

Pengembangan Aplikasi Pengelolaan Skripsi di Perguruan Tinggi Berbasis Web: Studi Kasus Politeknik Negeri Jakarta

Anggi Mardiyono ^{1*}, Fachroni Arbi Murad ², Bagas Rizkiyanto ¹

1. Program Studi Teknik Informatika, Politeknik Negeri Jakarta, Indonesia

2. Program Studi Teknik Komputer dan Jaringan, Politeknik Negeri Jakarta, Indonesia

* Korespondensi: anggi.mardiyono@tik.pnj.ac.id

Sitasi: A. Mardiyono, F. A. Murad, and B. Rizkiyanto. Pengembangan Aplikasi Pengelolaan Skripsi di Perguruan Tinggi Berbasis Web: Studi Kasus Politeknik Negeri Jakarta. Jurnal Teknologi Informasi Dan Multimedia, vol. 8, no. 3, pp. 681-695, 2026.
<https://doi.org/10.35746/jtim.v8i3.941>

Diterima: 30-12-2025

Direvisi: 28-03-2026

Disetujui: 17-07-2026



Copyright: © 2026 oleh para penulis. Karya ini dilisensikan di bawah Creative Commons Attribution-ShareAlike 4.0 International License. (<https://creativecommons.org/licenses/by-sa/4.0/>).

Abstract. This paper discusses the design and development of a web-based final project (thesis) management application using the Laravel framework for the Informatics and Computer Engineering Department at Jakarta State Polytechnic. The manual thesis management process causes time inefficiency and document issues, so this application supports all stages of student final project submissions, from title proposals, supervisor assignments, logbook recording, proposal seminars, thesis examinations, revisions, to the submission form for equipment handover or final project results from students. The application is designed for 7 user types (admin, committee chair, committee, chief examiner, examiner, supervisor, and students) with main features such as dashboard monitoring, student/lecturer data management, and submissions for various thesis processes. Application development uses the Prototyping method. Application testing employs Blackbox testing (positive and negative cases) and System Usability Scale (SUS). The application has been successfully implemented and tested with users, yielding SUS evaluation results with an average score of 84.375 (grade A, excellent). The obtained score indicates that the application is proven effective in minimizing paper usage and facilitating digital monitoring of the thesis process.

Keywords: thesis management application, laravel framework, information system, system usability scale, prototyping method

Abstrak. Paper ini membahas rancang bangun aplikasi pengelolaan tugas akhir (skripsi) berbasis web menggunakan framework Laravel untuk Jurusan Teknik Informatika dan Komputer Politeknik Negeri Jakarta. Proses pengelolaan skripsi yang masih manual menyebabkan inefisiensi waktu dan dokumen, sehingga aplikasi ini mendukung seluruh tahapan dari pengajuan pengajuan tugas akhir mahasiswa mulai dari pengajuan judul, pengajuan pembimbing, pencatatan logbook, seminar proposal, sidang skripsi, revisi, hingga penyerahan alat atau hasil dari tugas akhir. Aplikasi ini dirancang untuk 7 jenis pengguna (admin, ketua komite, komite, ketua penguji, penguji, pembimbing, dan mahasiswa) dengan fitur utama seperti dashboard monitoring, pengelolaan data mahasiswa/dosen, dan pengajuan berbagai proses skripsi. Pengembangan aplikasi ini menggunakan metode Prototyping. Pengujian aplikasi menggunakan pengujian Blackbox (*positive* dan *negative case*) dan System Usability Scale (SUS). Aplikasi ini telah berhasil diterapkan dan diujikan ke pengguna dengan hasil penilaian menggunakan SUS dengan skor rata-rata 84,375 (grade A, excellent). Skor yang diperoleh menunjukkan aplikasi ini terbukti efektif meminimalisir penggunaan kertas dan memudahkan monitoring proses skripsi secara digital.

Kata kunci: aplikasi skripsi; framework laravel; sistem informasi; system usability scale; metode prototyping

1. Pendahuluan

Mahasiswa Politeknik Negeri Jakarta (PNJ) diwajibkan untuk membuat skripsi sebagai syarat yang diperlukan untuk memperoleh gelar diploma politeknik. Skripsi adalah karya ilmiah yang dibuat oleh mahasiswa yang mencakup penelitian atau kajian mendalam tentang suatu topik tertentu dalam bidang tertentu [1]. Pelaksanaan skripsi tersebut juga terjadi pada jurusan Teknik Informatika dan Komputer (JTIK). Dalam pelaksanaan skripsi tersebut, tentunya ada beberapa tahap atau alur yang harus dilalui, mulai dari pengajuan judul dan dosen pembimbing, hingga pelaksanaan sidang akhir.

