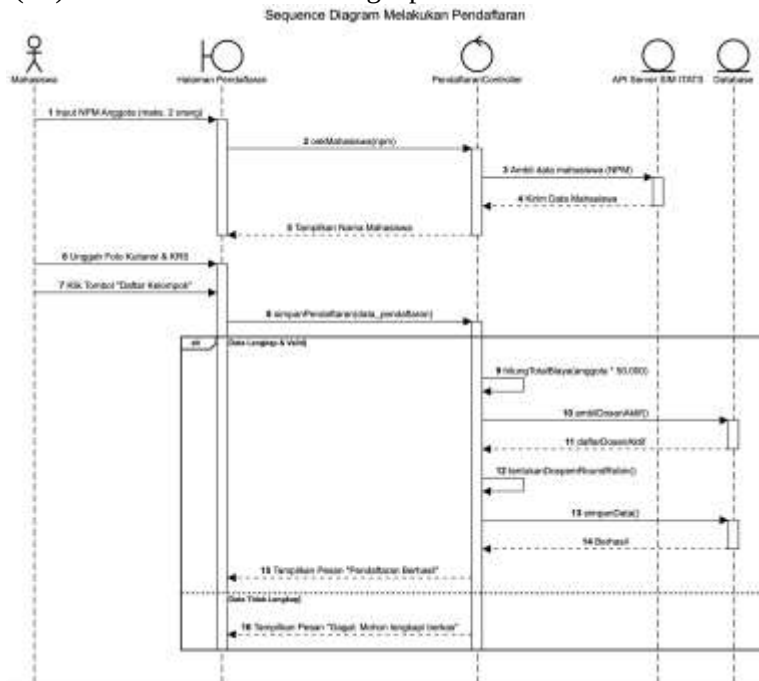
Gambar 2. Use Case Diagram



Gambar 3. Activity Diagram Pendaftaran Kelompok

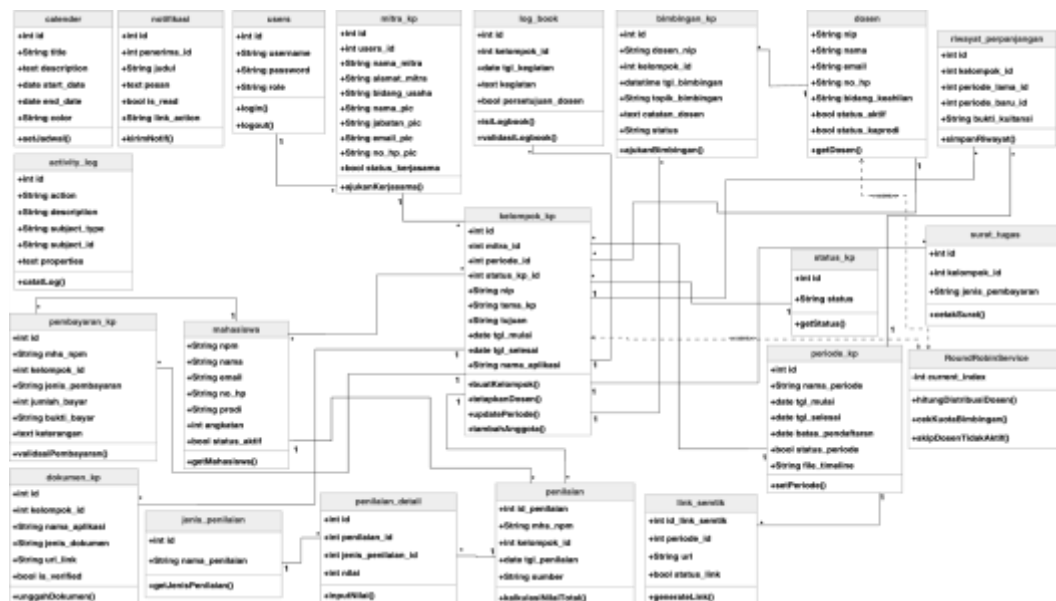
Untuk mendetailkan urutan kronologi waktu dan pengiriman pesan antar-objek pada skenario pendaftaran tersebut, *Sequence Diagram* diilustrasikan pada Gambar 4. Perancangan ini menggunakan

empat komponen utama yaitu *Actor*, *Boundary*, *Control*, dan *Entity*. Proses teknis dimulai dengan verifikasi NPM ke Simulasi API, dilanjutkan dengan eksekusi fungsi `hitungTotalBiaya()`. Sistem kemudian memuat daftar dosen berstatus aktif menggunakan fungsi `ambilDosenAktif()`, dan menetapkan pembimbing melalui fungsi `tentukanDospemRoundRobin()`. Alur diakhiri dengan percabangan bersyarat (*alt*) untuk memvalidasi kelengkapan berkas sebelum data direkam.



