

BAB III

METODOLOGI PENELITIAN

3.1 Metode Penelitian

Penelitian ini menggunakan metode Design Science Research (DSR) yang berorientasi pada penciptaan solusi nyata berbasis sistem informasi. Pendekatan DSR menekankan pada proses perancangan dan evaluasi artefak (dalam hal ini sistem pengelolaan DUPAK berbasis web) sebagai jawaban atas permasalahan administratif dalam pengajuan dan penilaian angka kredit dosen. Metode ini sesuai dengan tujuan penelitian untuk menghasilkan sistem yang tidak hanya berfungsi secara teknis, tetapi juga memiliki kontribusi praktis bagi efisiensi pengelolaan data dosen (Sulistiyanto et al., 2025).

Secara konseptual, DSR merupakan metode yang digunakan untuk menciptakan, menguji, dan mengevaluasi solusi teknologi informasi terhadap masalah organisasi yang nyata. Menurut (Tuunanen et al., 2024), pendekatan ini menggabungkan proses iteratif antara teori dan praktik agar solusi yang dihasilkan tidak hanya inovatif, tetapi juga valid secara ilmiah. DSR menjadi metode yang ideal ketika penelitian bertujuan mengembangkan artefak seperti perangkat lunak, model sistem, atau kerangka kerja manajerial.

Metode ini dipilih karena bersifat konstruktif dan iteratif, memungkinkan peneliti untuk menghasilkan artefak yang dapat diuji secara langsung serta disempurnakan berdasarkan hasil evaluasi pengguna. Pendekatan DSR juga dianggap relevan karena mengintegrasikan unsur rekayasa perangkat lunak dengan pendekatan ilmiah yang berfokus pada pemecahan masalah nyata di lingkungan organisasi pendidikan tinggi (Gledson et al., 2024).

3.2 Tahapan Penelitian DSR

DSR dilaksanakan melalui beberapa tahapan sistematis, yaitu:

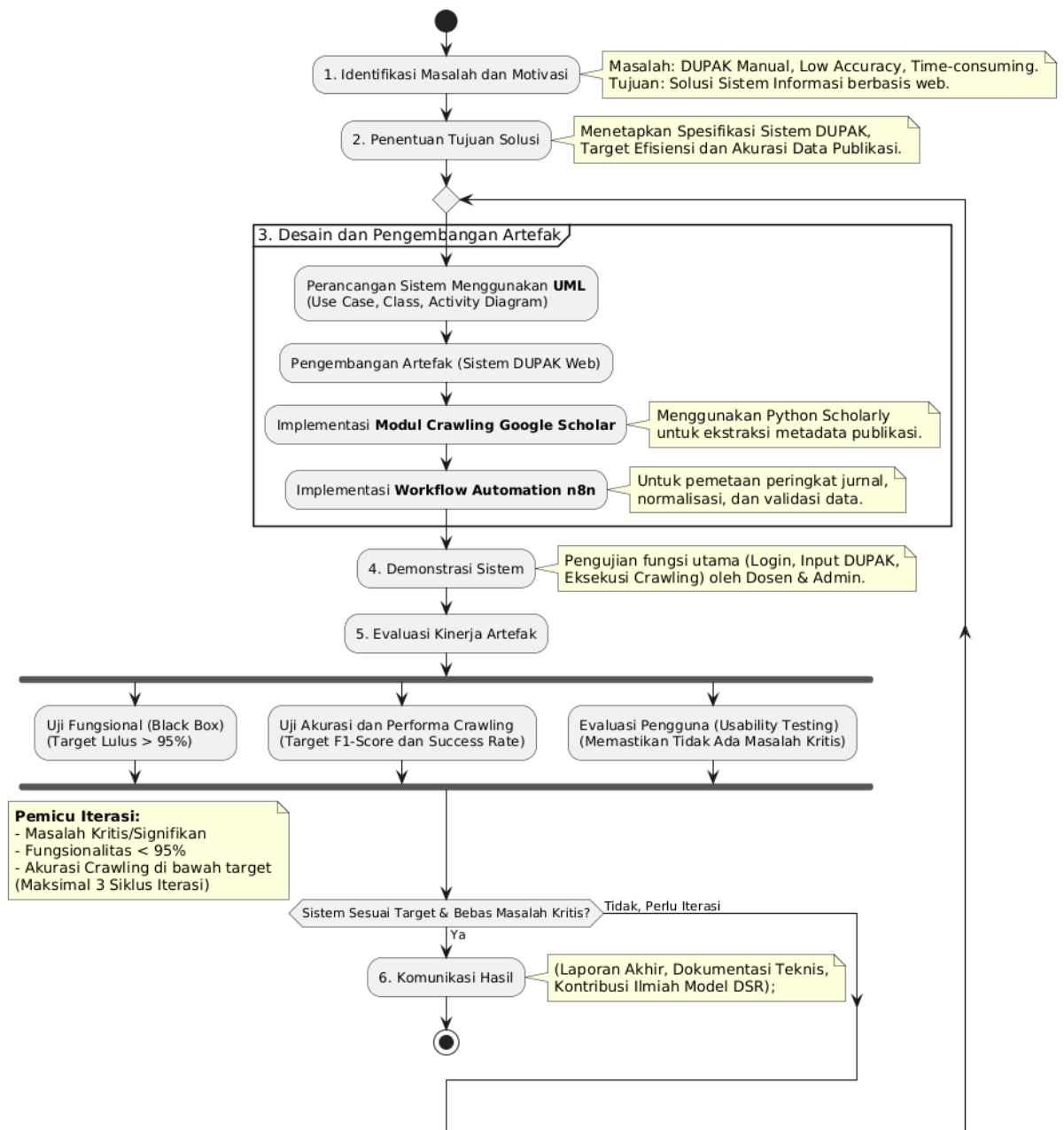
3.2.1 Identifikasi masalah dan motivasi

Tahapan identifikasi masalah dan motivasi dilakukan dengan mengamati proses pengajuan DUPAK di Fakultas Teknik Universitas Muhammadiyah Surabaya yang saat ini masih dilakukan secara manual, di mana dosen harus

mengumpulkan berkas fisik atau input data secara terpisah sehingga rawan terjadi ketidaksesuaian format, duplikasi data, serta keterlambatan dalam proses verifikasi.

Data pendukung seperti publikasi ilmiah juga belum terpusat, sehingga proses rekapitulasi membutuhkan waktu lebih lama dan meningkatkan beban administratif dosen maupun pihak validator. Kondisi ini menunjukkan kebutuhan akan sistem berbasis web yang mampu mengelola data DUPAK secara terstruktur, memberikan akses monitoring progres, serta mendukung otomatisasi pengambilan data publikasi melalui teknik web crawling untuk mengurangi input manual. Berdasarkan permasalahan tersebut, terdapat kesenjangan penelitian pada belum tersedianya sistem pengelolaan DUPAK yang terintegrasi, otomatis, dan efisien, sehingga mendorong pengembangan solusi digital sebagai artefak penelitian menggunakan pendekatan Design Science Research.





Gambar 1 Alur Metodologi Penelitian

3.2.2 Penentuan tujuan Solusi

Tahap penentuan tujuan solusi berfokus pada perumusan target yang harus dicapai oleh artefak sistem yang dikembangkan. Tujuan utama penelitian ini adalah membangun sistem berbasis web yang mampu mendukung proses pengelolaan DUPAK secara terpusat, menyediakan dashboard monitoring progres bagi dosen dan validator, serta memfasilitasi input dan verifikasi komponen angka kredit

secara lebih efisien. Sistem juga dirancang untuk mengotomatisasi pengambilan data publikasi ilmiah dosen melalui teknik crawling/scraping guna mengurangi input manual dan meningkatkan akurasi data (Anugrah et al., 2025). Dengan solusi ini diharapkan proses administrasi DUPAK menjadi lebih cepat, terstruktur, transparan, serta mampu mengurangi potensi kesalahan validasi maupun keterlambatan rekapitulasi.

3.2.3 Desain dan pengembangan artefak

Tahap desain dan pengembangan artefak merupakan proses perancangan dan pembangunan sistem pengelolaan DUPAK berbasis web sebagai solusi dari permasalahan yang telah diidentifikasi. Perancangan sistem dilakukan menggunakan pendekatan Unified Modeling Language (UML) untuk memodelkan kebutuhan dan alur sistem secara terstruktur, meliputi use case diagram, activity diagram, sequence diagram, dan class diagram (Anugrah et al., 2025). Selain itu, tahap ini juga mencakup analisis kebutuhan sistem, penyusunan flowchart alur proses, perancangan antarmuka (wireframe/mockup), penentuan arsitektur sistem, serta desain mekanisme crawling data publikasi dari Google Scholar.

3.2.3.1 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan untuk mengidentifikasi kebutuhan fungsional dan non-fungsional dari sistem pengelolaan DUPAK berbasis web. Kebutuhan ini disusun berdasarkan hasil observasi proses pengelolaan DUPAK di Fakultas Teknik Universitas Muhammadiyah Surabaya serta wawancara dengan dosen dan admin sebagai calon pengguna sistem. Tujuannya adalah memastikan sistem yang dibangun mampu menjawab permasalahan utama, yaitu proses manual input, validasi data publikasi, dan pengajuan DUPAK yang belum terintegrasi serta memerlukan otomatisasi melalui teknologi web dan crawling data publikasi.

1. Kebutuhan Fungsional

Kebutuhan fungsional menjelaskan fungsi utama yang harus tersedia dan dapat dijalankan oleh sistem agar mampu mendukung proses pengelolaan DUPAK secara digital. Sistem dirancang untuk memfasilitasi dosen dalam mengajukan DUPAK secara mandiri dan admin dalam melakukan validasi,

monitoring, serta manajemen data publikasi yang diambil secara otomatis melalui proses crawling.

Fungsi utama yang harus tersedia pada sistem adalah sebagai berikut:

- a. Sistem menyediakan fitur login dan manajemen akses pengguna (dosen dan admin) sesuai hak akses masing-masing.
- b. Sistem mampu mengambil data publikasi dosen secara otomatis (crawling/scraping) dari sumber eksternal seperti Google Scholar.
- c. Sistem menampilkan dan menyimpan hasil crawling ke database, serta memungkinkan dosen melakukan validasi atau koreksi data.
- d. Sistem memungkinkan dosen mengisi dan mengajukan komponen DUPAK sesuai kategori yang berlaku.
- e. Sistem menampilkan status pengajuan DUPAK (diproses, perlu revisi, atau disetujui) kepada dosen.
- f. Admin dapat mengelola data dosen, data pengajuan, dan hasil publikasi crawling.
- g. Sistem menyediakan dashboard monitoring untuk menampilkan ringkasan data pengajuan, status verifikasi, dan data publikasi.
- h. Sistem mampu menyimpan riwayat pengajuan dan perubahan data untuk keperluan audit dan pelacakan.
- i. Sistem memberikan notifikasi informasi perubahan status pengajuan kepada dosen melalui tampilan sistem.

2. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional menjelaskan aspek kualitas dari sistem seperti performa, keamanan, kemudahan penggunaan, serta batasan teknis yang harus dipenuhi agar sistem dapat berjalan optimal dan layak digunakan dalam lingkungan akademik.

Kriteria non-fungsional yang harus dipenuhi sistem antara lain:

- a. Efisiensi dan Performa: Sistem harus mampu merespons permintaan pengguna dengan cepat, termasuk saat melakukan crawling atau memuat data dalam jumlah besar.

- b. Keamanan: Sistem wajib mengamankan autentikasi pengguna, membatasi akses berdasarkan hak peran (role), dan menjaga keamanan penyimpanan data.
- c. Ketersediaan (Availability): Sistem dapat diakses kapan saja melalui perangkat yang terhubung internet, khususnya saat proses pengajuan DUPAK berlangsung.
- d. Usability (Kemudahan Penggunaan): Antarmuka sistem harus sederhana, mudah dipahami, dan tidak menyulitkan pengguna awam (dosen dan admin).
- e. Kompatibilitas Perangkat: Sistem berbasis web dan dapat digunakan melalui berbagai perangkat seperti laptop maupun smartphone.
- f. Maintainability: Struktur sistem dan kode harus mudah dikelola dan dikembangkan kembali untuk penambahan fitur di masa depan.
- g. Reliability: Sistem mampu menjaga konsistensi data, menangani kegagalan proses crawling, serta memberikan notifikasi jika terjadi error.

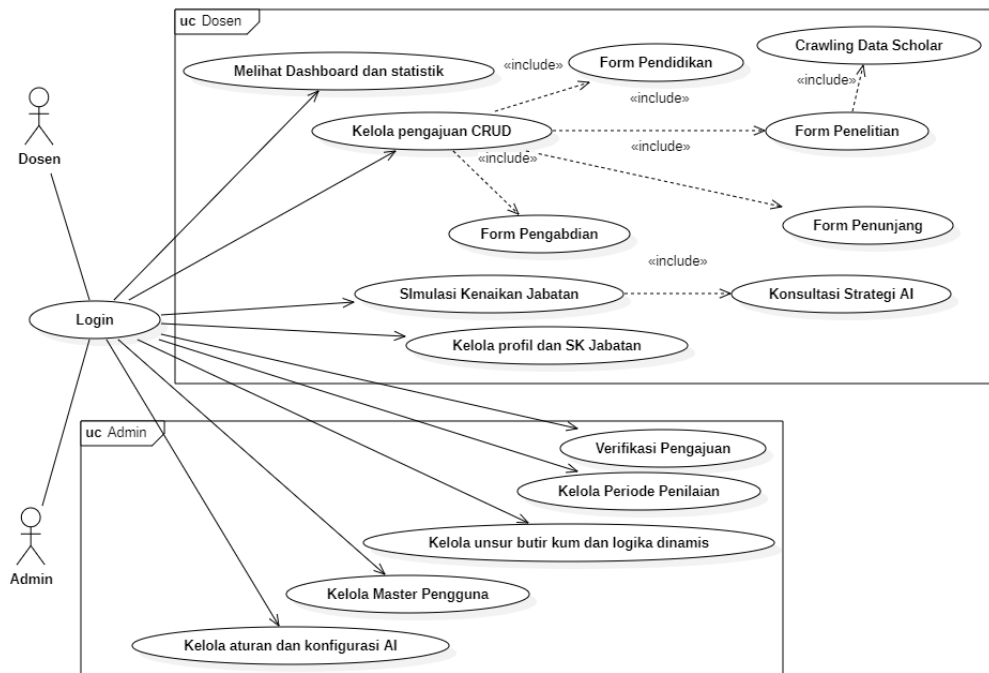
3.2.3.2 Perancangan Sistem

Perancangan sistem dilakukan untuk memodelkan kebutuhan dan alur proses dari Sistem Pengelolaan DUPAK berbasis web secara terstruktur sebelum masuk ke tahap implementasi. Pemodelan sistem menggunakan pendekatan Unified Modeling Language (UML) agar setiap proses, interaksi pengguna, hingga hubungan antar data dapat dipahami secara jelas oleh pengembang dan pemangku kepentingan. Diagram yang digunakan meliputi Use Case Diagram untuk menggambarkan interaksi aktor dan sistem, Activity Diagram untuk memetakan alur proses, Sequence Diagram untuk menunjukkan urutan komunikasi antar objek, dan Class Diagram untuk merepresentasikan struktur data serta relasi antar entitas. Selain itu, rancangan alur logika sistem juga divisualisasikan melalui flowchart untuk memperjelas proses operasional seperti alur login, pengajuan DUPAK, dan proses crawling publikasi.

1. Use Case Diagram

Use Case Diagram menggambarkan hubungan antara aktor (pengguna) dan sistem. Dalam sistem pengelolaan DUPAK ini terdapat dua aktor utama,

yaitu Admin dan Dosen, yang masing-masing memiliki hak akses dan aktivitas berbeda, seperti pengelolaan data, pengajuan DUPAK, dan pemantauan status pengajuan.



Gambar 2 Use Case Diagram

Admin berperan dalam memverifikasi data, memproses hasil penilaian, dan mengelola data publikasi hasil crawling. Hubungan antar aktor dan sistem menunjukkan batas interaksi dan tanggung jawab masing-masing.

2. Use Case Deskripsi

Deskripsi use case digunakan untuk menjelaskan secara rinci interaksi antara aktor dan sistem berdasarkan skenario yang terdapat pada diagram use case. Setiap use case memiliki alur utama, alur alternatif, serta kondisi yang menggambarkan bagaimana sistem merespons tindakan pengguna. Berikut ini merupakan deskripsi dari masing-masing use case yang terdapat pada sistem pengelolaan DUPAK berbasis web.