Pelaksanaan skripsi pada JTIK masih dilakukan secara manual, yaitu masih menggunakan Google Form untuk pengumpulan data, seperti pengajuan judul, pengajuan pembimbing, bahkan sampai pengajuan sidang. Sebelum melakukan sidang, tentunya diwajibkan mengikuti sejumlah bimbingan yang sudah ditentukan pada persyaratan awal, sedangkan pengecekan log book dan proposal pada saat bimbingan masih dilakukan secara manual, salah satunya adalah mengumpulkan melalui Google Drive. Dengan demikian, pengelolaan kegiatan skripsi pada JTIK PNJ dari awal hingga akhir masih kurang efektif dan efisien.

Permasalahan yang serupa juga terjadi pada jurusan Teknik Informatika Universitas Negeri Gorontalo (UNG). Permasalahan yang terjadi adalah pengelolaan data skripsi dan kerja praktik lapangan yang masih dilakukan secara manual. Hal ini berdampak pada pihak manajemen jurusan dan program studi yang kesulitan untuk memonitoring data usulan topik, tahapan, progres bimbingan, serta riwayat nilai ujian skripsi dan kerja praktik [2]. Pengembangan aplikasi untuk mendukung layanan akademik telah banyak dikembangkan salah satunya untuk pengelolaan tugas telah dikembangkan salah satunya untuk pengelolaan tugas [3]. Perlu adanya pengembangan aplikasi yang dapat mengelola tugas akhir mahasiswa agar proses ajuan proposal dan persetujuan topik dapat dimonitor oleh dosen dan mahasiswa sendiri.

Sebagai solusi yang dapat dipilih adalah pengembangan aplikasi berbasis web untuk membangun sistem pengajuan yang dapat diakses oleh banyak pengguna melalui internet. Aplikasi yang dikembangkan dapat memanfaatkan beberapa *framework* yang sudah cukup banyak dikembangkan oleh pengembang aplikasi salah satunya adalah Laravel. *Framework* ini telah dikembangkan di beberapa aplikasi akademik salah satunya aplikasi yang dikembangkan untuk keperluan informasi kerja bagi siswa [4]. Aplikasi lain yang kemudian menggunakan *framework* serupa adalah aplikasi Student Get Student yang telah dikembangkan di salah satu perguruan tinggi nasional [5].

Pengembangan aplikasi ini akan memperbaiki pelayanan akademik di lingkungan JTIK dalam rangka peningkatan layanan untuk mencapai sasaran kinerja Badan Layanan Umum di kampus PNJ. Tahapan pengembangan yang dilakukan menggunakan pendekatan metode Prototyping. Sistem yang dikembangkan akan diuji dengan metode System Usability Scale (SUS).

Penelitian ini tidak hanya berfokus pada digitalisasi administrasi skripsi sebagaimana studi sebelumnya, tetapi juga memberikan kontribusi pada pengembangan arsitektur sistem serta interoperabilitas data akademik. Kontribusi utama penelitian ini adalah perancangan arsitektur sistem berbasis modular *service (microservice-ready)* yang memungkinkan integrasi dengan sistem akademik internal dan Implementasi mekanisme *workflow tracking* berbasis *state machine* untuk memonitor progres skripsi secara *real-time*.

2. Bahan dan Metode

Pengembangan aplikasi pengelolaan tugas akhir berbasis web pada studi kasus Politeknik Negeri Jakarta memerlukan pendekatan sistematis untuk memastikan keberhasilan implementasi. Tahapan dalam penelitian ini meliputi analisis masalah, studi literatur, pengumpulan data, metode pengembangan sistem, perancangan sistem, pemilihan teknologi, dan prosedur pengujian.

2.1. Analisis Masalah

Proses kegiatan mahasiswa dalam pelaksanaan tugas akhir (skripsi) mulai dari pengajuan judul dan pemilihan dosen pembimbing masih dilakukan secara manual melalui Google Form, begitu pula dosen yang memeriksanya secara manual berdasarkan data dari Google Form tersebut. Proses yang saat ini berjalan memiliki banyak kendala diantaranya seringnya keterlambatan mahasiswa dalam mendapatkan informasi pembimbing dan topik yang telah disetujui. Dari sisi dosen, terdapat kendala yang sering dialami yaitu seringnya banyak data *logbook* mahasiswa yang sering terselip karena berkas yang dikirimkan sering terlambat diunggah oleh mahasiswa karena mahasiswa harus mencetak data *logbook* tersebut melalui formulir kertas yang disediakan sesuai dengan format yang diberikan oleh panitia skripsi (komite).

Beberapa kendala yang dialami baik dari sisi mahasiswa dan dosen memerlukan dibangunnya sistem pengelolaan berbasis web yang dapat memberikan informasi dan pengelolaan data yang lebih terstruktur khususnya di Jurusan Teknik Informatika dan Komputer PNJ. Sistem yang dikembangkan akan dapat memonitoring kegiatan skripsi dengan baik. Dengan adanya sistem ini proses pengajuan topik, dosen pembimbing, pengumpulan laporan, dan penilaian semuanya dapat dilakukan dalam satu ruang lingkup aplikasi ini.