Gambar 4. Sequence Diagram Pendaftaran Kelompok

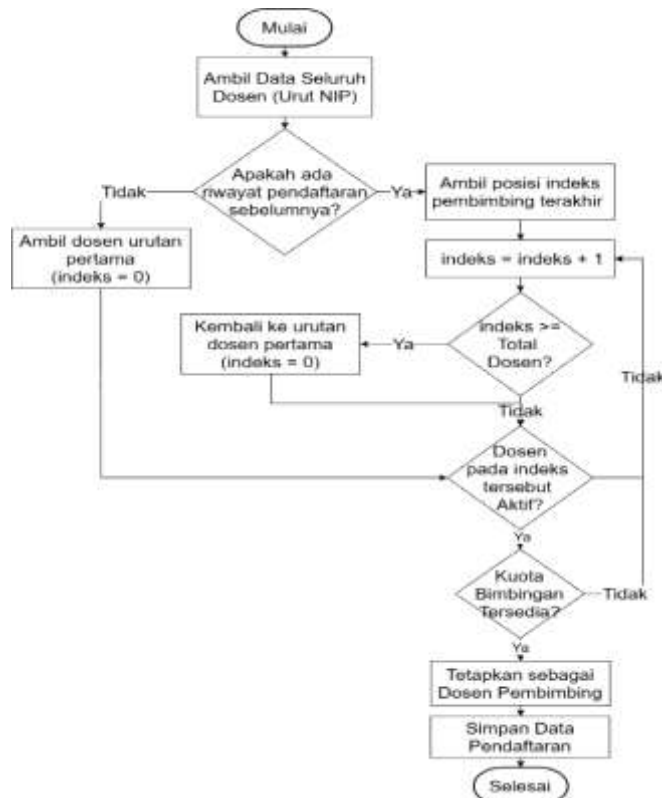
Secara arsitektural, rancangan struktur dan relasi antar-objek dimodelkan melalui *Class Diagram* pada Gambar 5. Pemodelan ini menempatkan `kelompok_kp` sebagai poros utama sistem dan `RoundRobinService` sebagai kelas *service* logika alokasi dosen dengan relasi *dependency*. Entitas pendukung lainnya meliputi `riwayat_perpanjangan`, `bimbingan_kp`, `dokumen_kp`, dan `pembayaran_kp` yang seluruh perubahannya terekam secara terpusat oleh `activity_log`.



Gambar 5. Class Diagram

3.6 Perancangan Algoritma Round Robin

Penetapan dosen pembimbing dirancang menggunakan antrean melingkar berbasis indeks dosen, berbeda dengan penjadwalan CPU yang menggunakan parameter waktu. Cara kerja algoritma ini dijelaskan melalui diagram alur pada Gambar 6.



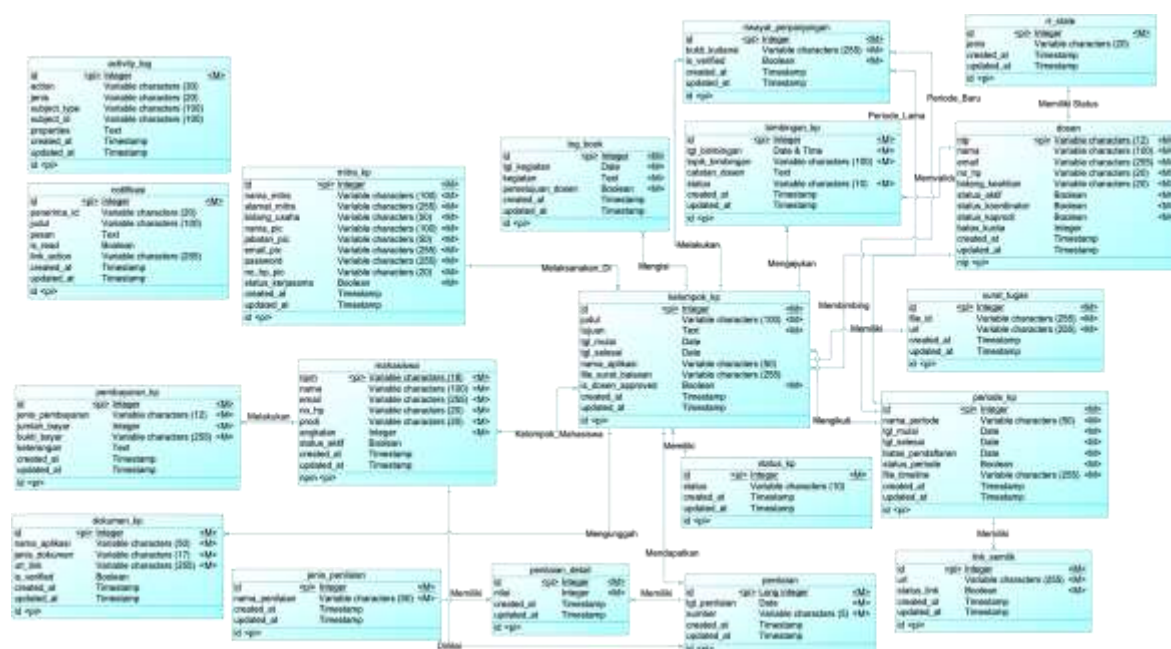
Gambar 6. Flowchart Algoritma Round Robin