Table 1 Use Case Deskripsi – Login

Nama Use Case	Login
Created By	Qudum Awang Dwi Cahyo
Aktor	Dosen, Admin
Deskripsi	Proses autentikasi pengguna agar dapat mengakses sistem sesuai hak aksesnya.
Trigger	Pengguna menekan tombol “Login” setelah mengisi email dan password.
Precondition	Pengguna sudah memiliki akun yang terdaftar.
Postcondition	Pengguna berhasil masuk ke sistem.
Normal Flow	1. Pengguna membuka halaman login.
	2. Memasukkan email dan password.
	3. Sistem memverifikasi data.
	4. Sistem menampilkan dashboard sesuai peran pengguna.
Alternative Flow	1. Pengguna lupa password -> diarahkan ke halaman reset password.
	2. Pengguna salah mengetik email atau password.
Exceptions	1. Kredensial Salah: Jika email/password tidak cocok, sistem menampilkan pesan "Email atau password salah" dan tetap di halaman login.
	2. Koneksi Database Gagal: Jika database down, sistem menampilkan halaman error 500.
	3. Input Kosong: Jika form dikirim kosong, validasi frontend menampilkan peringatan "Wajib diisi".
Priority	Tinggi
Frequency of Use	Setiap kali pengguna akan masuk ke sistem.
Business Rule	Setiap akun memiliki hak akses tertentu (Admin atau Dosen).
Other Information	Password disimpan dalam bentuk terenkripsi.
Assumptions	Pengguna telah memiliki akun aktif yang terdaftar.

Table 2 Use Case Deskripsi – Kelola Profil dan SK Jabatan

Nama Use Case	Kelola Profil & SK Jabatan
Created By	Qudum Awang Dwi Cahyo
Aktor	Dosen
Deskripsi	Dosen dapat mengisi dan memperbarui biodata pribadi seperti NIDN, nama, prodi, dan jabatan fungsional.
Trigger	Dosen memilih menu “Biodata” di dashboard.
Precondition	Dosen sudah login ke sistem.
Postcondition	Data profil diperbarui, SK tersimpan, dan nilai angka_kredit_terakhir di-reset sesuai jabatan baru.
Normal Flow	1. Sistem menampilkan form biodata.
	2. Dosen mengisi atau memperbarui data pribadi serta mengunggah data jabatan akademik baru
	3. Sistem menyimpan data ke database.

	4. Sistem menampilkan pesan sukses.
Alternative Flow	Dosen menekan batal sebelum menyimpan perubahan.
Exceptions	1. File Terlalu Besar: Jika PDF > 5MB, sistem menolak upload dan menampilkan pesan error.
	2. Format File Salah: Jika bukan PDF, sistem menolak.
	3. Gagal Upload Drive: Jika koneksi ke Google Drive API putus, sistem membatalkan simpan data dan memberi notifikasi "Gagal upload SK".
Priority	Tinggi
Frequency of Use	Saat awal registrasi atau pembaruan data dosen.
Business Rule	Biodata wajib diisi sebelum input DUPAK.
Other Information	Data diambil untuk identitas di laporan DUPAK.
Assumptions	Dosen sudah login ke sistem.

Table 3 Use Case Deskripsi – Crawling Data Publikasi

Nama Use Case	Crawling Data Publikasi
Created By	Qudum Awang Dwi Cahyo
Aktor	Sistem (dipicu oleh Dosen)
Deskripsi	Proses otomatis pengambilan dan pengisian awal data publikasi dari Google Scholar ke dalam Form Penelitian.
Trigger	Dipanggil dari use case Form Penelitian saat Dosen menekan tombol "Ambil Data Publikasi".
Precondition	Dosen sudah login dan berada di menu Penelitian.
Postcondition	Data publikasi berhasil disimpan di sistem dan ditampilkan pada daftar penelitian dosen.
Normal Flow	1. Dosen menekan tombol "Scholar".
	2. Sistem (Laravel) mengirim webhook ke server n8n dengan ID Scholar dosen.
	3. Sistem menampilkan loading spinner dan melakukan polling status setiap 3 detik.
	4. n8n melakukan scraping data dan mengirim balik hasil ke Laravel (API Callback).
	5. Laravel menyimpan hasil di Cache.
	6. Sistem menampilkan daftar publikasi di modal frontend.
Alternative Flow	Jika sumber tidak merespons maka sistem menampilkan notifikasi "Gagal mengambil data, silakan coba lagi".
Exceptions	1. ID Scholar Kosong: Sistem menolak request dan meminta dosen melengkapi profil.
	2. Timeout n8n: Jika n8n tidak merespons dalam 60 detik, sistem menghentikan polling dan menampilkan "Gagal terhubung ke server crawling".
	3. Profil Scholar Private: n8n mengembalikan data kosong atau error karena profil diprivat oleh dosen.

	4. Limit API: Jika request terlalu banyak dalam satu waktu, sistem menampilkan "Server sibuk, coba lagi nanti".
Priority	Tinggi
Frequency of Use	Sesuai kebutuhan Dosen saat berada di Form Penelitian dan ingin menggunakan fitur otomatisasi data.
Business Rule	Data yang disimpan harus mengikuti standar format komponen penelitian DUPAK terbaru.
Other Information	-
Assumptions	Internet tersedia dan sumber web (Google Scholar) dapat diakses.

Table 4 Use Case Deskripsi – Kelola Pengajuan (CRUD)

Nama Use Case	Kelola Pengajuan (CRUD)
Created By	Qudum Awang Dwi Cahyo
Aktor	Dosen
Deskripsi	Dosen dapat melihat daftar data DUPAK, menambah, memperbarui, dan menghapus data yang telah diinput pada komponen (Pendidikan, Penelitian, dll.).
Trigger	Dosen memilih menu "Data DUPAK" dan/atau menekan tombol aksi (Tambah, Edit, Hapus).
Precondition	Dosen sudah login dan memiliki data DUPAK.
Postcondition	Data tersimpan dengan status 'Pending' dan muncul di riwayat.
Normal Flow	1. Dosen memilih tab kategori (misal: Penelitian).
	2. Dosen menekan "Tambah Usulan".
	3. Sistem menampilkan formulir dinamis sesuai butir kegiatan.
	4. Dosen melengkapi data dan mengunggah bukti PDF.
	5. Sistem menghitung Angka Kredit (AK) otomatis berdasarkan rumus.
	6. Dosen menekan "Simpan".
	7. Sistem menyimpan data ke database.
Alternative Flow	1. Dosen memilih opsi import dari Google Scholar (Lanjut ke Crawling).
	2. Dosen membatalkan perubahan sebelum menyimpan.
Exceptions	1. Periode Tutup: Jika tidak ada periode aktif, sistem menolak penyimpanan.
	2. Duplikasi Gelar: Jika dosen menginput gelar yang sudah dimiliki (misal S2 padahal sudah S3), sistem menolak.
	3. Gagal Hitung Rumus: Jika rumus dinamis error (misal pembagian nol), sistem menggunakan nilai default dan mencatat log error.
Priority	Tinggi
Frequency of Use	Sering, sesuai kebutuhan Dosen untuk Input, Koreksi, dan Monitoring data DUPAK per periode.
Business Rule	Edit hanya bisa dilakukan sebelum verifikasi admin.
Other Information	Perubahan dicatat dalam log sistem.
Assumptions	Dosen memiliki data DUPAK aktif.

Table 5 Use Case Deskripsi – Simulasi Kenaikan Jabatan

Nama Use Case	Simulasi Kenaikan Jabatan
Created By	Qudum Awang Dwi Cahyo
Aktor	Dosen
Deskripsi	Dosen dapat melihat daftar data DUPAK, menambah, memperbarui, dan menghapus data yang telah diinput pada komponen (Pendidikan, Penelitian, dll.).
Trigger	Dosen mengakses menu dashboard dan mengubah pilihan pada dropdown "Target Jabatan".
Precondition	Dosen memiliki data poin yang disetujui (Approved).
Postcondition	Tampil tabel kekurangan poin per unsur.
Normal Flow	1. Dosen memilih "Target Jabatan" pada dropdown.
	2. Sistem mengambil aturan nilai minimal dari database.
	3. Sistem menghitung total poin dosen saat ini.
	4. Sistem menampilkan tabel perbandingan (Poin Saat Ini vs Target Wajib/Aman).
Alternative Flow	Jika dosen memilih target jabatan di bawah jabatan saat ini, sistem tidak menampilkan tabel simulasi dan hanya menampilkan info jabatan saat ini.
Exceptions	1. Loncat Jabatan: Jika dosen memilih target 2 tingkat di atas jabatan sekarang (misal: Asisten Ahli ke Lektor Kepala), sistem menampilkan peringatan "Tidak Dapat Loncat Jabatan".
	2. Data Kosong: Jika dosen belum punya riwayat sama sekali, simulasi tetap jalan dengan nilai awal 0.
Priority	Tinggi
Frequency of Use	Sesuai kebutuhan, biasanya saat dosen ingin merencanakan karir atau mengecek kelayakan naik pangkat.
Business Rule	Perhitungan simulasi menggunakan aturan terbaru yang aktif di database (Aturan Kenaikan).
Other Information	Hasil simulasi bersifat estimasi dan bukan keputusan final penilaian.
Assumptions	Data riwayat pengajuan dosen sudah lengkap dan terverifikasi (status Approved).

Table 6 Use Case Deskripsi – Konsultasi Strategi AI

Nama Use Case	Konsultasi Strategi AI
Created By	Qudum Awang Dwi Cahyo
Aktor	Dosen
Deskripsi	Meminta saran naratif dari AI mengenai cara memenuhi kekurangan angka kredit.
Trigger	Dosen menekan tombol "Tanya AI" pada tabel simulasi.
Precondition	Terdapat kekurangan poin pada hasil simulasi.
Postcondition	Muncul popup berisi teks saran strategi.
Normal Flow	1. Dosen klik "Tanya AI".
	2. Sistem mengirim data (Jabatan, Bidang Ilmu, Kekurangan Poin) ke API Google Gemini.

	3. Google Gemini memproses prompt dan mengembalikan teks saran.
	4. Sistem memformat teks dan menampilkannya di modal.
Alternative Flow	Jika request gagal, dosen dapat menekan tombol "Coba Lagi" pada pesan error.
Exceptions	1. API Error/Quota Habis: Jika API Google Gemini error (429 Too Many Requests), sistem menampilkan pesan "AI sedang sibuk, coba lagi nanti".
	2. Koneksi Internet: Jika server tidak bisa menghubungi Google, tampil pesan error koneksi.
Priority	Tinggi
Frequency of Use	Jarang, hanya digunakan ketika dosen membutuhkan saran strategi pemenuhan poin.
Business Rule	Fitur hanya aktif jika hasil simulasi menunjukkan adanya kekurangan poin (gap negatif).
Other Information	Saran dihasilkan oleh Generative AI (Google Gemini) berdasarkan prompt yang dikonfigurasi admin.
Assumptions	API Key Google Gemini valid dan kuota mencukupi.

Table 7 Use Case Deskripsi – Verifikasi Pengajuan DUPAK

Nama Use Case	Verifikasi Pengajuan DUPAK
Created By	Qudum Awang Dwi Cahyo
Aktor	Admin
Deskripsi	Admin memverifikasi kelengkapan dan validitas data DUPAK dosen.
Trigger	Admin membuka data DUPAK dosen untuk pemeriksaan.
Precondition	Terdapat data pengajuan berstatus 'Pending'.
Postcondition	Status pengajuan berubah menjadi 'Approved' atau 'Rejected'.
Normal Flow	1. Admin melihat daftar usulan masuk.
	2. Admin membuka file bukti (Link Google Drive).
	3. Admin memeriksa kesesuaian data.
	4. Admin memasukkan "Nilai Disetujui" (bisa diedit).
	5. Admin klik "Setujui".
	6. Sistem memperbarui status dan nilai di database.
Alternative Flow	Admin memberikan catatan revisi dan mengirim kembali ke dosen.
Exceptions	1. File Hilang: Jika file di Google Drive terhapus, admin tidak bisa memverifikasi (muncul error 404 pada link).
	2. Konflik Data: Jika dosen menghapus pengajuan saat admin sedang memeriksa, sistem akan gagal simpan dan me-refresh halaman.
Priority	Tinggi
Frequency of Use	Setiap kali ada pengajuan DUPAK baru.
Business Rule	Verifikasi hanya dapat dilakukan oleh admin.
Other Information	Riwayat verifikasi dicatat dalam sistem.
Assumptions	Data DUPAK sudah lengkap.

Table 8 Use Case Deskripsi – Kelola Unsur dan Logika Dinamis

Nama Use Case	Kelola Unsur & Logika Dinamis
Created By	Qudum Awang Dwi Cahyo
Aktor	Admin
Deskripsi	Admin dapat mengatur daftar komponen DUPAK dan rumus perhitungan angka kredit (pendidikan, penelitian, pengabdian, penunjang) agar sesuai aturan terbaru.
Trigger	Admin membuka menu “Manajemen KUM”.
Precondition	Admin sudah login.
Postcondition	Konfigurasi form dan rumus tersimpan di database (JSON).
Normal Flow	1. Admin membuka menu "Manajemen KUM".
	2. Admin menambah butir baru atau edit yang ada.
	3. Admin menyusun field input (Teks, Angka, File) via Form Builder.
	4. Admin menyusun logika hitung via Visual Logic Tree (misal: Jika 'Penulis Utama' maka nilai * 60%).
	5. Admin simpan.
	6. Sistem memperbarui JSON di database.
Alternative Flow	Admin membatalkan perubahan.
Exceptions	1. Rumus Invalid: Jika admin membuat rumus matematika yang salah (misal pembagian nol), sistem menolak simpan.
	2. Kode Duplikat: Jika kode alias (misal III.A.1) sudah ada, sistem menolak.
Priority	Sedang
Frequency of Use	Saat ada pembaruan regulasi penilaian.
Business Rule	Komponen harus sesuai peraturan terbaru Kemendikbud.
Other Information	Data komponen digunakan untuk perhitungan DUPAK.
Assumptions	Admin memiliki hak akses penuh.

Table 9 Use Case Deskripsi – Kelola Periode Penilaian

Nama Use Case	Kelola Periode Penilaian
Created By	Qudum Awang Dwi Cahyo
Aktor	Admin
Deskripsi	Mengatur sesi buka/tutup periode penilaian DUPAK (Semesteran/Tahunan).
Trigger	Admin menekan tombol "Buat Periode Baru" atau tombol "Aktifkan" pada tabel periode.
Precondition	Admin sudah login dan berada di halaman Manajemen Periode.
Postcondition	Hanya satu periode yang aktif, periode lain dinonaktifkan.
Normal Flow	1. Admin membuat periode baru (Start Date - End Date).
	2. Admin menekan "Aktifkan".
	3. Sistem menonaktifkan semua periode lain dan mengaktifkan periode terpilih.

Alternative Flow	Admin menonaktifkan periode yang sedang berjalan tanpa mengaktifkan yang baru (semua periode non-aktif).
Exceptions	1. Tanggal Terbalik: Tanggal selesai lebih awal dari tanggal mulai. 2. Hapus Periode Aktif: Periode yang sedang aktif tidak boleh dihapus jika sudah ada data pengajuan di dalamnya.
Priority	Tinggi
Frequency of Use	Jarang, biasanya dilakukan setiap semester.
Business Rule	Hanya boleh ada satu periode aktif dalam satu waktu untuk mencegah data masuk ke periode yang salah.
Other Information	Data pengajuan dosen akan otomatis terikat (foreign key) ke periode yang aktif saat itu.
Assumptions	Tidak ada proses input data oleh dosen saat pergantian periode dilakukan.