2.2. Studi Literatur

Tahap ini dilakukan untuk mengumpulkan sumber referensi dari jurnal, artikel, skripsi, dan sumber lainnya yang berkaitan dan dapat mendukung jalannya penelitian. Beberapa penelitian terkait telah dikumpulkan beberapa diantaranya adalah penelitian yang dilakukan oleh [2], [4], [6], [7], dan [8]. Penelitian-penelitian tersebut memberikan wawasan dan menjadi dasar pemilihan teknologi yang dapat diimplementasikan pada aplikasi berbasis web. Tabel 1 merupakan penelitian-penelitian terkait sebelumnya terkait dengan sistem yang dikembangkan di dalam penelitian ini.

Tabel 1. Komparasi Penelitian Terkait

No	Penelitian	Fitur Utama	Arsitektur	Interoperabilitas	Evaluasi	Kelemahan
1	Kasim et al. (2020)[2]	Manajemen skripsi web	Monolithic	Tidak ada	Blackbox	Tidak <i>scalable</i>
2	Kainde (2022)[4]	Sistem informasi web Laravel	Monolithic	Tidak ada	Usability	Tidak mendukung integrasi
3	Saputra et al. (2021)[5]	Aplikasi akademik	Monolithic	Terbatas	SUS	Fokus UI saja
4	Penelitian ini	Workflow skripsi end-to-end	Modular / Microservice-ready	REST API (SIKAD-ready)	SUS + Security + Reliability	-

Berdasarkan tabel komparasi, penelitian ini memiliki kebaruan pada aspek arsitektur sistem yang lebih fleksibel dan *interoperable* dibandingkan penelitian sebelumnya yang masih menggunakan pendekatan Monolithic tanpa dukungan integrasi sistem.

2.3. Pengumpulan Data

Data yang digunakan dalam penelitian ini berasal dari dua sumber utama, yaitu data primer dan data pendukung. Data primer diperoleh dari bagian akademik Jurusan Teknik Informatika dan Komputer (TIK) Politeknik Negeri Jakarta yang mencakup dokumen terkait prosedur pelaksanaan skripsi, alur pengajuan, serta berbagai kebutuhan administrasi akademik. Sementara itu, data pendukung dihimpun melalui wawancara semi-terstruktur dengan pihak-pihak yang terlibat secara langsung dalam penyelenggaraan kegiatan skripsi.

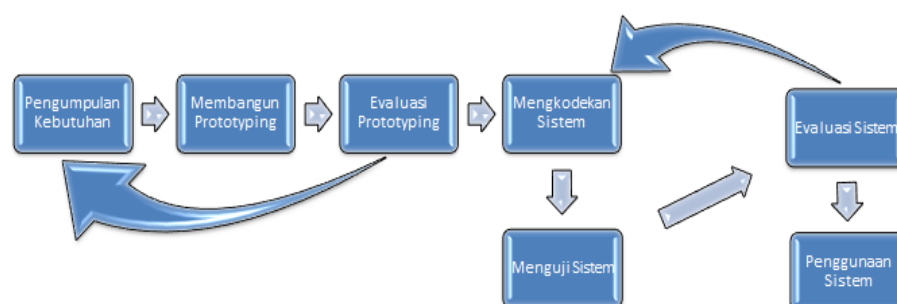
Kegiatan wawancara melibatkan sebanyak 12 partisipan yang terdiri atas lima mahasiswa tingkat akhir, empat dosen pembimbing, dua anggota komite skripsi, dan satu ketua komite skripsi. Setiap wawancara berlangsung antara 30 hingga 45 menit dan dilaksanakan baik secara tatap muka maupun melalui media daring selama periode Januari sampai Februari 2025.

Fokus wawancara diarahkan pada penggalian informasi mengenai permasalahan yang muncul dalam proses pengajuan skripsi, mekanisme pemantauan kegiatan bimbingan, kendala administratif yang sering terjadi, serta kebutuhan pengguna terhadap sistem yang akan dikembangkan. Informasi yang diperoleh kemudian dianalisis untuk mengidentifikasi kebutuhan sistem secara lebih komprehensif.

Untuk menjamin keandalan dan kredibilitas data, penelitian ini menerapkan metode triangulasi sumber dengan membandingkan hasil wawancara dari berbagai kelompok responden terhadap dokumen Standard Operating Procedure (SOP) pelaksanaan skripsi yang berlaku di lingkungan TIK PNJ. Selain itu, dilakukan pula proses *member checking* dengan meminta beberapa responden utama meninjau kembali hasil interpretasi peneliti guna memastikan bahwa informasi yang diperoleh telah sesuai dengan kondisi yang sebenarnya. Seluruh data yang terkumpul selanjutnya digunakan sebagai landasan dalam merumuskan kebutuhan fungsional maupun nonfungsional sistem sehingga solusi yang dikembangkan dapat mengakomodasi kebutuhan pengguna secara optimal.