Proses diawali dengan memuat seluruh data dosen terurut berdasarkan NIP. Jika belum ada riwayat pendaftaran, sistem mengambil dosen pada indeks pertama; jika sudah ada, sistem mengambil posisi indeks dosen pembimbing terakhir dan melanjutkan iterasi ke urutan berikutnya, dengan reset otomatis ke urutan pertama apabila iterasi melampaui batas akhir daftar dosen (*wrap around*). Setelah giliran ditentukan, sistem melakukan dua lapis validasi: status keaktifan dosen dan ketersediaan kuota bimbingan. Apabila tidak memenuhi salah satu syarat, sistem melanjutkan iterasi ke urutan berikutnya. Dosen yang terverifikasi aktif dan memiliki kuota kemudian ditetapkan sebagai pembimbing dan data pendaftaran disimpan ke basis data.

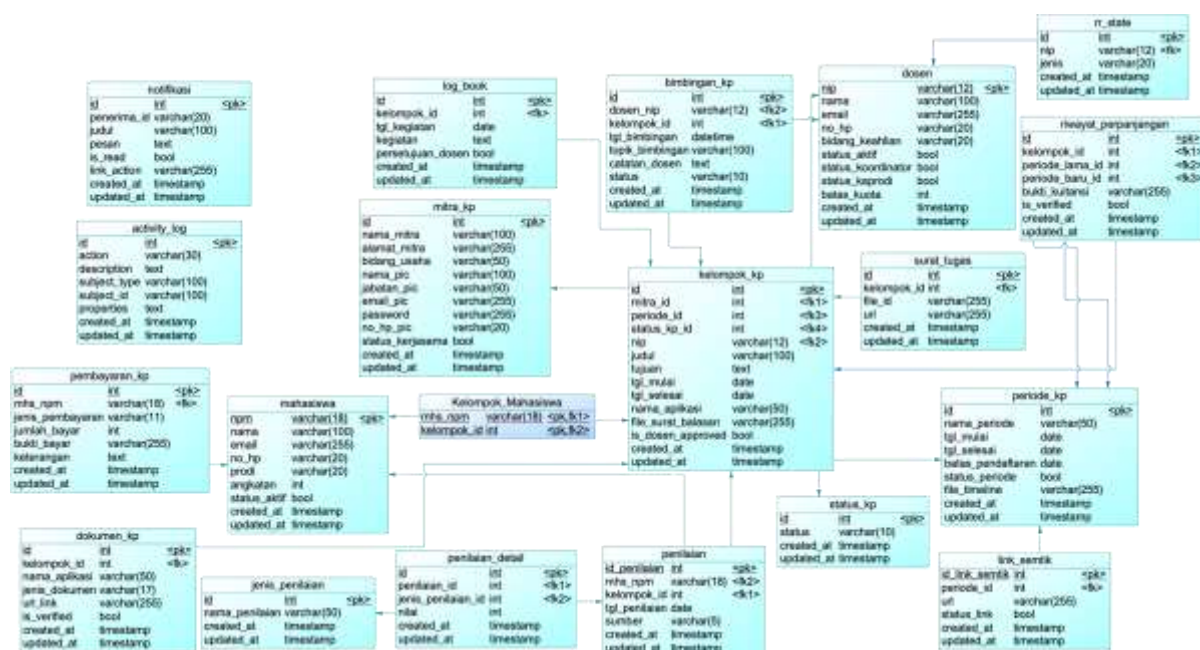
3.7 Perancangan Basis Data

Conceptual Data Model (CDM) menempatkan entitas kelompok_kp sebagai poros utama yang menghubungkan seluruh proses bisnis. Restrukturisasi dilakukan melalui penggabungan berkas administrasi kelompok ke entitas dokumen_kp, penambahan entitas pembayaran_kp dan bimbingan_kp, penambahan atribut file_timeline pada entitas periode_kp, integrasi activity_log sebagai subsistem audit, serta penghilangan atribut kata sandi lokal pada entitas mahasiswa dan dosen karena autentikasi dialihkan sepenuhnya ke integrasi API. Pemodelan konseptual ini divisualisasikan secara lengkap pada Gambar 7.