Table 10 Use Case Deskripsi – Kelola Master Pengguna

Nama Use Case	Kelola Master Pengguna
Created By	Qudum Awang Dwi Cahyo
Aktor	Admin
Deskripsi	Admin dapat menambah akun dosen baru, mengedit data akun, mereset password, dan menghapus akun pengguna.
Trigger	Admin memilih menu "Manajemen User" dan menekan tombol Tambah/Edit/Hapus.
Precondition	Admin sudah login sebagai Administrator.
Postcondition	Data pengguna (Dosen/Admin) berhasil diperbarui di database.
Normal Flow	1. Admin membuka halaman Manajemen User. 2. Admin menekan tombol "Tambah User". 3. Admin mengisi Nama, Email, Password, dan Role. 4. Admin menekan "Simpan". 5. Sistem menyimpan data user baru.
Alternative Flow	Admin memilih user, mengosongkan kolom lain, dan hanya mengisi field password baru untuk mereset.
Exceptions	1. Email Duplikat: Jika email sudah terdaftar, sistem menolak dan menampilkan pesan error. 2. Hapus Diri Sendiri: Admin tidak bisa menghapus akun yang sedang login.
Priority	Tinggi
Frequency of Use	Jarang, saat ada dosen baru atau permintaan reset password.
Business Rule	Hanya boleh ada satu periode aktif dalam satu waktu untuk mencegah data masuk ke periode yang salah.
Other Information	-
Assumptions	-

Table 11 Use Case Deskripsi – Melihat Dashboard dan Statistik

Nama Use Case	Melihat Dashboard & Statistik
Created By	Qudum Awang Dwi Cahyo
Aktor	Dosen
Deskripsi	Dosen melihat ringkasan kinerja, total poin KUM yang disetujui, dan grafik perkembangan poin.
Trigger	Dosen berhasil login atau menekan menu "Dashboard".
Precondition	Dosen sudah login.
Postcondition	Informasi ringkasan tampil di layar utama.
Normal Flow	1. Sistem mengambil data rekapitulasi poin dosen dari database.
	2. Sistem menghitung total KUM Approved.
	3. Sistem menampilkan Card Status Jabatan, Total Poin, dan Grafik per kategori.
Alternative Flow	-
Exceptions	1. Data Kosong: Jika dosen baru, grafik menampilkan nilai 0 tanpa error.
Priority	Tinggi
Frequency of Use	Sering, setiap kali dosen login.
Business Rule	Hanya boleh ada satu periode aktif dalam satu waktu untuk mencegah data masuk ke periode yang salah.
Other Information	-
Assumptions	-

Table 12 Use Case Deskripsi – Kelola Aturan dan Konfigurasi AI

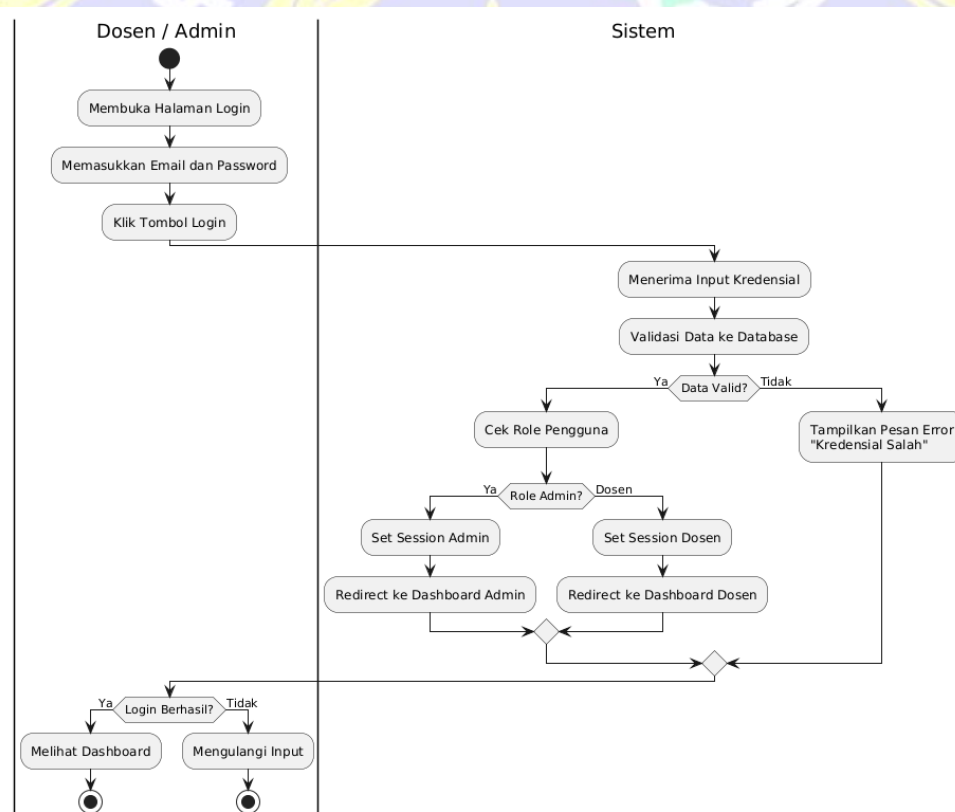
Nama Use Case	Kelola Aturan & Konfigurasi AI
Created By	Qudum Awang Dwi Cahyo
Aktor	Admin
Deskripsi	Admin mengatur parameter simulasi (Target AK, Persentase Min/Max) dan template prompt untuk AI.
Trigger	Admin mengubah nilai pada menu "Aturan Simulasi" atau "Konfigurasi".
Precondition	Admin sudah login.
Postcondition	Parameter perhitungan simulasi dan perilaku AI diperbarui.
Normal Flow	1. Admin membuka menu Aturan Simulasi.
	2. Admin mengedit target angka kredit untuk jabatan tertentu (misal: Lektor Kepala = 400).
	3. Admin mengubah persentase target aman (misal: 30%).
	4. Admin mengedit template prompt AI.
	5. Admin menekan tombol Simpan.
Alternative Flow	Admin membatalkan perubahan dan kembali ke menu sebelumnya.
Exceptions	1. Input Invalid: Jika persentase total melebihi 100% atau format angka salah, sistem menolak simpan dan menampilkan pesan error.

Priority	Sedang
Frequency of Use	Sangat Jarang, hanya saat ada perubahan kebijakan kampus/dikti.
Business Rule	Aturan simulasi mempengaruhi hasil perhitungan gap poin semua dosen.
Other Information	-
Assumptions	-

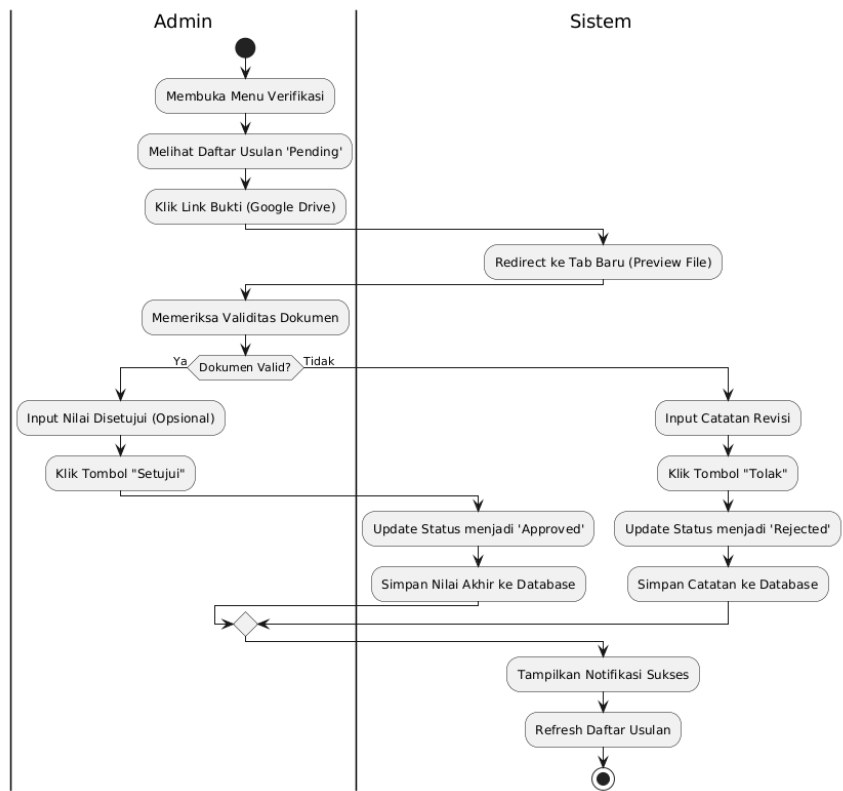
3. Activity Diagram

Activity Diagram menggambarkan alur aktivitas sistem dari perspektif pengguna. Karena sistem memiliki beberapa fungsi utama, maka dibuat 7 activity diagram berdasarkan use case utama.

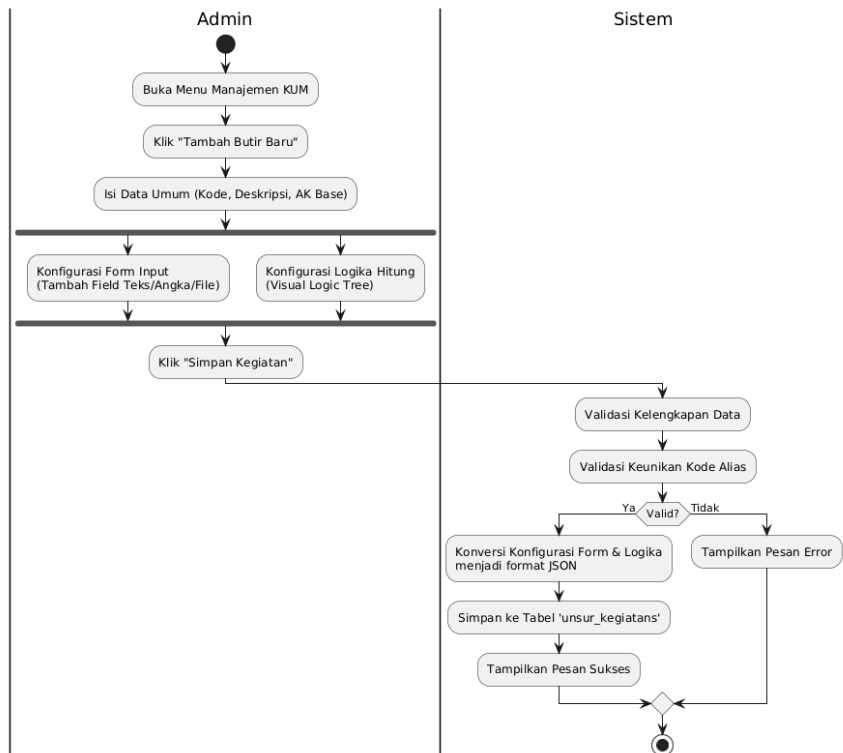
Proses dimulai dari dosen yang melakukan login, kemudian menginput data DUPAK. Sistem akan memproses data dan menampilkan status verifikasi. Jika data lengkap, maka pengajuan akan diteruskan ke admin untuk verifikasi. Jika tidak, sistem akan meminta revisi dari pengguna.



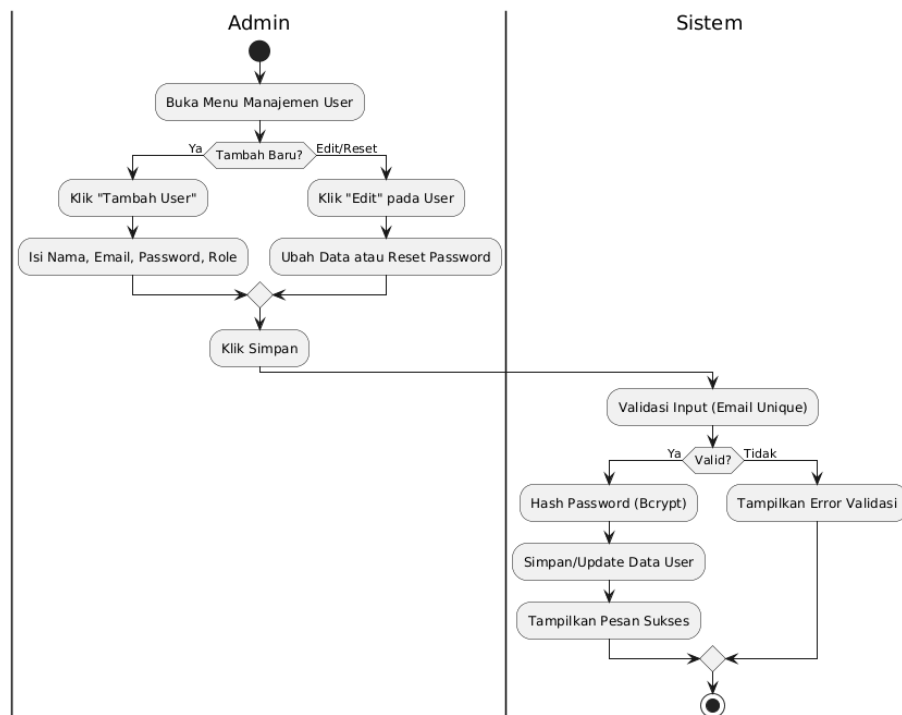
Gambar 3 Activity Diagram - Login



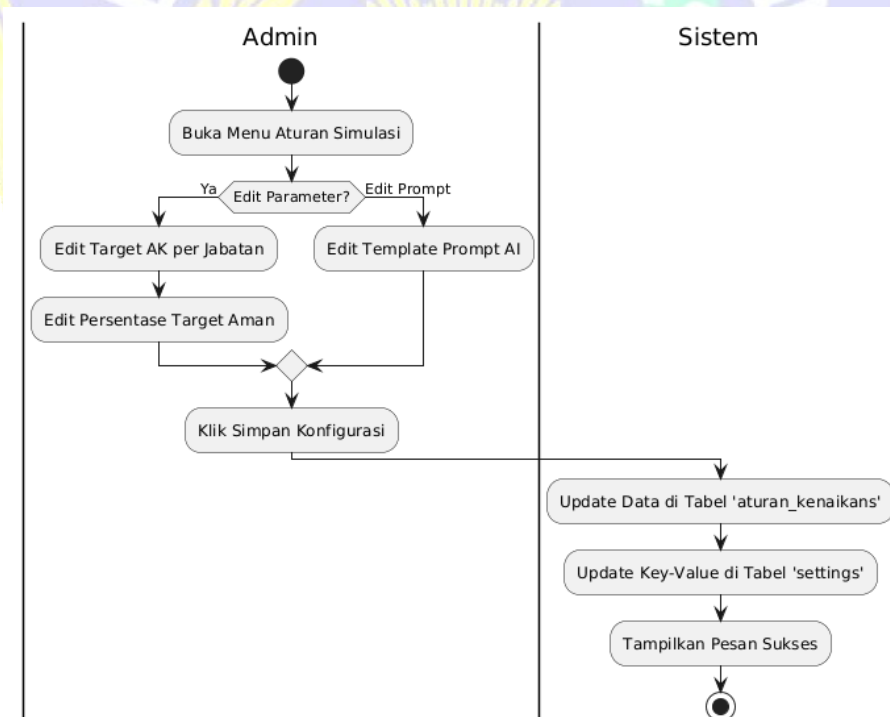
Gambar 4 Activity Diagram - Verifikasi Pengajuan



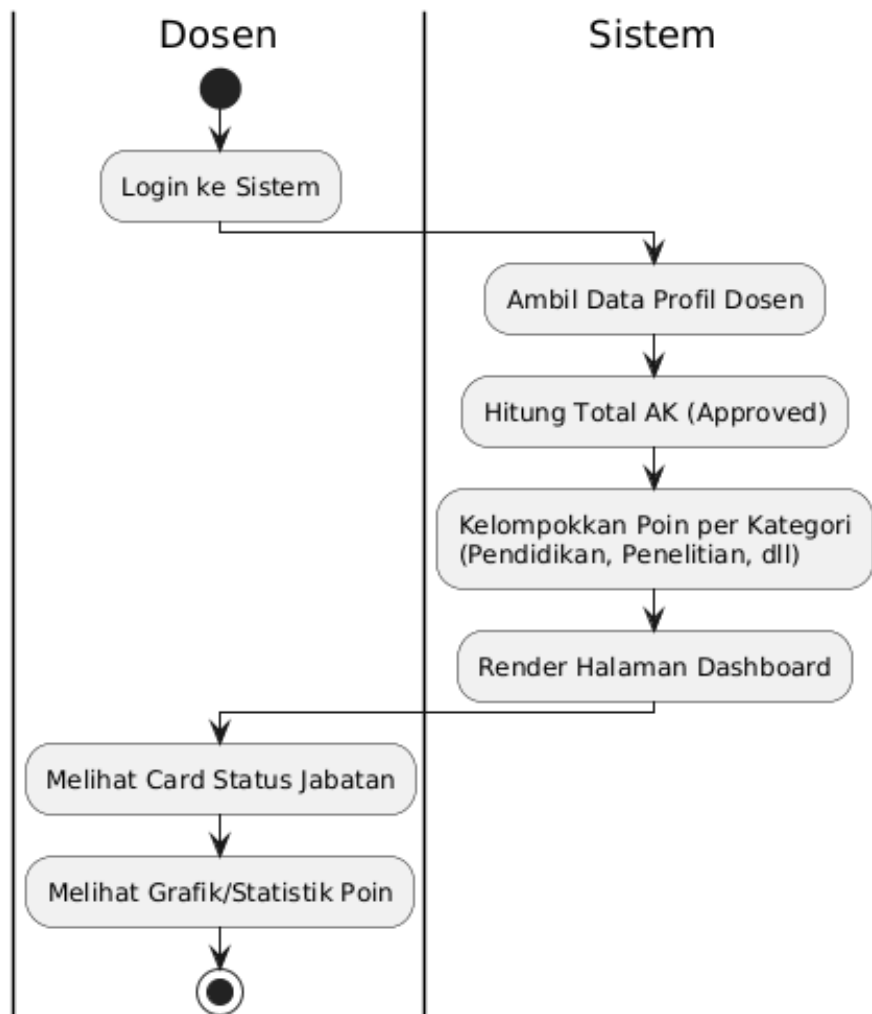
Gambar 5 Activity Diagram - Kelola Unsur & Logika Dinamis