2.4. Metode Pengembangan Sistem

Sistem yang dikembangkan menggunakan Metode Prototyping. Metode Prototyping adalah pendekatan dalam pengembangan perangkat lunak dengan menciptakan model awal (*prototype*) terlebih dahulu oleh pengembang berdasarkan aplikasi yang akan dibangun. Prototype ini akan berfungsi untuk mendapatkan umpan balik dari pengguna agar pengembang dapat memahami dan mengidentifikasi kebutuhan pengguna sebenarnya. Tujuan utama metode ini adalah untuk memahami kebutuhan pengguna dengan lebih baik dan mendapatkan umpan balik dengan cepat, sehingga perbaikan atau perubahan dapat dilakukan sebelum membuat sistem secara keseluruhan. Dengan demikian, tentunya metode ini dapat mengurangi risiko kesalahan desain dan kegagalan dalam masa penerapan produk [9]. Metode ini memiliki beberapa tahapan yang terlihat pada Gambar 1.



Gambar 1. Metode Prototyping

Berikut ini merupakan tahapan dari Metode Prototyping [10]:

1. Pengumpulan Kebutuhan

Tahap pertama adalah mengidentifikasi dan memahami kebutuhan pengguna. Tahap ini biasanya dilakukan melalui wawancara, penelitian, dan diskusi tim, sehingga tim pengembang harus mendapatkan informasi yang tepat berdasarkan kebutuhan pengguna sebelum membangun sistem.

2. Membangun *Prototyping*

Setelah mengetahui kebutuhan yang ada, langkah berikutnya adalah merancang prototype yang merupakan Gambaran kasar prototype yang ingin dibuat.

3. Evaluasi *Protoyping*

Tahap evaluasi ini dilakukan oleh pengguna untuk memastikan prototype yang sudah dibangun sebelumnya sudah sesuai dengan kebutuhan.

4. Mengkodekan Sistem

Pada tahap ini dimulai pembuatan sistem berdasarkan prototype yang telah disepakati sebelumnya.

5. Menguji Sistem

Setelah aplikasi berhasil dibuat, tentunya harus melalui tahap pengujian terlebih dahulu sebelum digunakan.

2.5. Pemilihan Teknologi

Teknologi atau *framework* yang digunakan untuk mengembangkan aplikasi berbasis web pada penelitian ini adalah Laravel. Pemilihan ini didasari oleh beberapa hal diantaranya adalah Keamanan dan Manajemen Otentikasi [11], Efisiensi Pengembangan dan Skalabilitas [12], Komunitas Besar dan Dukungan Ekosistem [13], Optimal untuk Microservices dan Arsitektur Modern [14], dan Efisiensi Biaya dan Produktivitas [15].