Physical Data Model (PDM) menerjemahkan rancangan konseptual menjadi tabel dengan relasi *Foreign Key*, termasuk relasi *many-to-many* antara mahasiswa dan kelompok_kp melalui tabel pivot kelompok_mahasiswa. Struktur teknis relasi basis data ini direpresentasikan pada Gambar 8.



Gambar 7. Conceptual Data Model (CDM)



Gambar 8. *Physical Data Model (PDM)*

3.8 Prosedur Pengembangan

Sprint Planning dilakukan pada awal setiap iterasi melalui diskusi peneliti dan Koordinator KP untuk menyeleksi Product Backlog prioritas tertinggi menjadi Sprint Backlog. *Sprint Execution* dan *Daily Scrum* diadaptasi menjadi evaluasi harian mandiri untuk memantau progres dan kendala teknis. *Sprint Review* dilaksanakan pada akhir siklus untuk mendemonstrasikan modul kepada Koordinator KP dan mengolah umpan balik menjadi kebutuhan baru pada Product Backlog. *Sprint Retrospective* digunakan untuk mengevaluasi efektivitas proses kerja dan merumuskan perbaikan pada siklus berikutnya. Evaluasi penerapan Scrum diukur melalui tiga indikator: adaptabilitas terhadap perubahan prioritas backlog, persentase penyelesaian target modul per Sprint, dan keberhasilan penerapan umpan balik Sprint Review.

3.9 Rencana Pengujian Sistem

Rencana pengujian mencakup tahap persiapan skenario uji, validasi fungsional (Black Box Testing), validasi penerimaan (UAT), dan analisis hasil. Pengujian akurasi algoritma Round Robin dirancang melalui 7 skenario uji (UJI RR01–UJI RR07) yang mengukur penunjukan dosen pertama kali, mekanisme *skip* dosen tidak aktif, *rotation wrap around*, *skip* kuota penuh dan nonaktif, penanganan batas maksimal kapasitas kuota, pemulihan kuota setelah penghapusan kelompok, dan penanganan status dosen terakhir yang dinonaktifkan, dengan rumus:

$$\text{Akurasi Algoritma} = (\text{Jumlah Skenario Sesuai} / \text{Total Skenario Pengujian}) \times 100\%$$

Kriteria keberhasilan pengujian akurasi ditetapkan sebesar 100%. Black Box Testing dirancang melalui 19 skenario uji (TC01–TC19) yang mencakup pengujian kondisi normal, gagal, dan batas nilai untuk seluruh fitur utama pada Product Backlog. UAT dilaksanakan menggunakan kuesioner Skala Likert 5 poin (Sangat Setuju=5, Setuju=4, Cukup=3, Tidak Setuju=2, Sangat Tidak Setuju=1) dengan 7 pernyataan meliputi aspek kemudahan penggunaan, efisiensi proses, transparansi data, koordinasi sistem, keandalan teknis, kepuasan pengguna, dan penerimaan sistem, dengan rumus persentase skor (Sugiyono, 2013):

$$P = (\sum x / \sum xi) \times 100\%$$

Kriteria interpretasi kelayakan UAT disajikan pada Tabel 3.

Sistem dinyatakan lulus pengujian UAT apabila mencapai kategori Layak atau Sangat Layak ($\geq 61\%$).

Tabel 3. Kriteria Interpretasi Kelayakan UAT

Persentase Pencapaian	Kriteria Kelayakan
81% – 100%	Sangat Layak
61% – 80%	Layak
41% – 60%	Cukup
21% – 40%	Kurang Layak
0% – 20%	Tidak Layak

Sumber: (Sugiyono, 2013), dimodifikasi.