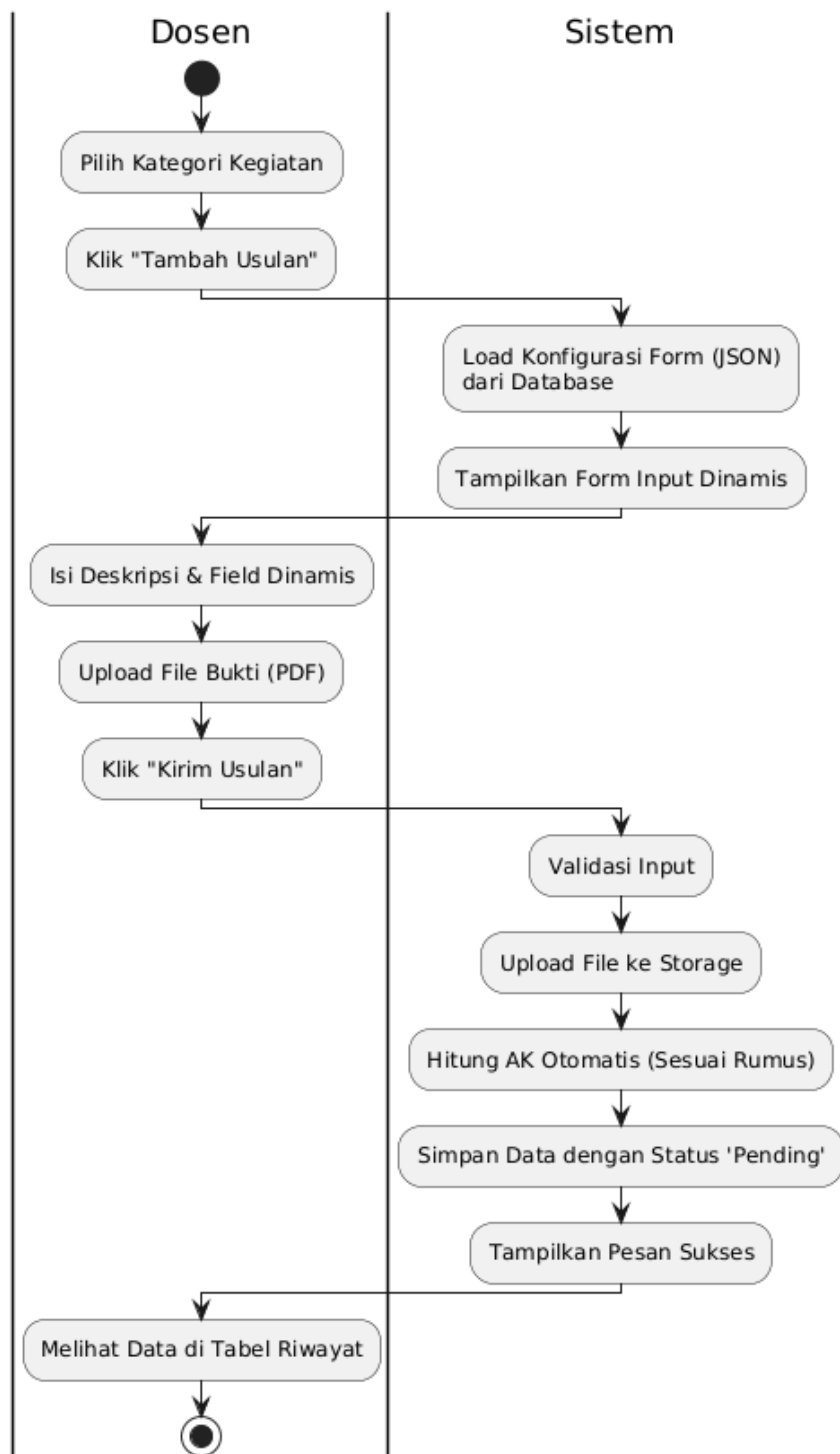
Gambar 6 Activity Diagram - Kelola Master Pengguna



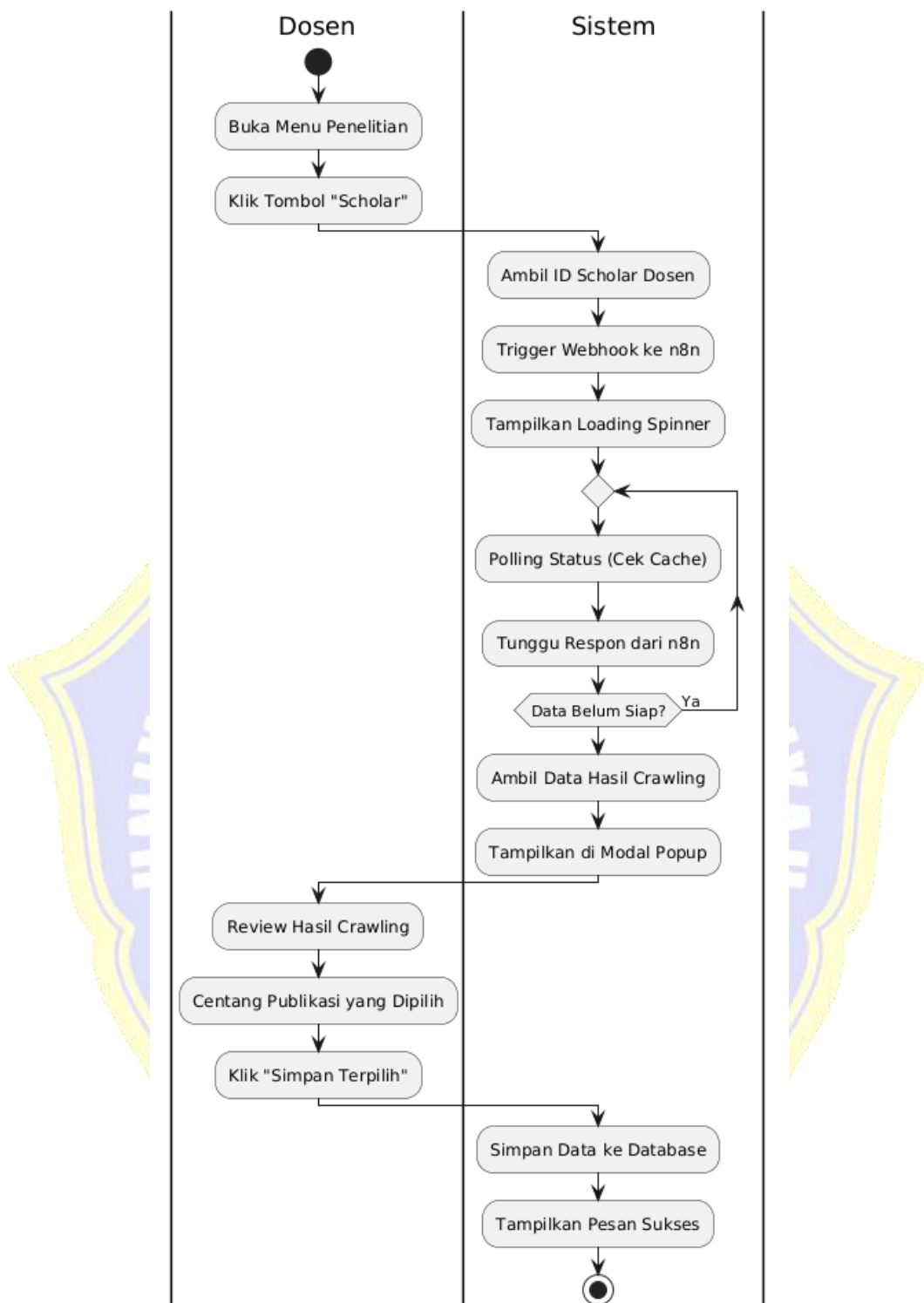
Gambar 7 Activity Diagram - Lihat Rekap



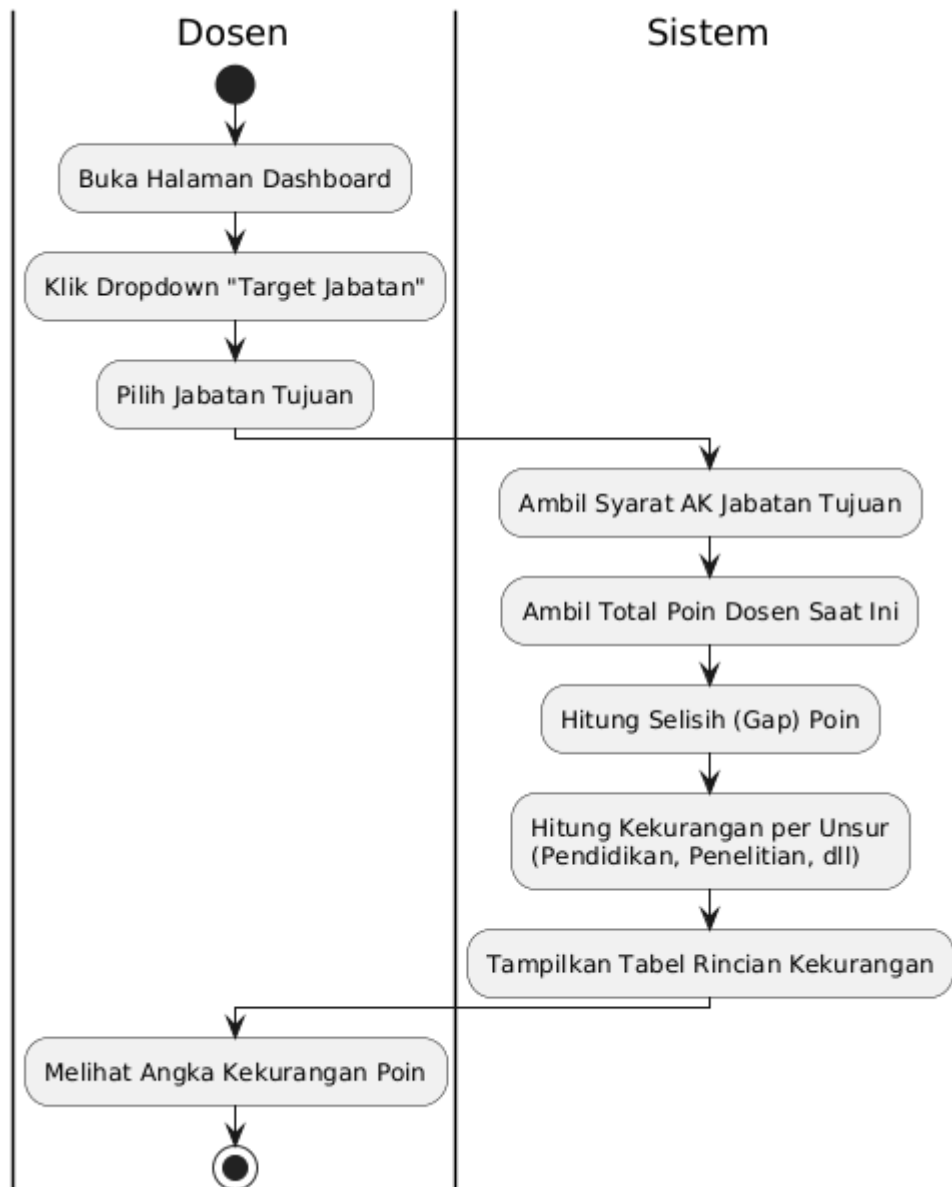
Gambar 8 Activity Diagram - Melihat Dashboard & Statistik



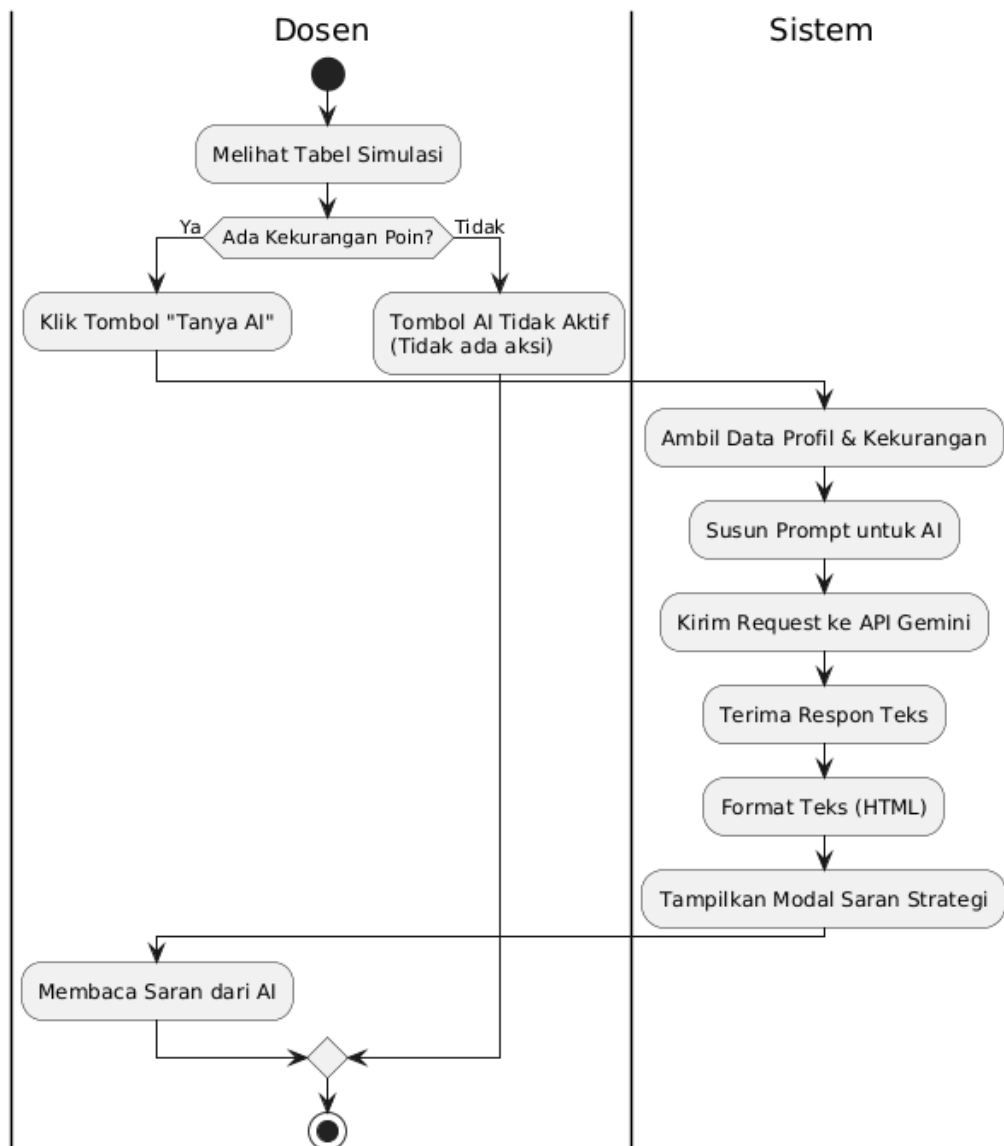
Gambar 9 Activity Diagram - Kelola Pengajuan (CRUD)



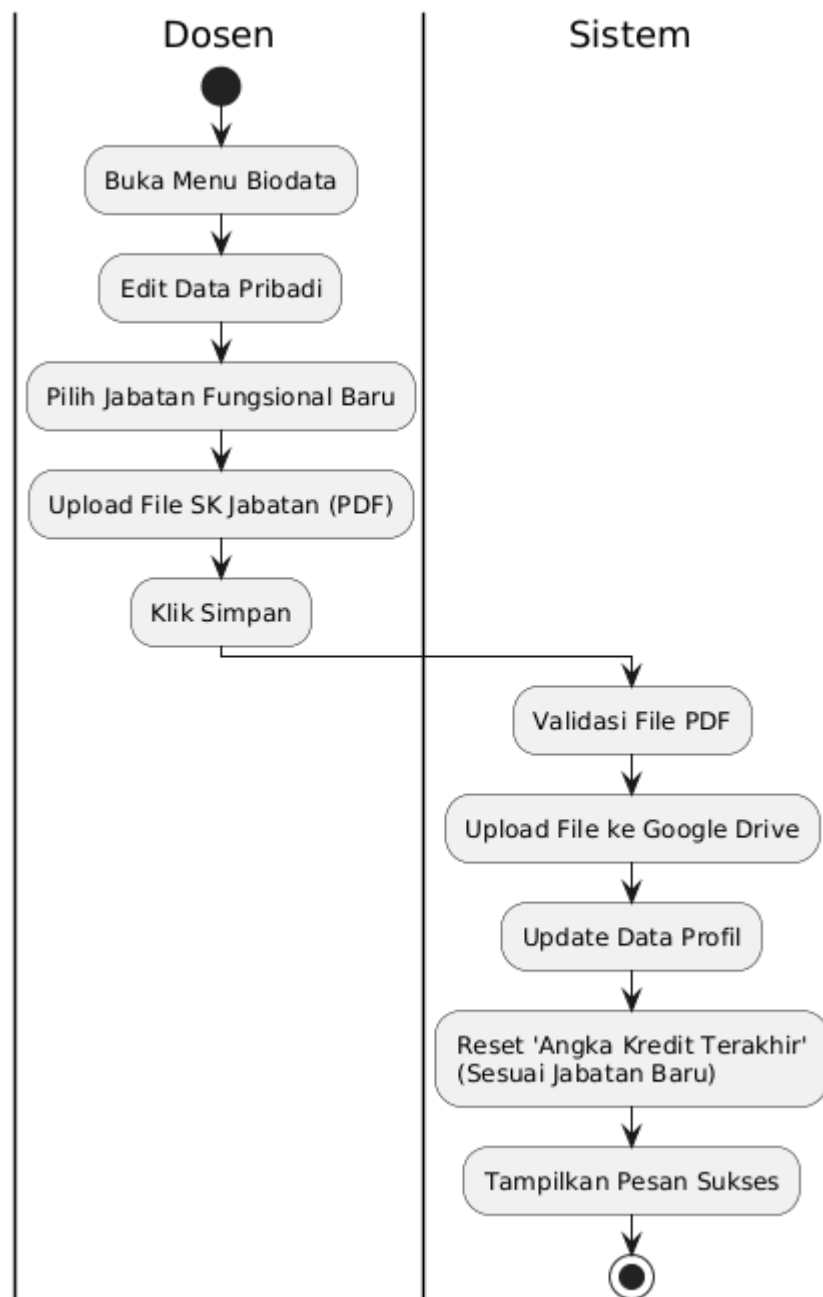
Gambar 10 Activity Diagram - Crawling Data Scholar