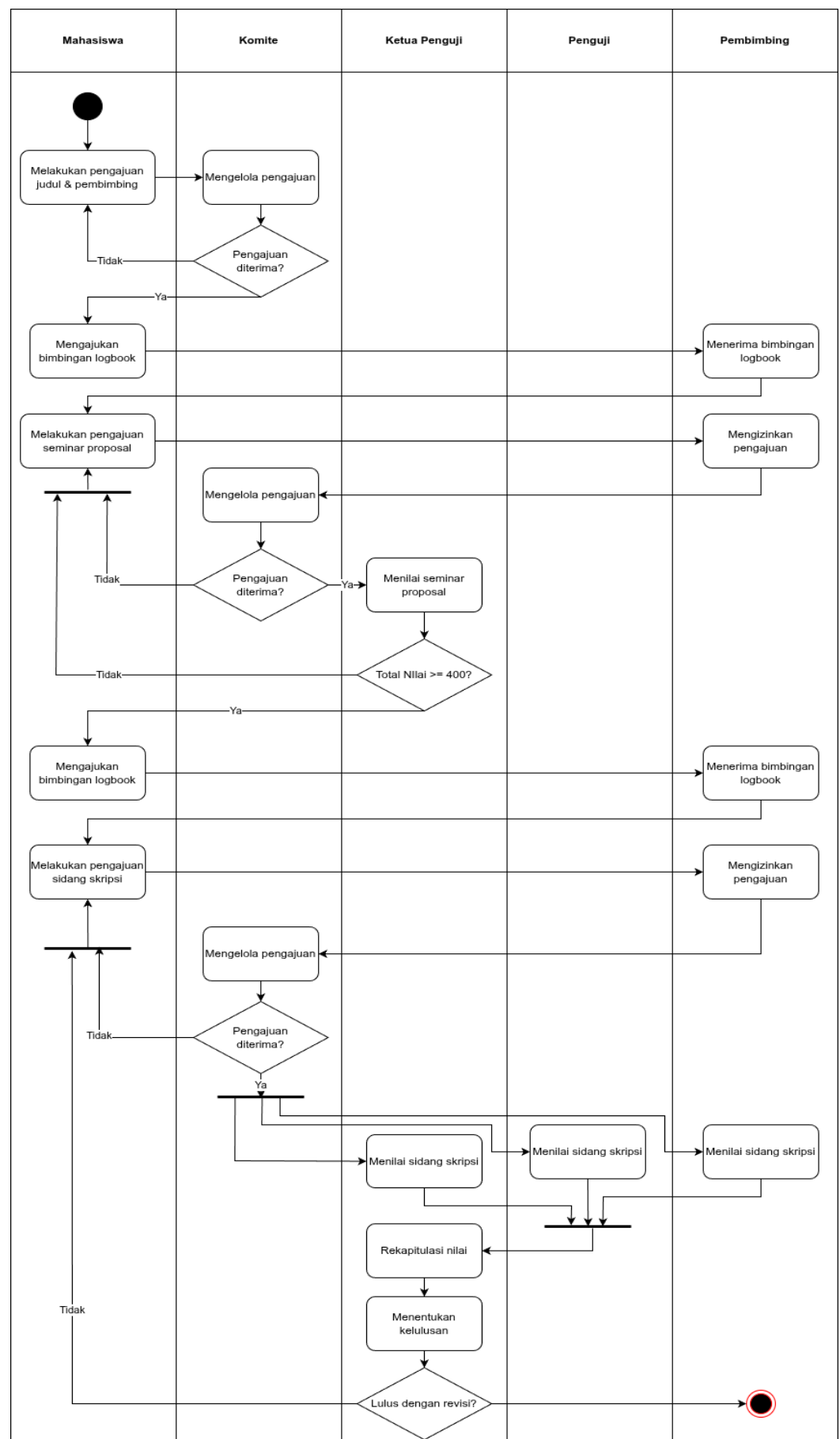
2.6. Prosedur Pengujian

Sistem yang dikembangkan dan akan diterapkan untuk pengguna terlebih dahulu akan diuji. Prosedur pengujian sistem menggunakan pengujian Blackbox dan System Usability Scale (SUS). Penggunaan metode Blackbox ini digunakan untuk menguji fungsional dari sistem. Metode ini dipilih karena masih efektif untuk menguji kebutuhan fungsional dari sistem [16]. Metode ini memiliki kemudahan dalam menganalisis kesalahan pada antarmuka aplikasi dan ketidak konsistenan input data [17]. Pengujian SUS digunakan untuk melihat seberapa mudah aplikasi digunakan oleh pengguna. Pengujian ini telah dapat mengukur kemudahan penggunaan aplikasi pada beberapa penelitian [17], [18], [19], [20].

3. Hasil dan Pembahasan

3.1. Pengumpulan Kebutuhan

Analisis kebutuhan sistem diperoleh dengan wawancara pada masing-masing pihak yang terkait, yaitu komite, ketua penguji, dosen penguji, dan dosen pembimbing. Analisis ini juga dilakukan dengan melihat *Standard Operating Procedure* (SOP) pelaksanaan skripsi di Jurusan Teknik Informatika dan Komputer Politeknik Negeri Jakarta (JTIK PNJ). Tahap ini juga menghasilkan proses bisnis dari kegiatan skripsi yang saat ini berjalan (Gambar 2).



Gambar 2. Proses bisnis pengajuan judul dan dosen pembimbing s.d. pelaksanaan sidang skripsi.

1. Kebutuhan Fungsional

Kebutuhan ini merupakan Gambaran yang dapat dilakukan oleh aplikasi pengelolaan skripsi yang dijabarkan pada Tabel 2.

Tabel 2. Kebutuhan Fungsional

<i>Role</i>	<i>Kemampuan pada aplikasi</i>
Admin	1. Mengelola konten dashboard.
	2. Mengelola user (komite, dosen pembimbing, dosen penguji, mahasiswa).
	3. Monitoring pelaksanaan sidang.
	4. Mengelola pengajuan judul, sempro, sidang skripsi, dan serah terima alat.
	5. Monitoring pengajuan revisi.
Ketua Komite	1. Mengelola revisi mahasiswa yang sudah diperiksa oleh pembimbing dan penguji.
Komite	1. Mengelola konten dashboard.
	2. Mengelola pengajuan judul, sempro, sidang skripsi, dan serah terima alat.
Ketua Penguji	1. Memberi nilai sempro mahasiswa teruji.
	2. Rekapitulasi nilai total sidang skripsi teruji.
	3. Menentukan kelulusan teruji.
	4. Memberikan revisi kepada teruji.
Dosen Penguji	1. Melihat konten dashboard
	2. Monitoring sempro dan sidang skripsi mahasiswa teruji.
	3. Memberikan nilai sidang skripsi mahasiswa teruji.
	4. Memeriksa perbaikan revisi dari teruji.
Dosen Pembimbing	1. Melihat konten dashboard
	2. Mengelola pengajuan logbook mahasiswa.
	3. Mengelola pengajuan sidang sempro mahasiswa bimbingan.
	4. Memberikan nilai sidang skripsi mahasiswa teruji.
	5. Menerima perbaikan revisi mahasiswa bimbingan.
Mahasiswa	1. Melihat konten dashboard.
	2. Melakukan pengajuan logbook judul, sempro, sidang skripsi, serah terima alat.
	3. Melakukan perbaikan revisi.
	4. Monitoring pengajuan yang sudah diajukan.
	5. Mengunggah dokumen skripsi.