3.10 Hipotesis Penelitian

Penelitian ini mengajukan tiga hipotesis. Pertama, hipotesis adaptabilitas Scrum, yaitu implementasi Agile Scrum mampu mengakomodasi perubahan kebutuhan fungsional secara adaptif tanpa mengganggu *Definition of Done* pada setiap Sprint. Kedua, hipotesis validitas integrasi Simulasi API, yaitu integrasi Simulasi API SIM Akademik mampu mengotomatisasi pengisian data pendaftaran secara *real-time* sekaligus mencegah anomali atau redundansi data. Ketiga, hipotesis konsistensi algoritma Round Robin, yaitu algoritma Modifikasi Round Robin mampu mendistribusikan beban kuota dosen pembimbing secara merata dan konsisten meskipun terdapat interupsi perubahan dosen secara manual oleh Koordinator KP, serta adaptif dalam mengecualikan dosen berstatus tidak aktif.

4. Hasil dan Pembahasan

4.1 Penyesuaian Perencanaan Sprint

Sistem direalisasikan menggunakan *framework* Laravel dan basis data MySQL. Berdasarkan evaluasi bersama Koordinator KP selaku *Product Owner*, urutan pengerjaan Product Backlog disesuaikan agar transisi arsitektur dari input manual ke sistem berbasis API diselesaikan lebih awal dan terpusat. Penyesuaian ini menggeser fitur Melakukan Pendaftaran (US02) ke Sprint 1 bersama Simulasi API (US01), sedangkan fitur pengelolaan data master bergeser ke Sprint 2, sebagaimana dirangkum pada Tabel 4.

Tabel 4. Penyesuaian Product Backlog per Sprint

Sprint	Periode	Backlog Item	Effort
1	28 April – 4 Mei	US01 (13), US02 (8)	21
2	5 Mei – 11 Mei	US08 (3), US09 (1), US11 (2), US13 (5), US14 (1)	12

3	12 Mei – 18 Mei	US03 (8), US04 (5), US05 (3), US12 (2)	18
4	19 Mei – 26 Mei	US06 (8), US07 (5), US10 (3)	16

4.2 Implementasi Sistem Berdasarkan Sprint

Setiap Sprint dievaluasi menggunakan *Definition of Done* (DoD): fungsionalitas berjalan benar melalui pengujian, kode bebas dari *error*, dan hasil implementasi memenuhi ekspektasi *User Story*.

Sprint 1 menyelesaikan US01 dan US02 melalui pembangunan *controller* simulasi API berbasis kontainer Docker yang di-*deploy* ke platform *cloud* Render, menerapkan metode *Multi-layer API Security* melalui *Client Credentials* dan *Stateless Bearer Token*. Fitur Pendaftaran Kelompok berhasil mengeksekusi pencarian dan penarikan data mahasiswa secara otomatis. Seluruh backlog Sprint 1 dinyatakan memenuhi DoD setelah pengujian fungsional lintas server tanpa *error*.

Sprint 2 menyelesaikan US08, US09, US11, US13, dan US14. Kendala ditemukan pada fitur Approve Dosen Pembimbing (US14): perubahan dosen pembimbing secara manual oleh Koordinator menyebabkan urutan distribusi otomatis pada antrean berikutnya tidak sesuai mekanisme rotasi. Kendala ini diselesaikan dengan pemisahan kolom nip (menyimpan dosen akhir hasil perubahan manual) dan nip_rr (mencatat dosen pilihan murni hasil perhitungan algoritma) agar perhitungan antrean rotasi tetap konsisten.

Sprint 3 menyelesaikan US03, US04, US05, dan US12. Ditemukan celah logika di mana kuota bimbingan dosen tidak otomatis pulih (*reset*) ketika kelompok binaannya dinyatakan selesai, serta potensi mahasiswa menyelesaikan administrasi KP tanpa memenuhi standar minimal bimbingan. Kendala ini diselesaikan melalui pembaruan logika kuota otomatis berdasarkan status akhir kelompok dan penambahan validasi minimal pengisian logbook.

Sprint 4 menyelesaikan US06, US07, dan US10. Dilakukan refaktorisasi fungsionalitas List Aplikasi KP (US10) dan Upload Dokumen (US07) menjadi satu modul Dokumen Kelompok, karena fungsi awal US10 dinilai terlalu sempit jika hanya menampilkan tautan aplikasi. Dengan selesainya Sprint 4, seluruh Product Backlog berhasil direalisasikan menjadi sistem informasi yang utuh.

4.3 Hasil Uji Akurasi Algoritma Round Robin

Pengujian akurasi algoritma Round Robin dilaksanakan melalui 7 skenario uji, dengan hasil disajikan pada Tabel 5.