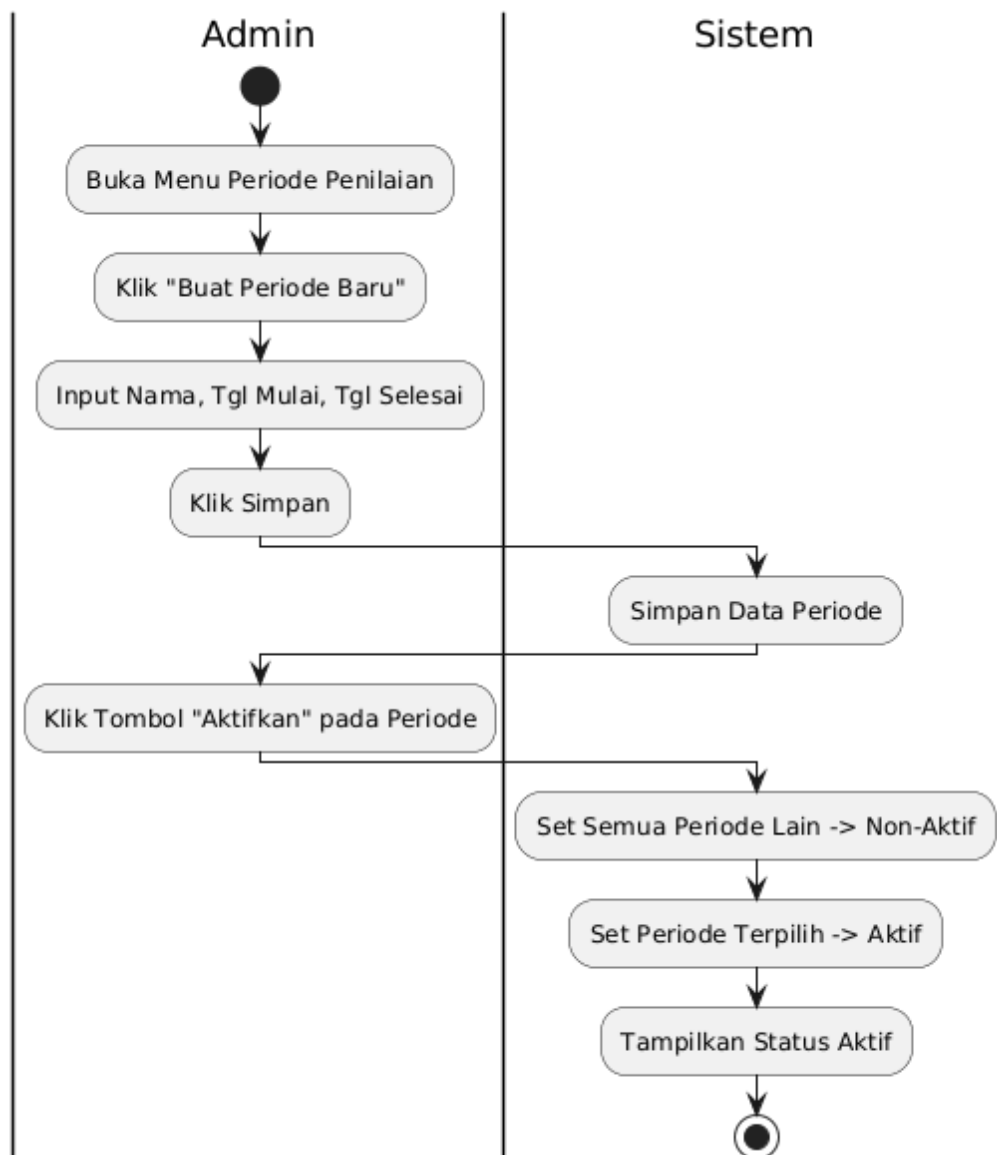
Gambar 11 Activity Diagram - Simulasi Kenaikan Jabatan



Gambar 12 Activity Diagram - Konsultasi Strategi AI



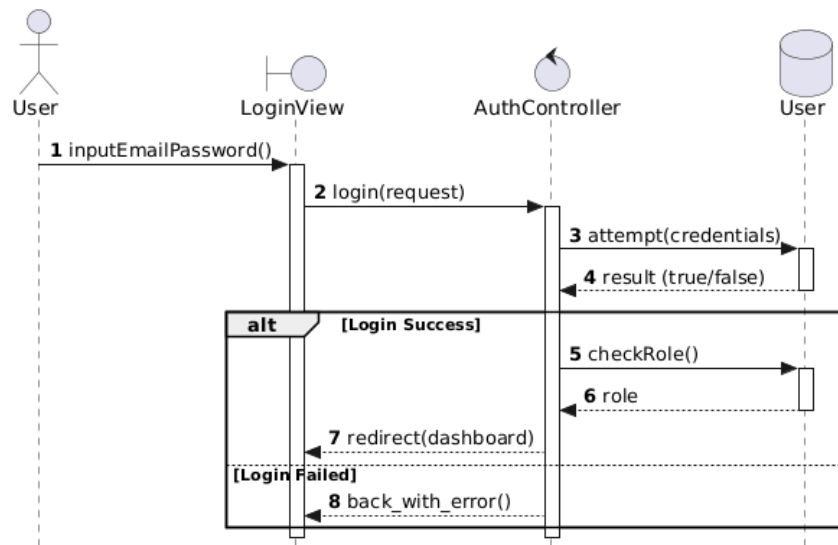
Gambar 13 Activity Diagram - Kelola Profil & SK Jabatan



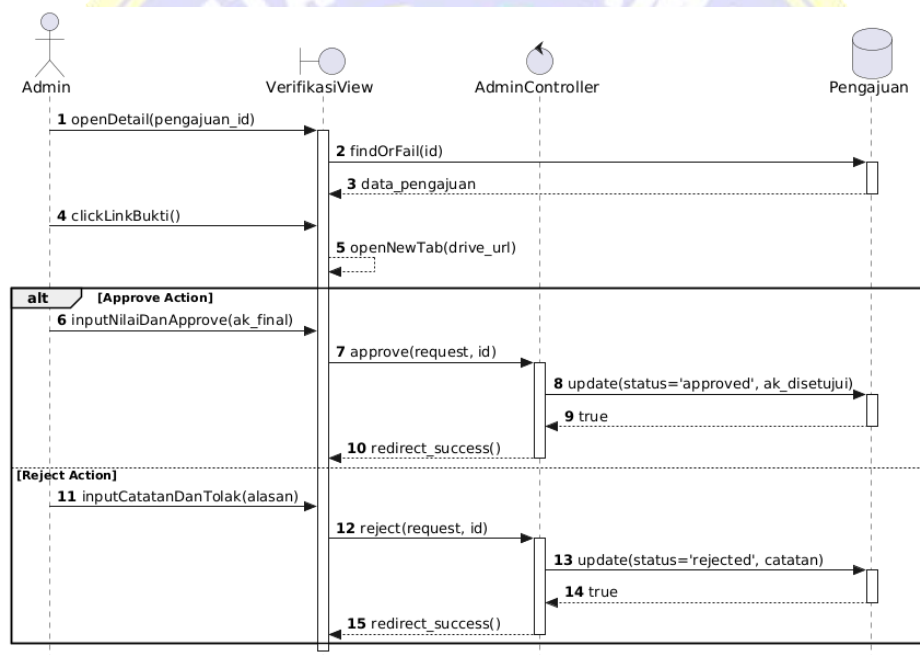
Gambar 14 Activity Diagram - Kelola Periode Penilaian

4. Sequence Diagram

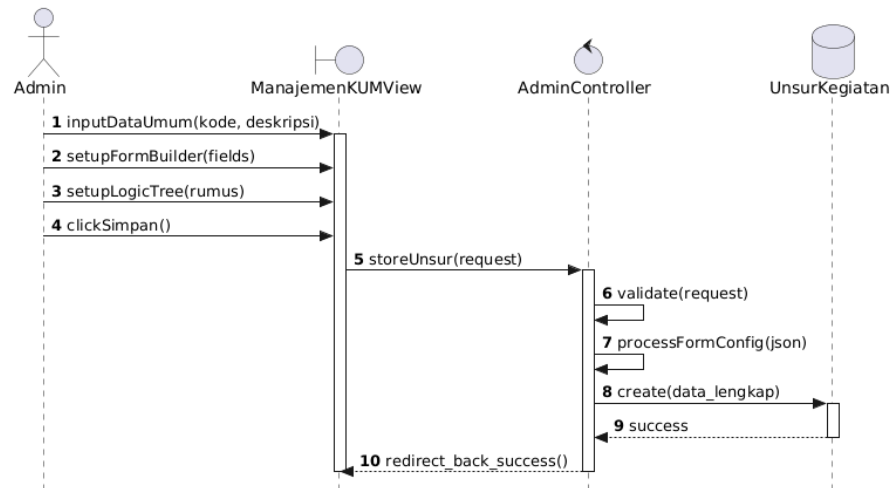
Sequence Diagram menggambarkan urutan interaksi antar objek untuk setiap proses dalam sistem. Karena sistem memiliki banyak proses utama, maka dibuat 7 sequence diagram yang mewakili setiap use case penting.



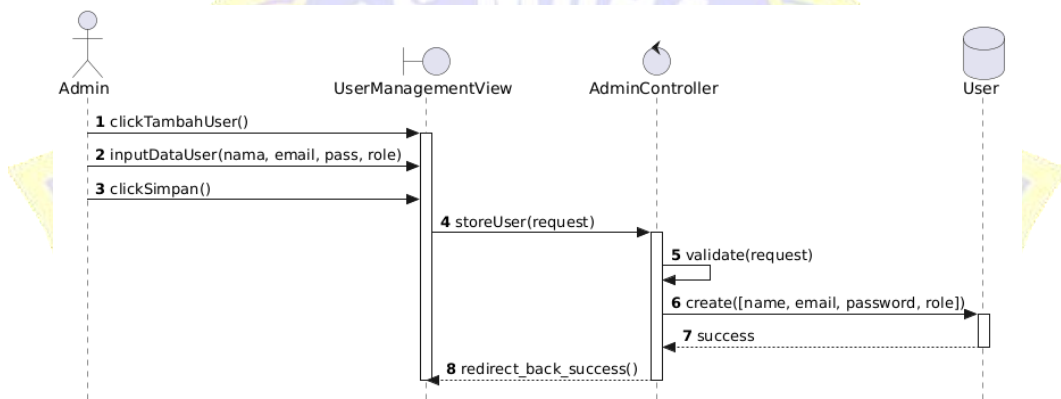
Gambar 15 Sequence Diagram – Login



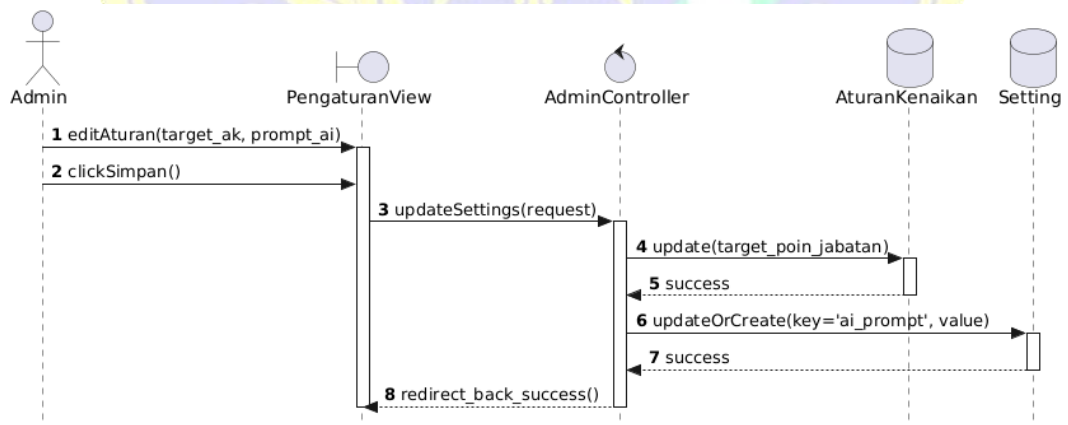
Gambar 16 Sequence Diagram - Verifikasi Pengajuan



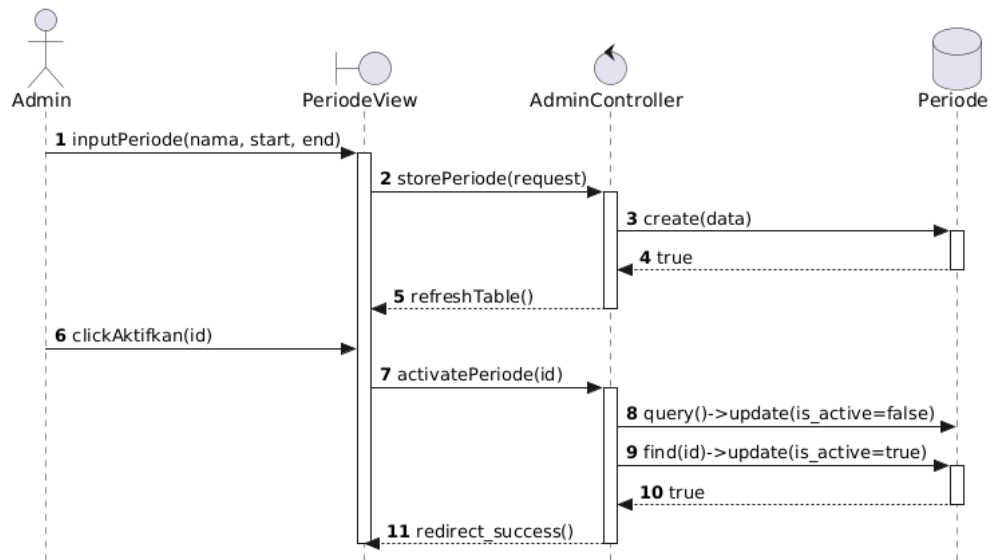
Gambar 17 Sequence Diagram – Kelola Unsur & Logika Dinamis



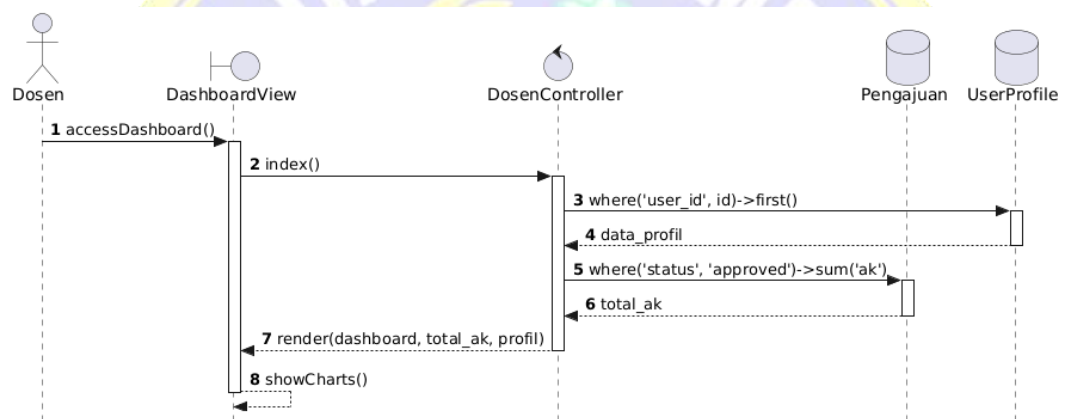
Gambar 18 Sequence Diagram – Kelola Master Pengguna