2. Kebutuhan Non-Fungsional

Kebutuhan ini memberikan Gambaran sistem aplikasi pengelolaan skripsi itu bekerja, seperti yang dijelaskan pada Tabel 3.

Tabel 3. Kebutuhan Non-Fungsional

<i>Kategori</i>	<i>Kebutuhan</i>
Portabilitas	Aplikasi ini dapat diakses dimana saja dan kapan saja karena hanya membutuhkan perangkat yang <i>browser</i> dan koneksi internet.
Kegunaan	Aplikasi ini mempunyai tampilan antarmuka yang mudah un-

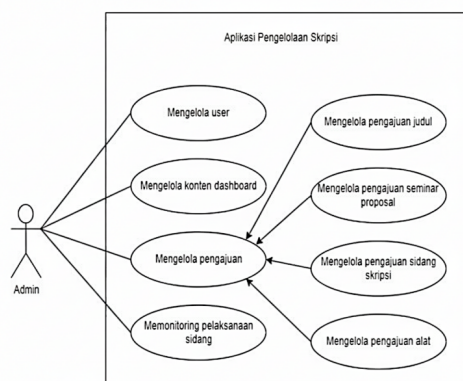
Kategori	Kebutuhan
	tuk dimengerti.
Dokumentasi	Aplikasi ini terdokumentasi dengan cukup baik dengan adanya perancangan sistem dan desain sistem.
Ketersediaan dokumen	Aplikasi ini menyediakan <i>form</i> dan skripsi dalam satu <i>platform</i> , sehingga memudahkan untuk diakses bahkan diunduh.

3.2. Membangun Prototyping

Tahap ini menghasilkan representasi abstrak sebagai gambaran dalam pembuatan aplikasi yang berdasarkan pada kebutuhan pengguna. Dengan adanya tahapan ini, tentunya akan memberikan gambaran lebih jelas dan pembuatan aplikasi yang lebih terstruktur.

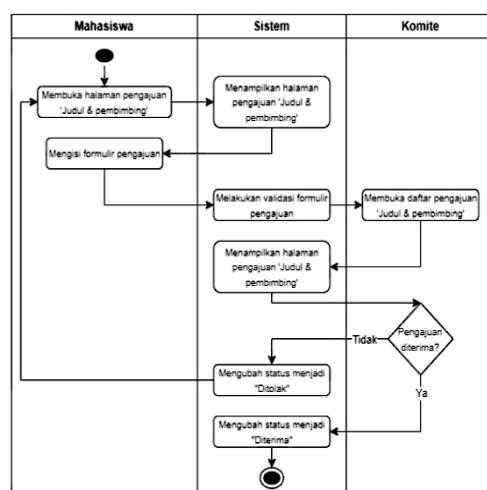
1. Use Case Diagram

Gambar 3 merupakan *use case diagram* pada *role* admin. Admin dapat melakukan pengolahan data untuk komite, dosen, dan mahasiswa. Pengolahannya dapat berupa menambah pengguna baru, melihat data pengguna, mengubah data pengguna, dan menghapusnya. Admin juga mengunggah *timeline* dan alur skripsi. Admin juga dapat melakukan monitoring segala bentuk pengajuan judul, seminar proposal (*sempro*), skripsi, dan alat yang memungkinkan admin untuk menerima pengajuan tersebut bila diinginkan. Selain itu, admin juga dapat memonitoring pengajuan revisi yang diajukan dan pelaksanaan sidang yang sedang berlangsung maupun yang sudah selesai.