Tabel 5. Hasil Pengujian Akurasi Algoritma Round Robin

Kode Uji	Skenario Pengujian	Status
UJI RR01	Penunjukan dosen pertama kali (tanpa riwayat)	Sesuai
UJI RR02	Mekanisme <i>skip</i> dosen tidak aktif	Sesuai
UJI RR03	Mekanisme <i>rotation wrap around</i>	Sesuai
UJI RR04	Mekanisme <i>skip</i> kuota penuh & nonaktif	Sesuai
UJI RR05	Penanganan batas maksimal kapasitas kuota	Sesuai
UJI RR06	Pemulihan kuota setelah penghapusan kelompok	Sesuai
UJI RR07	Penanganan status dosen terakhir dinonaktifkan	Sesuai

Tingkat akurasi algoritma dihitung menggunakan Rumus:

$$\text{Akurasi Algoritma} = (7/7) \times 100\% = 100\%$$

Seluruh skenario membuktikan algoritma secara konsisten mendistribusikan beban bimbingan secara berurutan, melompati dosen yang tidak memenuhi syarat kuota maupun status keaktifan, serta merespons perubahan status data secara *real-time* tanpa menimbulkan *deadlock* atau kesalahan alokasi.

4.4 Hasil Black Box Testing

Pengujian Black Box dilakukan terhadap 19 skenario uji (TC-01 hingga TC-19) yang mencakup fitur login, pendaftaran kelompok, reservasi bimbingan, notifikasi, rekap dana, upload dokumen, upload surat balasan mitra, log aktivitas, dashboard timeline, perpanjangan KP, persetujuan bimbingan, data dosen, dokumen kelompok, ekspor rekap dana, manajemen data dosen, dan approve dosen pembimbing. Seluruh 19 skenario dinyatakan Sesuai dengan hasil yang diharapkan. Persentase keberhasilan dihitung menggunakan Rumus:

$$\text{Persentase Keberhasilan} = (19/19) \times 100\% = 100\%$$

Hasil ini membuktikan seluruh respons aktual sistem telah sesuai dengan hasil yang diharapkan, termasuk fitur alokasi dosen Round Robin, reservasi bimbingan, simulasi API, pencatatan dana, dan ekspor laporan, serta mekanisme *error handling* yang menangani input kosong maupun file melebihi batas kapasitas.

4.5 Hasil User Acceptance Test (UAT)

UAT dilaksanakan oleh 27 responden (26 Mahasiswa dan 1 Koordinator KP) menggunakan kuesioner 7 butir pernyataan berskala Likert. Hasil rekapitulasi skor UAT disajikan pada Tabel 6.

Tabel 6. Hasil Pengujian UAT

No	Aspek Penilaian	Persentase (%)	Kriteria
1	Kemudahan Penggunaan	94,81	Sangat Layak
2	Efisiensi Proses	92,59	Sangat Layak
3	Transparansi Data	92,59	Sangat Layak
4	Koordinasi Sistem	94,81	Sangat Layak
5	Keandalan Teknis	94,81	Sangat Layak
6	Kepuasan Pengguna	94,81	Sangat Layak
7	Penerimaan Sistem	97,04	Sangat Layak
Rata-Rata		94,50	Sangat Layak

Skor tertinggi dicapai indikator Penerimaan Sistem (97,04%), menegaskan kesiapan sistem untuk diimplementasikan di lingkungan program studi. Nilai rata-rata akumulatif sebesar 94,50% menempatkan sistem pada kriteria Sangat Layak menurut Tabel 3.

4.6 Pembuktian Hipotesis

Hipotesis 1 (Adaptabilitas Scrum) terbukti melalui dua fase krusial: penataan ulang prioritas backlog pada *Sprint Planning* (Sprint 1 dan 2), di mana fitur Pendaftaran (US02) dipindah ke Sprint 1 untuk diintegrasikan langsung dengan Simulasi API (US01); serta refaktorisasi fungsionalitas List Aplikasi KP (US10) menjadi modul Dokumen Kelompok pada Sprint Execution Sprint 4. Kedua adaptasi tersebut tidak mengganggu pencapaian *Definition of Done* pada masing-masing Sprint.

Hipotesis 2 (Validitas Integrasi Simulasi API) terbukti melalui keberhasilan sistem mengamankan jalur komunikasi via *Client Credentials*, menyimpan *Access Token* pada *Laravel Session*, dan menggunakan sesi tersebut sebagai *Bearer Token* untuk memicu *auto-fill* data mahasiswa (TC-03). Fungsi *updateOrCreate()* pada basis data lokal terbukti mencegah duplikasi atau redundansi akun mahasiswa.

Hipotesis 3 (Konsistensi Algoritma Round Robin) terbukti melalui hasil pengujian TC-03 (alokasi otomatis merata) dan TC-19 (intervensi manual Koordinator tidak merusak urutan rotasi berikutnya, karena pemisahan kolom nip dan nip_rr), serta TC-18 yang membuktikan algoritma adaptif mengecualikan dosen berstatus "Tidak Aktif" tanpa mengganggu indeks rotasi dosen lainnya.