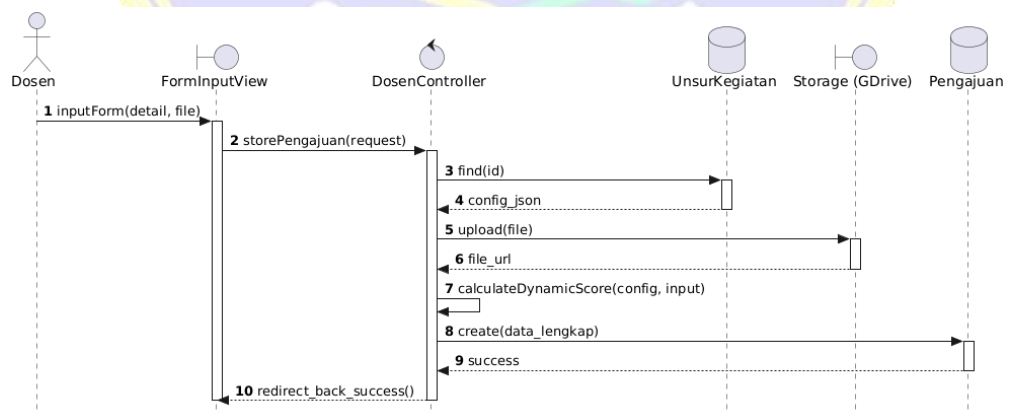
Gambar 19 Sequence Diagram - Kelola Aturan & Konfigurasi AI



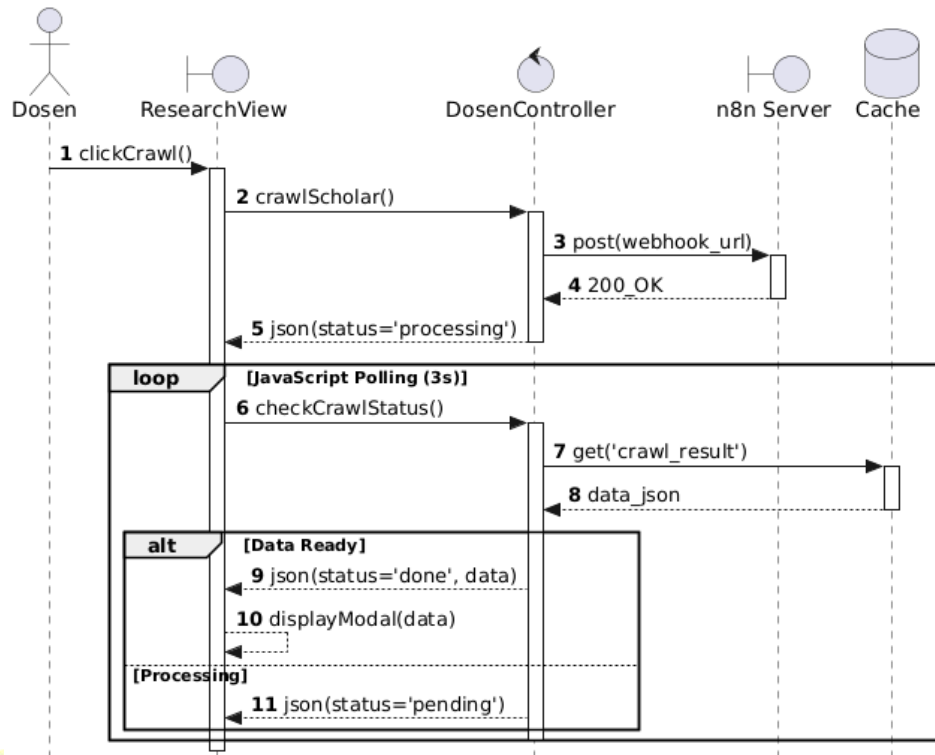
Gambar 20 Sequence Diagram - Melihat Dashboard & Statistik



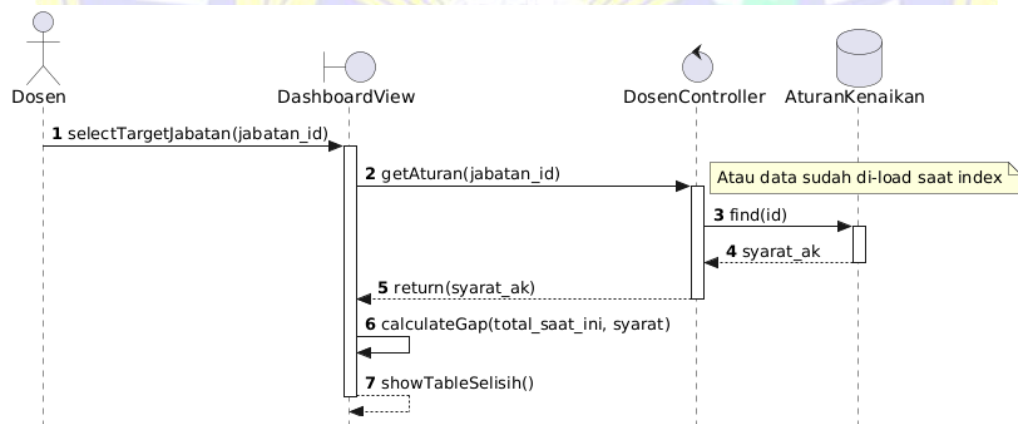
Gambar 21 Sequence Diagram - Kelola Pengajuan (CRUD)



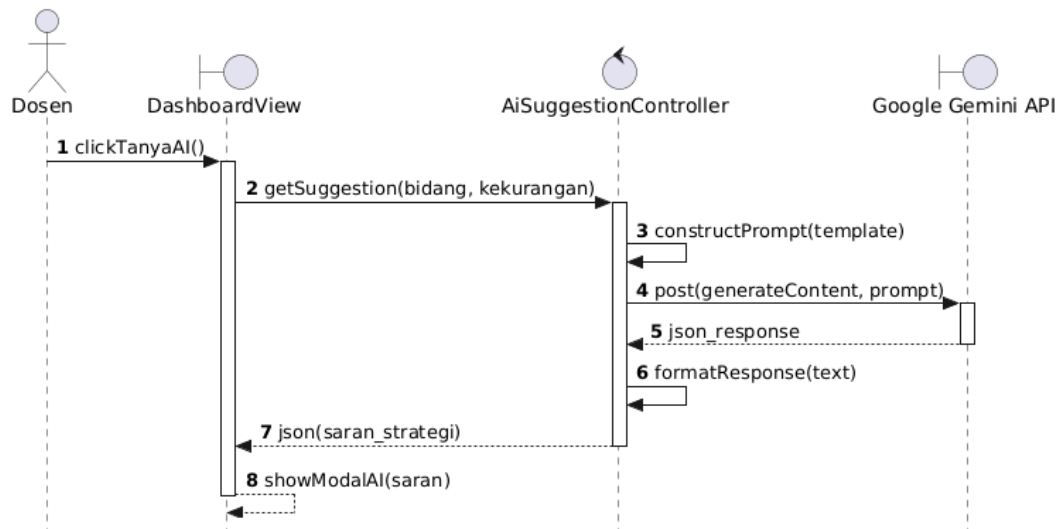
Gambar 22 Sequence Diagram - Crawling Data Scholar



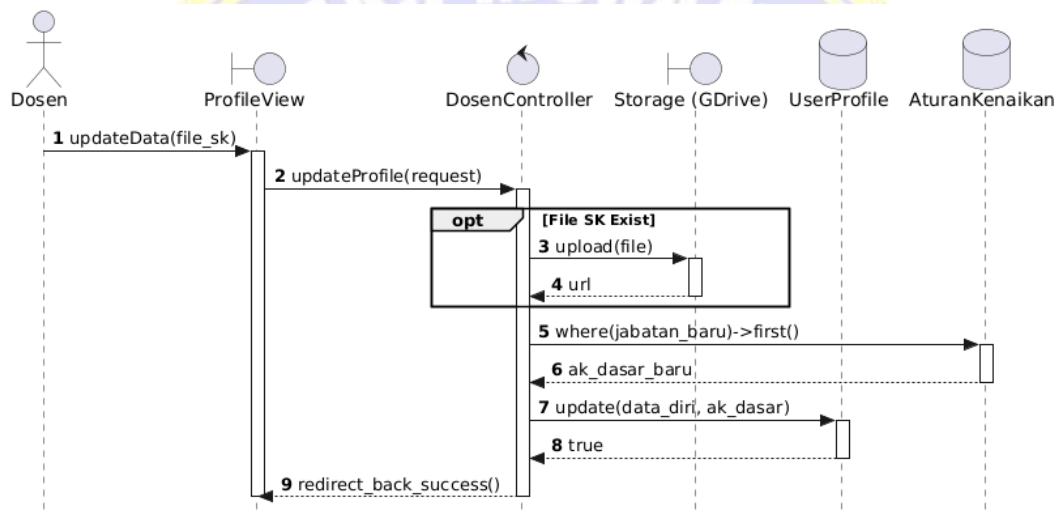
Gambar 23 Sequence Diagram - Simulasi Kenaikan Jabatan



Gambar 24 Sequence Diagram - Konsultasi Strategi AI

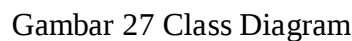


Gambar 25 Sequence Diagram - Kelola Profil & SK Jabatan



Gambar 26 Sequence Diagram - Kelola Periode Penilaian

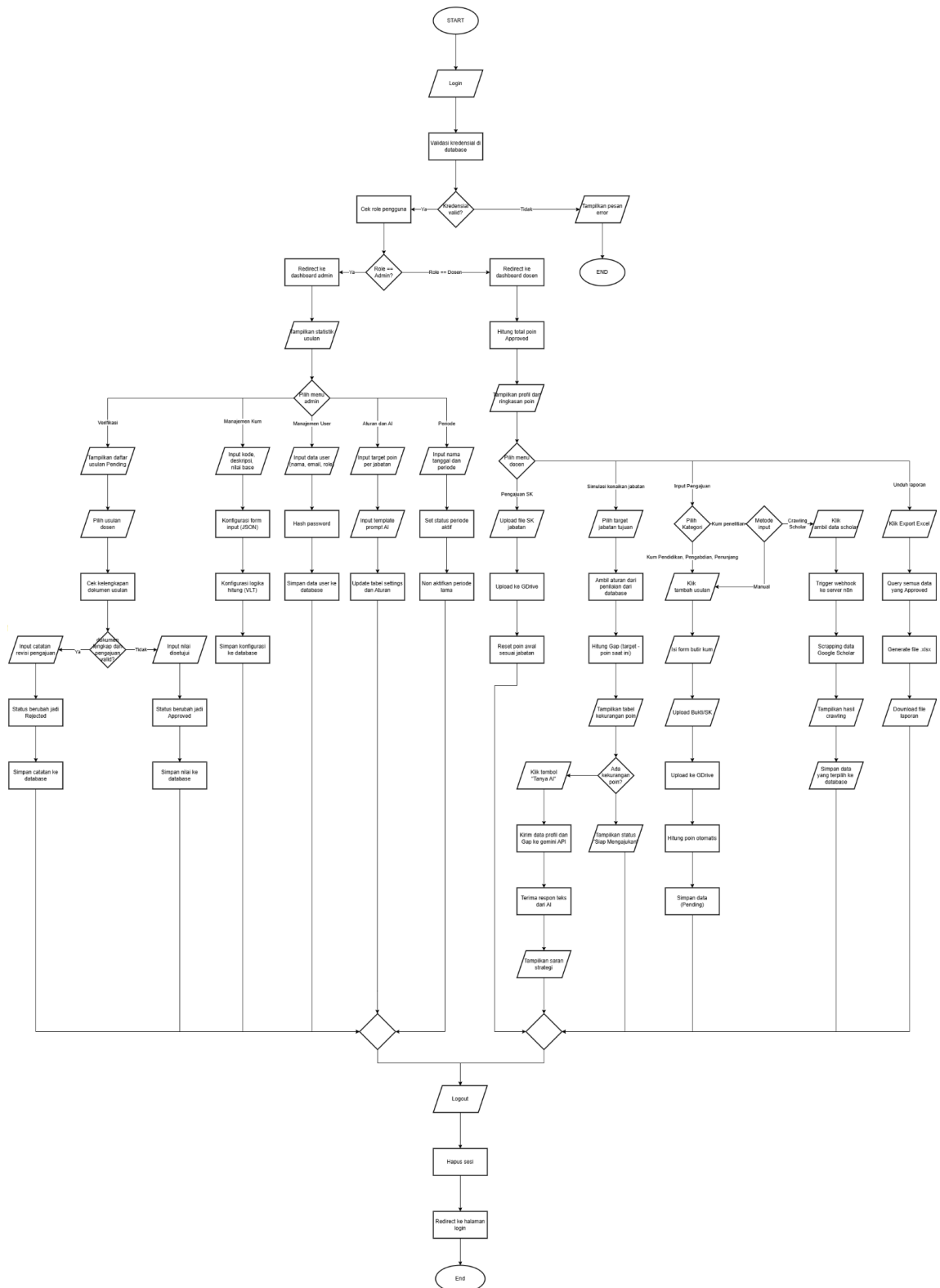
Class Diagram digunakan untuk menggambarkan struktur sistem secara statis, termasuk kelas-kelas utama, atribut, dan hubungan antar kelas. Diagram ini membantu dalam tahap implementasi database dan logika backend.



Flowchart dirancang untuk menggambarkan alur proses sistem secara runtut dan logis berdasarkan cara kerja sistem saat dioperasikan oleh pengguna. Diagram alur ini menjelaskan langkah demi langkah proses

utama mulai dari pengguna mengakses sistem, melakukan autentikasi, hingga proses pengajuan DUPAK dan pengambilan data publikasi secara otomatis melalui teknik crawling. Flowchart juga menggambarkan percabangan kondisi seperti validasi data, pengecekan status pengajuan, serta penanganan jika proses crawling gagal atau data membutuhkan revisi. Dengan adanya flowchart, alur kerja sistem dapat dipahami dengan lebih jelas, baik dari sisi pengguna maupun pengembang, serta dapat dijadikan dasar implementasi proses pada sisi backend sistem.





Gambar 28 Flowchart Sistem

3.2.3.3 Desain Antarmuka

Desain antarmuka (User Interface/UI) dalam sistem pengelolaan DUPAK berbasis web dirancang untuk mendukung pengalaman penggunaan yang mudah, intuitif, dan efisien bagi dosen maupun admin. Pada tahap ini, perancangan antarmuka dilakukan melalui penyusunan wireframe sebagai kerangka visual awal, serta perancangan alur User Experience (UX) untuk memastikan setiap proses sistem dapat dipahami dan digunakan dengan baik sesuai kebutuhan pengguna. Desain yang dihasilkan berfokus pada kejelasan navigasi, konsistensi elemen tampilan, efisiensi interaksi, serta penyajian informasi yang relevan dalam setiap halaman sistem.

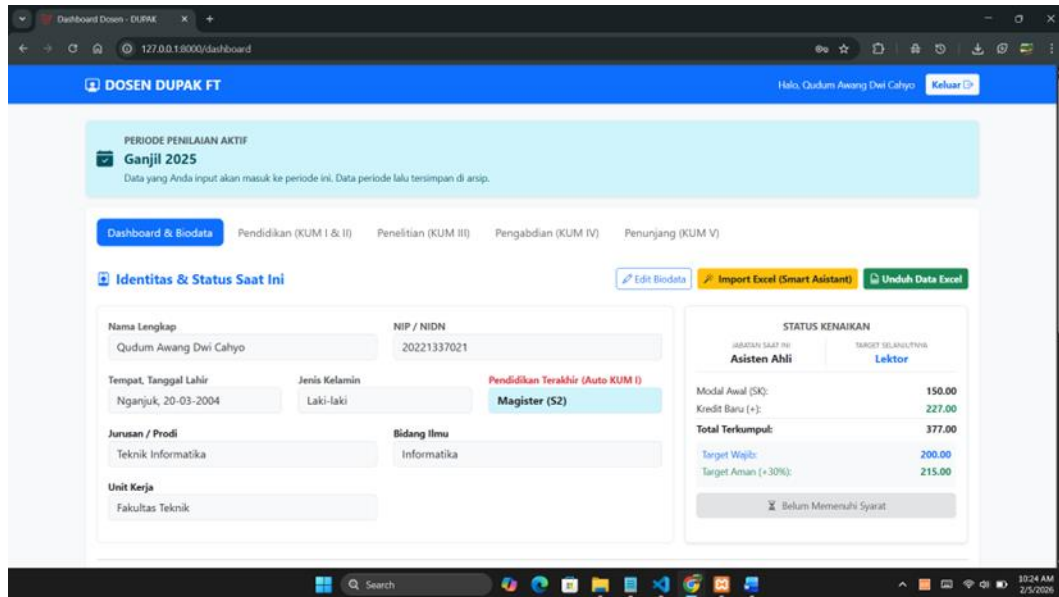
1. Wireframe

Wireframe dibuat sebagai rancangan visual awal dari antarmuka sistem untuk menggambarkan tata letak elemen utama tanpa detail desain seperti warna atau grafis. Perancangan ini bertujuan untuk memetakan struktur halaman agar alur interaksi sistem menjadi jelas sejak awal pengembangan.