Gambar 3. Use Case Diagram Admin

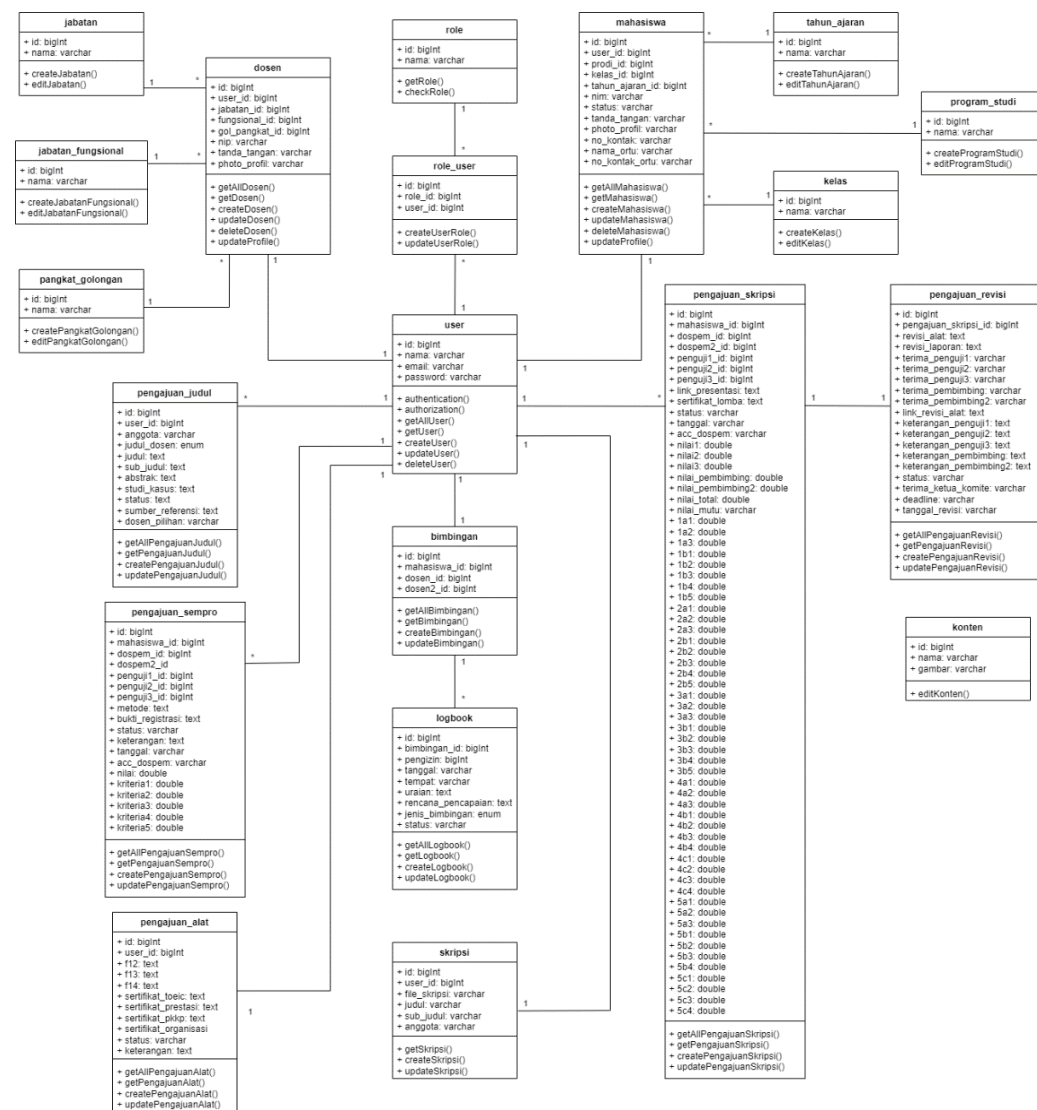
2. Activity Diagram



Gambar 4. Activity Diagram pengajuan judul dan dosen pembimbing

3. Class Diagram

Error! Reference source not found. menunjukkan *class diagram* pada aplikasi pengelolaan skripsi. *Class Diagram* ini memberikan Gambaran tentang struktur dan interaksi antar *class* dengan lengkap yang tentunya berperan sangat penting dalam pembangunan aplikasi ini. Detil nama *class* dapat dilihat pada Tabel 4.



Gambar 5. Class Diagram

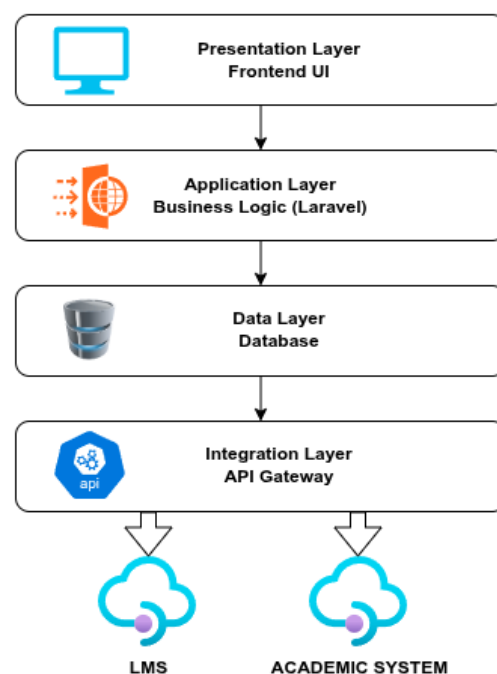
Tabel 4. Nama-nama Class

No	Nama Class
1	user
2	dosen
3	mahasiswa
4	role
5	role_user
6	jabatan
7	tahun_ajaran
8	program_studi

No	Nama Class
9	kelas
10	jabatan_fungsional
11	pangkat_golongan
12	pengajuan_judul
13	pengajuan_skripsi
14	pengajuan_revisi
15	pengajuan_semprom
16	bimbingan
17	logbook
18	konten
19	pengajuan_alat

4. Arsitektur Sistem

Sistem ini dikembangkan dengan menggunakan pendekatan *layered architecture* yang dapat dilihat pada Gambar 1. Keunggulan arsitektur pada sistem adalah mendukung *scalability* melalui pemisahan layanan dapat diintegrasikan dengan sistem informasi akademik dan Learning Management System (LMS). Sistem ini juga menggunakan REST API untuk mendukung *interoperability*.