5. Simpulan

Berdasarkan hasil rekayasa perangkat lunak, pelaksanaan siklus Sprint, dan analisis pengujian sistem, dapat disimpulkan: (1) penerapan metodologi Agile Scrum melalui empat siklus Sprint terbukti adaptif dan fleksibel terhadap perubahan kebutuhan fungsional maupun kebijakan akademik, ditunjukkan melalui penataan ulang prioritas backlog pada *Sprint Planning* serta penyesuaian modul Dokumen Kelompok pada *Sprint Execution*; (2) algoritma modifikasi Round Robin menghasilkan tingkat akurasi 100% dari 7 skenario pengujian, terbukti adil dan konsisten dalam mendistribusikan beban bimbingan, melompati dosen nonaktif maupun berkuota penuh, merespons pemulihan kuota, serta menangani intervensi koordinator secara presisi melalui skema pemisahan atribut nip dan nip_rr; (3) pengujian Black Box Testing mencatatkan tingkat keberhasilan 100% dari 19 skenario, membuktikan seluruh fungsi utama berjalan valid tanpa cacat fungsional dan didukung mekanisme *error handling* yang andal; (4) pengujian UAT terhadap 27 responden menghasilkan nilai rata-rata akumulatif 94,50% dengan kategori Sangat Layak, dengan skor tertinggi pada indikator Penerimaan Sistem (97,04%); dan

(5) sistem berhasil menerapkan arsitektur integrasi data bersama REST API SIM Akademik menggunakan skema *Client Credentials* dan *Bearer Token* berbasis *Laravel Session* yang mampu mengotomatisasi sinkronisasi profil mahasiswa secara *real-time* sekaligus meminimalkan risiko duplikasi data.

Beberapa saran untuk pengembangan lanjutan meliputi: pengembangan platform *mobile* (Android/iOS) untuk meningkatkan fleksibilitas akses; integrasi terpusat dengan sistem eksternal seperti SEMTIK agar siklus penjadwalan seminar, penentuan dosen penguji, dan administrasi sidang terotomatisasi secara terpadu; serta penguatan keamanan dan infrastruktur melalui analisis mendalam terkait keamanan jaringan, audit penetrasi, dan pengujian keandalan infrastruktur server produksi.

Referensi

- Adzkia, R., & Anastasya, F. (2024). MANAJEMEN KEUANGAN SEKOLAH: TRANSPARANSI DAN AKUNTABILITAS DALAM PENGELOLAAN DANA. *Al-Gafari: Manajemen Dan Pendidikan*, 2(3), 278–289. <https://jurnal.zarilgapari.org/index.php/gafari/article/view/140>
- Chauhan, Naresh. (2010). *Software testing : principles and practices*. Oxford University Press.
- Direktorat Jenderal Pendidikan Tinggi. (2020). *Buku Panduan Merdeka Belajar-Kampus Merdeka* (1st ed.). <https://repositori.kemendikdasmen.go.id/18005/>
- Eryc, S. M. (2021). Perancangan dan Implementasi Aplikasi Sistem Pendaftaran Sidang KP, Skripsi dan Tesis Online dengan Metode Scrum. *Journal of Information System and Technology*, 02(02), 49–55. <https://journal.uib.ac.id/index.php/joint/article/view/6086/2021>
- Fadhilah, I., & Siregar, H. (2025). Systematic Literature Review : Perbandingan Algoritma Round Robin dan Shortest Job First dalam Penjadwalan CPU. *BIOS : Jurnal Teknologi Informasi Dan Rekayasa Komputer*, 6(2), 74–83. <https://doi.org/10.37148/bios.v6i2.168>
- Fagarasan, C., Popa, O., Pisla, A., & Cristea, C. (2021). Agile, waterfall and iterative approach in information technology projects. *IOP Conference Series: Materials Science and Engineering*, 1169(1), 012025. <https://doi.org/10.1088/1757-899x/1169/1/012025>
- Firdaus, R., Hikmawati, N. K., Durachman, Y., Nanang, H., Khairani, D., & Hazimi, M. S. (2022, December). Usability Testing Analysis of a Company Website in Indonesia. *2022 Seventh International Conference on Informatics and Computing (ICIC)*. <https://doi.org/10.1109/ICIC56845.2022.10006910>
- Hamurdani, & Kholik, A. (2023). SISTEM INFORMASI MANAJEMEN KESISWAAN: SYSTEMATIC LITERATURE REVIEW. *AL - KAFF: Jurnal Sosial Humaniora*, 1(4). <https://ojs.unida.info/index.php/al-kaff/article/view/9171/4010>
- Haro Peralta, J. (2023). *Microservice APIs*. Manning Publications Co.
- Institute of Electrical and Electronics Engineers (IEEE). (2017). *IEEE Standard for System, Software, and Hardware Verification and Validation*. IEEE. <https://doi.org/10.1109/IEEESTD.2017.8055462>
- Kurniasari, N., Setiaji, P., & Latifah, N. (2025). Portal Monitoring Praktek Kerja Lapangan (PKL) Berbasis Web Pada Program Studi Sistem Informasi Universitas Muria Kudus. *JurnalUnitek*, 18(1). <https://doi.org/10.52072/unitek.v18i1.1239>
- Kustanto, P., Pamungkas, R. W. P., & Fathurrozi, A. (2021). Pembangunan Aplikasi E-Magang Perguruan Tinggi dengan Memanfaatkan SDLC SCRUM pada Agile Project Management. *Journal of Information and Information Security (JIFORTY)*, 2(12). <https://doi.org/10.31599/8gnssc35>
- Nova, S. H., Widodo, A. P., & Warsito, B. (2022). Analisis Metode Agile pada Pengembangan Sistem Informasi Berbasis Website: Systematic Literature Review. *Techno.COM*, 21(1), 139–148. <https://doi.org/10.33633/tc.v21i1.5659>
- Prakasa, P. R. S., & Chusairi, A. (2022). Sebuah Tinjauan Literatur: Peran Perguruan Tinggi Untuk Meningkatkan Kesiapan Kerja Para Lulusan. *PROCEEDING SERIES OF PSYCHOLOGY*, 1(1), 159–169. <https://psikologi.unair.ac.id/proceeding-series-of-psychology/index.php/psp/id/article/view/25>
- Pressman, R. S. (2010). *Software Engineering: A Practitioner's Approach* (7th ed.). www.mhhe.com/pressman.