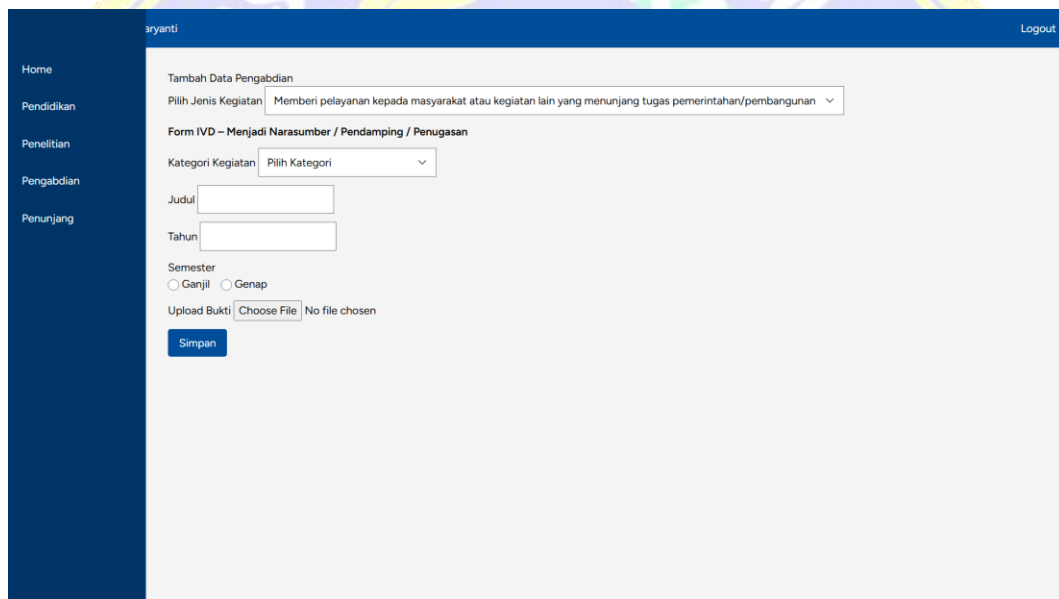
Wireframe yang dirancang mencakup halaman-hal utama pada sistem, antara lain:

- a. Login (akses dosen dan admin),
- b. Dashboard (tampilan ringkasan aktivitas berdasarkan role),
- c. Form pengajuan dan input DUPAK,
- d. Hasil crawling Google Scholar,
- e. Halaman verifikasi pengajuan oleh admin, dan
- f. Halaman monitoring status pengajuan.

Setiap wireframe menampilkan elemen interaktif utama seperti tombol aksi, menu navigasi, tabel data, form input, serta komponen umpan balik sistem (feedback) untuk pengguna.



Gambar 29 Tampilan Dashboard Dosen



Gambar 30 Tampilan Input Pengabdian

Data Pendidikan

+ Tambah Gelar + Tambah Pelaksanaan Pendidikan

Total Angka Kredit KUM A (Pendidikan - Gelar): 150

Daftar Gelar

Gelar	Instansi	Tanggal Kelulusan	Aksi
Magister	Universitas Muhammadiyah Surakarta	2025-11-10	Edit Hapus

Daftar Kegiatan Mengajar

Tahun Ajaran	Semester	Program Studi	Mata Kuliah	SKS	Aksi
2024/2025	Ganjil	Informatika	Metode Penelitian	3	Edit Hapus
2024/2025	Ganjil	Informatika	Pengembangan Web	2	Edit Hapus

Daftar Pembimbing Akademik (IIB)

Nama Mahasiswa	Semester	Tahun Ajaran	Aksi
Sanu	1	2024	Edit Hapus
aku	2	2025	Edit Hapus

Daftar Pembimbing Skripsi / TA (IIC)

Jenis Membimbing	Semester	Tahun Ajaran	Aksi
------------------	----------	--------------	------

Gambar 31 Tampilan Tabel Pendidikan

2. Penjelasan UX dan alur pengguna

Perancangan User Experience (UX) difokuskan pada kemudahan penggunaan, efisiensi alur proses, konsistensi tampilan, serta pengurangan langkah interaksi yang tidak diperlukan. Sistem dirancang agar pengguna dapat menyelesaikan proses pengajuan DUPAK secara cepat, jelas, dan terstruktur, dengan minim kemungkinan kesalahan input atau kebingungan navigasi.

Alur interaksi pengguna dirancang sebagai berikut:

Aktor Dosen

- Melakukan login ke sistem.
- Masuk ke dashboard dosen yang menampilkan menu pengajuan DUPAK dan status progres.
- Menginput data pengajuan secara manual atau menggunakan fitur crawling Google Scholar untuk menarik data publikasi otomatis.
- Meninjau hasil crawling, memilih data yang relevan, lalu menyimpan ke draft DUPAK.
- Mengirim pengajuan DUPAK ke admin untuk diverifikasi.
- Memantau status pengajuan (diterima, revisi, dalam proses).

Aktor Admin

- a. Login ke sistem dan diarahkan ke dashboard admin.
- b. Mengakses daftar pengajuan masuk dari dosen.
- c. Meninjau, memverifikasi, atau memberikan catatan revisi terhadap pengajuan.
- d. Mengelola data hasil crawling serta melakukan approval data publikasi yang masuk.
- e. Sistem memperbarui status pengajuan yang dapat dipantau oleh dosen secara real-time.

Dalam perancangan UX, sistem juga mengimplementasikan:

- a. Navigasi yang konsisten di setiap halaman.
- b. Pemisahan hak akses yang jelas antar peran pengguna.
- c. Notifikasi status proses untuk mengurangi ketidakpastian pengguna.
- d. Tampilan form terstruktur untuk meminimalkan kesalahan input.

Dengan strategi UX dan alur pengguna yang dirancang sistematis, antarmuka sistem tidak hanya berfungsi secara visual, tetapi juga mampu memberikan pengalaman interaksi yang efisien, terarah, dan sesuai kebutuhan operasional pengelolaan DUPAK.

3.2.3.4 Arsitektur Sistem

Arsitektur sistem dirancang untuk mendukung pengembangan artefak berupa sistem pengelolaan DUPAK berbasis web dengan integrasi modul crawling publikasi ilmiah. Arsitektur menggunakan pendekatan tiga lapis (three-tier architecture) yang memisahkan proses presentasi, logika aplikasi, dan pengelolaan data, sehingga sistem lebih mudah dikembangkan, diuji, serta diintegrasikan dengan komponen eksternal seperti crawler Python dan workflow otomatisasi (Gledson et al., 2024).

1. Lapisan Arsitektur

Sistem dibangun berbasis web dengan arsitektur tiga lapis yang memisahkan tampilan, logika proses, dan penyimpanan data, yaitu:

a. Frontend

Lapisan frontend menyediakan antarmuka bagi dua peran utama, yaitu dosen dan admin. Implementasi menggunakan Blade Template Laravel, HTML, CSS responsif, dan komponen JavaScript ringan untuk mendukung interaksi pada fitur dashboard, form komponen DUPAK, serta tampilan

hasil crawling. Frontend dirancang responsif untuk memastikan aksesibilitas dari perangkat berbeda (Tandra et al., 2024).

b. Backend

Backend dikembangkan menggunakan Framework Laravel sebagai pusat logika bisnis. Backend menangani:

- a) autentikasi berbasis role (dosen dan admin),
- b) pengelolaan komponen DUPAK (CRUD komponen A–D),
- c) logika penilaian dan validasi pengajuan,
- d) API internal untuk menerima hasil crawling Python,
- e) endpoint pengiriman data ke workflow otomatisasi n8n,
- f) pengelolaan notifikasi status pengajuan, dan
- g) pemrosesan data dashboard.

(Manurung et al., 2024)

Backend juga menjadi koordinator komunikasi antara sistem web, modul crawling Python, serta layanan workflow otomatisasi.

c. Database

Database menggunakan MySQL yang menyimpan seluruh data pengguna, komponen DUPAK, hasil crawling publikasi, metadata publikasi, riwayat pengajuan, serta log aktivitas. Skema database dirancang relasional dan dioptimalkan untuk query evaluasi, penghitungan kum, dan pencatatan perubahan per iterasi (Khalida & Fadhilla Ramdhania, 2024).

d. Hosting & Infrastruktur

Sistem dihosting pada server berbasis Linux dengan web server Apache, mendukung PHP–Laravel serta kontainerisasi untuk modul Python dan workflow n8n. Infrastruktur dipilih agar mendukung integrasi antar-layanan dan kemudahan deployment menggunakan Docker Compose (Baker et al., 2024).

2. Modul Crawling (Python Scholarly)

Modul crawling dikembangkan terpisah dari backend Laravel menggunakan bahasa Python dan library scholarly, karena library tersebut lebih stabil untuk mengekstraksi profil Google Scholar (Winn et al., 2025).

- a. Mengambil data profil publikasi dosen berdasarkan nama atau Google Scholar ID.
- b. Ekstraksi elemen utama seperti judul, penulis, tahun, sumber, sitasi, dan tautan artikel.
- c. Parsing & normalisasi data agar sesuai skema database.
- d. Error handling, termasuk: perubahan DOM, request limit, CAPTCHA, timeout, dan invalid profile.
- e. Pengiriman data ke backend melalui endpoint REST API atau langsung ke workflow n8n.
- f. Pencatatan log crawling untuk keperluan troubleshooting & evaluasi akurasi ekstraksi (Fahrudin et al., 2025).

Modul crawling dipanggil secara on-demand ketika dosen menekan tombol Ambil Data Publikasi, dan hasilnya ditampilkan sebagai rekomendasi pengisian komponen DUPAK.

3. Workflow Otomatisasi (n8n Automation Workflow)

Sebagai bagian dari arsitektur, sistem memanfaatkan n8n untuk mengotomasi alur pemrosesan data hasil crawling. Workflow n8n berfungsi sebagai middleware otomatisasi antara crawler Python, backend Laravel, dan database (Tebourbi et al., 2025).

- a. menerima payload hasil crawling dari Python,
 - b. melakukan data cleaning dan format normalisasi,
 - c. melakukan pencarian dan pengenalan peringkat jurnal (SJR/Scopus/DOAJ) berdasarkan judul atau DOI,
 - d. menambah metadata rank jurnal sebelum masuk database,
 - e. memicu notifikasi atau update status tertentu secara otomatis,
 - f. mencatat log eksekusi untuk keperluan debugging dan evaluasi.
- (Friebe, 2025)

Integrasi n8n memungkinkan tiap perubahan data bisa diproses secara konsisten tanpa penanganan manual.

4. Fitur Utama Sistem

- a. Manajemen data dosen dan komponen DUPAK (komponen Kum A–E).

- b. Otomasi crawling & ekstraksi publikasi ilmiah untuk mempercepat pengisian komponen Penelitian.
- c. Dashboard monitoring yang menampilkan progres pengajuan dosen dan statistik aktivitas.
- d. Workflow otomatisasi untuk pencarian rank jurnal dan pemrosesan data publikasi.
- e. Notifikasi status pengajuan (diterima, revisi, ditolak).

5. Framework

Komponen Sistem	Teknologi / Framework	Fungsi
Frontend	Laravel Blade, HTML, CSS, JS	Tampilan dan interaksi pengguna
Backend	Laravel (PHP)	Logika bisnis, API, pemrosesan DUPAK
Crawling	Python + Scholarly	Pengambilan & ekstraksi data publikasi
Database	MySQL	Penyimpanan data sistem & hasil crawling
Workflow Automation	n8n	Otomasi pencarian rank jurnal & proses data
Hosting	Server Linux + Docker	Deployment layanan Laravel, Python, dan n8n

3.2.4 Demonstrasi

Tahap demonstrasi dilakukan dengan mengimplementasikan sistem pengelolaan DUPAK berbasis web pada lingkungan terbatas di Fakultas Teknik Universitas Muhammadiyah Surabaya untuk menunjukkan bahwa artefak sistem dapat berjalan sesuai tujuan perancangan. Demonstrasi dilakukan dengan melibatkan pengguna representatif, yaitu dosen sebagai pengusul DUPAK dan admin sebagai pihak verifikator.

Skenario yang diperagakan meliputi proses login ke sistem, input dan pengelolaan data DUPAK, monitoring status pengajuan, serta eksekusi fitur crawling publikasi dari Google Scholar yang kemudian ditampilkan kembali pada dashboard sistem. Selain itu, ditunjukkan juga alur validasi data oleh admin dan penyimpanan hasil crawling ke dalam basis data.

Tahap ini berfokus pada pembuktian bahwa sistem dapat berjalan dan seluruh fitur utama berfungsi sebagaimana dirancang, bukan untuk mengukur kualitas atau tingkat kepuasan pengguna (yang akan dibahas pada tahap evaluasi).

3.2.5 Evaluasi

Evaluasi sistem dilakukan untuk menilai keberhasilan artefak dalam memenuhi tujuan penelitian, yaitu membantu proses pengelolaan DUPAK menjadi lebih otomatis, valid, dan mudah digunakan. Evaluasi juga menjadi bagian integral dari penerapan Design Science Research Methodology (DSRM); pada konteks DSR, evaluasi berfungsi untuk memvalidasi apakah artefak telah memenuhi tujuan dan persyaratan yang ditetapkan pada Objectives and Requirements Echelon serta menjadi dasar keputusan iterasi (Tuunanen et al., 2024).

Secara garis besar, evaluasi dilakukan melalui tiga pendekatan komplementer:

1. Pengujian Fungsional (Black Box Testing)

Pengujian fungsional bertujuan memastikan bahwa setiap fitur memenuhi spesifikasi kebutuhan tanpa melihat implementasi internal. Aspek yang diuji meliputi autentikasi per peran, alur pengajuan DUPAK, manajemen/CRUD komponen DUPAK, pemanggilan dan penyimpanan hasil crawling, serta tampilan dashboard dan riwayat aktivitas. Hasil pengujian dinilai berdasarkan status pass/fail tiap test case dan persentase test case yang lulus (Wulandari et al., 2023).

2. Pengujian Akurasi dan Performa Crawling (Evaluasi Algoritma)

Evaluasi teknis untuk komponen crawling fokus pada:

- a. Kemampuan mengekstrak atribut publikasi (judul, penulis, tahun, sumber, link)
 - b. Kesesuaian/validitas data hasil crawling terhadap sumber asli (ground truth)
 - c. Ketahanan terhadap perubahan struktur halaman dan kegagalan akses.
- Metode pengukuran yang digunakan mencakup metrik precision, recall, dan F1-score untuk menilai kualitas ekstraksi; serta pengukuran waktu respon rata-rata per permintaan crawling dan tingkat keberhasilan (success rate). Ambang keberhasilan awal ditetapkan (misal akurasi/precision $\geq 80\%$ atau

$F1 \geq 0.80$), dan apabila ambang tersebut belum tercapai, komponen algoritma dianggap perlu di-iterasi (Fahrudin et al., 2025).

3. User Testing (Usability / Uji Keterpakaian)

Evaluasi pengguna dilakukan untuk menilai sejauh mana artefak mampu mendukung penyelesaian tugas inti dalam proses pengelolaan DUPAK, seperti melakukan login, menjalankan proses crawling, memeriksa hasil ekstraksi publikasi, mengisi komponen DUPAK, dan mengajukan berkas. Evaluasi dilaksanakan melalui observasi langsung terhadap perilaku pengguna (dosen dan admin) ketika menjalankan tugas-tugas tersebut. Selama proses ini, peneliti mencatat setiap hambatan yang muncul, baik berupa kesalahan interaksi, langkah yang membingungkan, maupun kendala teknis (Manurung et al., 2024). Selama uji coba diklasifikasikan menjadi tiga jenis:

1. Masalah Kritis (Critical/Showstopper)

Masalah yang membuat pengguna tidak dapat menyelesaikan tugas inti, seperti proses crawling gagal total, data tidak tersimpan, atau pengajuan DUPAK tidak dapat dilakukan.

2. Masalah Signifikan (Significant) Hambatan yang mengganggu alur kerja penting (misalnya langkah terlalu membingungkan, validasi tidak muncul saat perlu), menyebabkan kesalahan berulang, atau membuat pengguna kesulitan tetapi tetap bisa menyelesaikan tugas.

3. Masalah Minor (Nuisance/Minor) Masalah kecil bersifat kosmetik atau usulan peningkatan, misalnya ikon kurang jelas atau tata letak kurang rapi.

Target: diperbolehkan (dicatat sebagai saran pengembangan lanjutan).

Evaluasi ini menjadi dasar dalam mekanisme iteratif penelitian Design Science Research (Manurung et al., 2024). Setiap siklus pengembangan diakhiri dengan pemeriksaan terhadap tiga aspek utama: fungsionalitas sistem, akurasi dan performa crawling, serta kualitas pengalaman pengguna berdasarkan hasil observasi. Iterasi diperlukan apabila masih ditemukan masalah kritis atau signifikan, apabila tingkat keberhasilan fungsionalitas belum mencapai 95%, atau apabila performa dan akurasi crawling belum memenuhi batas minimal yang ditetapkan (misalnya $F1\text{-score} \geq 0.80$ dan $\text{success rate} \geq 95\%$). Proses iteratif ini berhenti ketika seluruh indikator telah terpenuhi atau ketika jumlah

iterasi mencapai batas maksimum tiga siklus, sebagaimana diterapkan pada penelitian DSR terdahulu (Jat et al., 2024). Masalah minor yang masih tersisa tidak memicu iterasi lanjutan, tetapi dicatat sebagai rekomendasi pengembangan setelah penelitian selesai.

3.2.6 Komunikasi hasil

Hasil penelitian disajikan dalam beberapa bentuk luaran untuk memastikan penyampaian informasi yang jelas, terstruktur, dan bermanfaat bagi pengembangan penelitian selanjutnya. Luaran tersebut meliputi:

1. Laporan Skripsi, sebagai dokumentasi utama yang memuat seluruh proses penelitian, mulai dari identifikasi masalah, perancangan, implementasi, hingga evaluasi sistem.
2. Presentasi Seminar Proposal dan Sidang Skripsi, digunakan untuk menyampaikan rancangan dan hasil penelitian secara lisan kepada dosen pembimbing dan penguji.
3. Dokumentasi Teknis Sistem, mencakup desain UML, alur sistem, arsitektur, serta hasil implementasi sebagai referensi pengembangan sistem.

Penyajian hasil penelitian ini diharapkan dapat menjadi referensi akademik dan teknis untuk penelitian lanjutan maupun pengembangan sistem serupa di masa mendatang (Gregor & Zwikael, 2024).

3.3 Pengumpulan Data

Metode pengumpulan data dalam penelitian ini dilakukan melalui beberapa tahapan yang disesuaikan dengan pendekatan Design Science Research (DSR). Tujuan dari pengumpulan data ini adalah memperoleh informasi yang relevan baik secara konseptual maupun empiris untuk mendukung proses perancangan artefak sistem pengelolaan DUPAK berbasis web.

1. Studi Literatur

Studi literatur dilakukan untuk memperoleh landasan teoretis terkait sistem informasi DUPAK, metode DSR, teknik web crawling, dan perancangan sistem dengan UML. Sumber data diperoleh dari publikasi ilmiah yang diindeks oleh Google Scholar, meliputi jurnal, prosiding, dan artikel ilmiah yang relevan. Penelitian sebelumnya seperti (Khalida & Fadhilla Ramdhan, 2024)

menjelaskan pentingnya pengambilan keputusan berbasis sistem dalam proses kenaikan jabatan dosen, sehingga menjadi acuan dalam merancang sistem DUPAK yang efisien. Selain itu, penelitian (Anugrah et al., 2025) digunakan sebagai referensi dalam penerapan teknik crawling untuk pengambilan data publikasi secara otomatis. Referensi lain yang turut digunakan antara lain (Figueiredo & García-Peñalvo, 2024) yang mendukung penggunaan DSR dalam konteks pengembangan sistem berbasis iteratif.

2. Observasi

Observasi dilakukan secara langsung di lingkungan Universitas Muhammadiyah Surabaya untuk memahami kondisi eksisting pada proses pengajuan dan penilaian DUPAK. Pengamatan difokuskan pada mekanisme pengumpulan data publikasi dosen, alur validasi, dan kendala administratif yang muncul akibat proses manual. Observasi ini juga membantu dalam mengidentifikasi kebutuhan pengguna dan mendefinisikan fungsi utama sistem yang akan dikembangkan sesuai pendekatan (Gregor & Zwikael, 2024), di mana pemahaman konteks nyata menjadi dasar pengembangan solusi efektif.

3. Wawancara

Wawancara dilakukan terhadap pihak-pihak yang terlibat langsung dalam proses DUPAK, seperti dosen pengusul. Tujuannya untuk memperoleh data primer tentang kebutuhan sistem, hambatan yang dihadapi, serta preferensi fitur yang diharapkan. Pendekatan ini sejalan dengan penelitian (Sulistiyanto et al., 2025) yang menekankan pentingnya pemetaan kebutuhan sumber daya manusia dalam pengembangan sistem pendukung kinerja akademik. Hasil wawancara digunakan untuk menyusun kebutuhan fungsional dan non-fungsional sistem, yang kemudian menjadi dasar perancangan artefak menggunakan model UML.

4. Dokumentasi

Selain tiga teknik di atas, penelitian ini juga menggunakan teknik dokumentasi untuk mengumpulkan data sekunder berupa pedoman DUPAK, peraturan kenaikan jabatan fungsional, serta dokumen administratif universitas.

Data ini berfungsi sebagai acuan validasi sistem agar sesuai dengan ketentuan yang berlaku dan mempermudah proses verifikasi data dosen. Hal ini didukung

oleh pendekatan (Manurung et al., 2024) yang menunjukkan pentingnya pemanfaatan data administratif untuk validasi sistem akademik.

3.4 Rencana Evaluasi

Rencana evaluasi disusun untuk memastikan bahwa artefak yang dibangun benar-benar memenuhi tujuan penelitian, baik dari sisi fungsionalitas, performa teknis, maupun pengalaman pengguna. Rencana ini juga konsisten dengan prinsip DSR, bahwa evaluasi harus memvalidasi pemenuhan tujuan dan persyaratan artefak (Objectives and Requirements Echelon) serta memberikan dasar untuk iterasi desain berikutnya (Tuunanen et al., 2024). Rencana evaluasi ini mengoperasionalkan prinsip evaluasi yang telah dijelaskan pada bagian 3.2.5, sehingga mekanisme pengujian, indikator keberhasilan, dan pemicu iterasi mengikuti struktur evaluasi DSR yang telah ditetapkan sebelumnya.

1. Pengujian Fungsional (Blackbox Testing)

Pengujian fungsional bertujuan untuk memastikan bahwa setiap skenario operasional sistem berjalan sesuai dengan kebutuhan pengguna. Evaluasi dilakukan tanpa melihat struktur kode, tetapi berfokus pada kesesuaian antara input, proses, dan output sistem (Wulandari et al., 2023).

a. Sasaran.

Menjamin bahwa seluruh fungsi utama, login sesuai role, input dan pengelolaan data DUPAK, pemanggilan proses crawling, penyimpanan hasil ekstraksi, serta penampilan dashboard dan Riwayat berjalan sebagaimana dirancang.

b. Metode.

Pengujian dilakukan dengan menyusun daftar test case terperinci yang mencakup langkah uji, input yang diberikan, output yang diharapkan, serta status keberhasilan uji. Setiap fitur diuji pada tiap iterasi pengembangan, sehingga memungkinkan deteksi dini terhadap defect yang muncul pada proses integrasi sistem (Arsa et al., 2024).

c. Indikator Keberhasilan.

Persentase test case yang lulus pengujian. Ambang keberhasilan ditetapkan minimal $\geq 95\%$ test case berstatus pass pada tiap iterasi.

d. Instrumen & Pelaporan.

Hasil pengujian dilaporkan dalam bentuk tabel test case per iterasi, daftar defect, kategori keparahan, dan rekomendasi perbaikan. Dokumentasi ini berfungsi sebagai dasar penentuan prioritas pada siklus iterasi berikutnya (Jat et al., 2024).

2. Evaluasi Kinerja Sistem (Pengujian Crawling & Kinerja Teknis)

Evaluasi kinerja difokuskan pada bagaimana sistem melakukan ekstraksi data publikasi secara otomatis serta bagaimana performa teknis sistem merespons berbagai kondisi akses. Pengujian ini penting karena keakuratan dan kecepatan ekstraksi merupakan inti kontribusi penelitian (Fahrudin et al., 2025).

a. Sasaran.

Menilai efisiensi, stabilitas, dan kualitas hasil ekstraksi publikasi yang diambil dari Google Scholar.

b. Metode.

Pengujian dilakukan dengan mengeksekusi crawling terhadap sampel publikasi yang telah memiliki ground truth untuk dibandingkan. Hasil ekstraksi dihitung menggunakan metrik precision, recall, dan F1-score. Selain itu dihitung waktu rata-rata per request serta success rate saat sistem menghadapi skenario perubahan struktur halaman (DOM), timeout, CAPTCHA, atau data tidak ditemukan (Fahrudin et al., 2025).

Indikator & Ambang Keberhasilan.

a) Precision ≥ 0.80 .

b) Recall ≥ 0.80 .

c) F1-score ≥ 0.80 .

d) Waktu crawling $\leq X$ detik per request (nilai X ditentukan pada uji pilot).

e) Success rate $\geq 95\%$ pada kondisi normal.

(Fahrudin et al., 2025)

c. Instrumen.

Skip pengujian otomatis, log aktivitas crawling, serta spreadsheet berisi perbandingan hasil ekstraksi dengan ground truth termasuk true positive, false positive, dan false negative (Le Clercq, 2024).

3. User Testing (Uji Pengguna)

Uji pengguna dilakukan untuk memperoleh gambaran bagaimana pengguna berinteraksi dengan sistem dalam kondisi nyata. Fokus pengujian meliputi kelancaran alur penggunaan, efektivitas proses otomatisasi crawling, serta kemampuan sistem mendukung penyelesaian tugas inti pengelolaan DUPAK. Pengguna diminta menjalankan langkah-langkah sesuai skenario penggunaan sebenarnya, sementara peneliti mengamati perilaku mereka, kendala yang muncul, dan umpan balik spontan selama proses berlangsung (Manurung et al., 2024).

a. Sasaran.

Uji pengguna bertujuan menilai seberapa baik sistem mendukung penyelesaian tugas inti seperti login, pengambilan data publikasi melalui crawling, verifikasi hasil ekstraksi, pengisian komponen DUPAK, dan pengajuan. Penilaian difokuskan pada keberhasilan penyelesaian tugas serta hambatan-hambatan yang muncul selama interaksi.

b. Partisipan.

Pengujian melibatkan admin dan dosen sebagai representasi pengguna akhir. Setiap partisipan diminta menggunakan sistem sesuai perannya untuk memberikan gambaran yang realistis mengenai alur penggunaan dan potensi hambatan yang muncul di lapangan. (Gandhi et al., 2024).

c. Metode.

Seluruh proses dilakukan melalui observasi langsung tanpa instrumen kuesioner. Peneliti mencatat langkah-langkah yang ditempuh pengguna, kesalahan yang terjadi, alur kerja yang membingungkan, serta feedback langsung yang diberikan. Setiap temuan kemudian dikategorikan sebagai:

- a) Masalah kritis, yaitu hambatan yang menyebabkan tugas inti tidak dapat diselesaikan.

- b) Masalah signifikan, yaitu hambatan yang mengganggu alur kerja namun tugas masih dapat diselesaikan.
- c) Masalah minor, yaitu gangguan kecil atau saran peningkatan yang tidak memengaruhi keberhasilan tugas.

Sistem dinyatakan berhasil apabila tidak ditemukan masalah kritis maupun signifikan pada akhir pengujian (Gandhi et al., 2024). Pendekatan ini konsisten dengan metode evaluasi pada bagian 3.2.5, yaitu menilai usability melalui observasi langsung tanpa instrumen kuesioner formal seperti SUS atau UMUX-Lite.

d. Instrumen.

Instrumen evaluasi terdiri atas lembar observasi, skenario tugas yang harus diselesaikan pengguna, catatan umpan balik spontan, dan dokumentasi berupa tangkapan layar atau rekaman error jika diperlukan. Pendekatan ini memungkinkan peneliti menangkap dinamika penggunaan secara langsung tanpa bergantung pada instrumen kuesioner formal (Gandhi et al., 2024).

4. Iterasi dan Keputusan Perbaikan

Mekanisme iterasi dirancang agar evaluasi pengguna tidak hanya menghasilkan temuan, tetapi juga memicu perbaikan berkelanjutan sesuai prinsip Design Science Research. Setiap siklus pengembangan diakhiri dengan evaluasi atas fungsionalitas sistem, akurasi dan performa crawling, serta kualitas interaksi pengguna berdasarkan hasil observasi (Tuunanen et al., 2024).

a. Batas Iterasi.

Penelitian ini menetapkan maksimal tiga siklus iterasi untuk menjaga ruang lingkup dan durasi penelitian tetap terkendali, sejalan dengan praktik umum penelitian DSR.

b. Aturan Pemicu Iterasi.

Iterasi dilakukan apabila pada evaluasi suatu siklus masih ditemukan salah satu dari kondisi berikut:

- a) Terdapat masalah kritis atau signifikan pada hasil uji pengguna.
- b) Tingkat keberhasilan fungsionalitas belum mencapai 95%.
- c) Akurasi crawling (precision/recall/F1) berada di bawah target minimal.

- d) Performa crawling belum memenuhi ambang batas waktu dan keberhasilan yang ditetapkan.
- e) Masalah minor tidak memicu iterasi dan hanya dicatat sebagai rekomendasi perbaikan, sesuai ketentuan pada bagian evaluasi (3.2.5).

Dengan demikian, keputusan iterasi tidak bergantung pada persepsi subjektif pengguna, tetapi pada indikator objektif yang berasal dari tujuan desain sistem.

c. Proses Iterasi.

Setiap iterasi berfokus pada perbaikan temuan prioritas, seperti penghapusan bug kritis, penyempurnaan antarmuka untuk mengurangi kebingungan pengguna, optimasi algoritma crawling, dan peningkatan performa sistem. Setelah perbaikan selesai diterapkan, pengujian diulang menggunakan prosedur yang sama untuk memastikan peningkatan dapat divalidasi secara empiris (Tuunanen et al., 2024).

d. Kondisi Berhenti.

Iterasi dihentikan apabila seluruh indikator keberhasilan telah terpenuhi atau jumlah iterasi mencapai batas tiga siklus. Masalah minor yang masih tersisa tetap didokumentasikan sebagai rekomendasi pengembangan pasca-penelitian, karena tidak memengaruhi penyelesaian tugas inti maupun keberhasilan evaluasi (Tuunanen et al., 2024).

3.5 Requirement Traceability Matrix (RTM)

Untuk memastikan bahwa seluruh kebutuhan sistem yang telah didefinisikan pada tahap awal benar-benar dirancang dan diuji secara menyeluruh, penelitian ini menggunakan *Requirement Traceability Matrix* (RTM). RTM berfungsi sebagai alat kendali mutu untuk melacak konsistensi antara kebutuhan fungsional pengguna (*User Requirements*), pemodelan sistem (*Use Case*), hingga skenario pengujian fungsional (*Test Case*) yang dilaksanakan pada tahap demonstrasi dan evaluasi artefak. Matriks pelacakan ini menjamin tidak ada satupun fitur yang terlewat dalam proses perancangan maupun pengujian.

Table 13 Requirement Traceability Matrix (RTM)

ID Kebutuhan (REQ)	Deskripsi Kebutuhan Fungsional	Kode Use Case	Modul / Fitur Sistem	ID Skenario Uji
REQ-01	Sistem menyediakan fitur autentikasi bagi Dosen dan Admin sesuai dengan tingkat hak akses.	UC-001	Autentikasi	UJ-01
REQ-02	Dosen dapat mengelola biodata diri dan mengunggah dokumen SK Jabatan Fungsional terbaru.	UC-011	Profil Dosen	UJ-01
REQ-03	Sistem mampu mengekstraksi data publikasi secara otomatis dari pangkalan data Google Scholar.	UC-008	Integrasi Crawling (n8n)	UJ-04
REQ-04	Dosen dapat menambah, mengubah, dan menghapus draf pengajuan DUPAK.	UC-007	Formulir DUPAK	UJ-05
REQ-05	Dosen dapat melihat visualisasi statistik dan ringkasan poin yang telah diverifikasi.	UC-006	Dashboard Dosen	UJ-01
REQ-06	Sistem dapat melakukan simulasi kalkulasi selisih poin berdasarkan target jabatan fungsional.	UC-009	Simulasi Kenaikan	UJ-02
REQ-07	Sistem menyediakan fitur Asisten AI untuk merumuskan saran strategi pemenuhan angka kredit.	UC-010	Integrasi API Gemini	UJ-03
REQ-08	Dosen dapat mengunduh rekapitulasi pengajuan yang disetujui dalam format Excel.	UC-006	Ekspor Laporan	UJ-06
REQ-09	Admin dapat memverifikasi berkas, menyetujui, menolak, atau memberikan catatan revisi.	UC-002	Verifikasi Admin	AD-03
REQ-10	Admin dapat merancang butir kegiatan baru dan menyusun logika kalkulasi (logic_config) secara visual.	UC-003	Manajemen KUM & VLT	AD-02
REQ-11	Admin dapat menambah dan mengatur data pengguna sistem (Dosen/Admin).	UC-004	Manajemen Pengguna	AD-01
REQ-12	Admin dapat menyesuaikan parameter PO PAK (target poin) dan konfigurasi prompt AI.	UC-005	Pengaturan Aturan	AD-01
REQ-13	Admin berhak membuka, menutup, dan mengelola linimasa periode penilaian DUPAK aktif.	UC-012	Manajemen Periode	AD-01