- Ramadhan, J. A., Haniva, D. T., & Suharso, A. (2023). Systematic Literature Review Penggunaan Metodologi Pengembangan Sistem Informasi Waterfall, Agile, dan Hybrid. *Jurnal Ilmiah Elektronik & Engineering Technology*, 7(1). <https://doi.org/10.26740/jieet.v7n1.p36-42>
- Risal, M. F. N., Ambarsari, N., & Sutoyo, E. (2022). PEMBUATAN SISTEM INFORMASI ADMINISTRASI KERJA PRAKTEK DAN PENGABDIAN MASYARAKAT BERBASIS WEBSITE. *JURNAL SYNTAX FUSION*, 2. <https://doi.org/10.54543/fusion.v2i01.133>
- Rofi'i, M. (2024). Analisis Manfaat dan Tantangan Sistem Informasi Akademik dalam Manajemen Perguruan Tinggi: Pendekatan Systematic Literature Review. *IMEJ: Islamic Management and Education Journal*, 1(1). <https://doi.org/10.62730/imej.v01i01.135>
- Romney, M. B., Steinbart, P. J., Summers, S. L., & Wood, D. A. (2021). *Accounting Information Systems* (15th ed.). Pearson Education Limited. https://api.pageplace.de/preview/DT0400.9781292353289_A41441360/preview-9781292353289_A41441360.pdf
- Rubin, K. S. (2012). *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. <https://ptgmedia.pearsoncmg.com/images/9780137043293/samplepages/0137043295.pdf>
- Rusdiana, H. A. , & Irfan, M. (2014). *Sistem Informasi Manajemen* (1st ed.). CV Pustaka Setia. <https://umnaw.ac.id/wp-content/uploads/2021/08/Buku-Sistem-Informasi-Manajemen.pdf>
- Schwaber, K. (2004). *Agile Project Management with Scrum*. Microsoft Press.
- Sugiyono. (2013). *Metode Penelitian Kuantitatif, Kualitatif, dan R&D*. Alfabeta.
- Tanenbaum, A. S., & Bos, H. (2015). *MODERN OPERATING SYSTEMS* (4th ed.). Pearson Prentice Hall.
- Widiarto, W., Maheswari, D., Sari, D. P., & Arianto, K. J. (2024). Implementasi Algoritma Round Robin dan Priority Pada Sistem Antrian Rumah Sakit. *Jurnal FASILKOM (Teknologi InFormASi Dan ILmu KOMputer)*. <https://doi.org/10.37859/jf.v14i2.7334>