Gambar 6. Arsitektur Sistem

Penggunaan arsitektur ini memungkinkan untuk diadopsi pada berbagai perguruan tinggi (*generic system*). Melalui pendekatan ini, sistem yang dikembangkan tidak terbatas pada kebutuhan di PNJ saja, melainkan berpotensi menjadi *generic academic workflow system* yang dapat diterapkan di berbagai institusi dengan penyesuaian minimal pada lapisan integrasi.

3.3. Evaluasi Prototyping

Tahap ini melakukan evaluasi pada pembangunan *prototype* yang sudah dibuat sebelumnya. Jika *prototype* belum sesuai dengan yang diharapkan, maka dilakukan perbaikan dengan cara mengulangi langkah-langkah sebelumnya. Selain itu, jika *prototype* sudah sesuai dengan yang diharapkan, maka dapat melanjutkan ke langkah

berikutnya, yaitu tahap mengkodekan sistem untuk pembuatan desain aplikasi antarmuka (UI) dan logika pada aplikasi.

3.4. Mengkodekan Sistem

Proses pengkodean dilakukan dari tahap evaluasi prototype yang sudah dilakukan sebelumnya. Aplikasi dibangun menggunakan Framework Laravel. Sistem diimplementasikan menggunakan *server* kampus. Setiap user memiliki yang fitur atau menu yang disesuaikan dengan *role* masing-masing. Gambar 7 merupakan tampilan halaman utama pengguna mahasiswa pada aplikasi.

The screenshot displays the student portal interface for Politeknik Negeri Jakarta. The main heading is 'Pengajuan Seminar Proposal'. Below this, there is a form with two main sections: 'Abstrak*' and 'Metode Penyelesaian Masalah*'. The 'Abstrak*' section contains the text 'ini adalah abstrak'. The 'Metode Penyelesaian Masalah*' section contains the text 'Tahap 1, Tahap 2, Tahap3, dst'. At the bottom of the form, there is a small text area with the placeholder 'Nama Anggota tim (Jika ada, contoh penulisan: ilham_budi_deti)'. The top navigation bar includes links for 'Home', 'Logbook', 'Pengajuan', 'Informasi', 'Skripsi', and 'Revisi Sidang'. The user's name 'Rizki' is displayed in the top right corner.

Gambar 7. Tampilan Halaman Mahasiswa

3.5. Menguji Sistem

1. Pengujian Blackbox

Pengujian Blackbox ini terbagi menjadi dua jenis skenario, yaitu *positive case* dan *negative case*. Gambaran detail item pengujian pada role admin terlihat pada Tabel 5. Pengujian Blackbox. User lainnya akan memiliki detail item pengujian disesuaikan dengan fitur yang terdapat pada *role* masing-masing.

Tabel 5. Pengujian Blackbox

Modul	Skenario Uji	Expected Result	Actual Result	Kesimpulan
Melakukan login.	<i>Positive case</i> : memasukkan email dan password dengan benar, lalu tekan tombol login.	Login berhasil, lalu akan menampilkan halaman home.	Login berhasil, lalu akan menampilkan halaman home.	Sesuai
	<i>Negative case</i> : memasukkan email dan password dengan salah, lalu tekan tombol login.	Login gagal, lalu akan mendapatkan pesan kesalahan yang sesuai.	Login gagal, lalu akan mendapatkan pesan kesalahan yang sesuai.	Sesuai
Melakukan logout.	<i>Positive case</i> : menekan tombol logout.	Logout berhasil, lalu menampilkan halaman login.	Logout berhasil, lalu menampilkan halaman login.	Sesuai
Mendapatkan data pengguna	<i>Positive case</i> : nama profile di pojok kanan atas berubah	Nama profile menampilkan sesuai pengguna yang	Nama profile menampilkan sesuai pengguna yang	Sesuai

Modul	Skenario Uji		Expected Result	Actual Result	Kesimpulan
yang sedang login.	sesuai dengan login.	pengguna yang sedang login.		login.	

2. System Usability Scale (SUS)

SUS adalah metode pengujian yang melibatkan pengguna akhir untuk melakukan penggunaan aplikasi secara langsung dengan tujuan mengevaluasi kegunaan (*usability*) dari aplikasi yang diujikan. SUS terdiri dari 10 pertanyaan yang dinilai menggunakan skala likert 5 poin, yaitu Sangat Tidak Setuju, Tidak Setuju, Ragu-Ragu, Setuju, dan Sangat Setuju. Jumlah pengguna yang terlibat dalam pengujian ini biasanya berada pada rentang 8 hingga 12 orang untuk mendapatkan hasil yang reliabel. Tabel 6 menunjukkan hasil kuesioner dari 12 responden dengan total 10 pertanyaan. Rumusnya adalah untuk setiap pertanyaan ganjil dihitung dengan total skor – 1, sementara itu untuk setiap pertanyaan genap dihitung dengan 5 – total skor. Langkah selanjutnya adalah jumlah skor masing-masing responden lalu dikalikan dengan 2,5 (Tabel 7). Hasil skor SUS menunjukkan aplikasi dapat diterima dengan baik oleh user (pengguna) dengan skor nilai sebesar 84,375 dengan Grade A (Excellent) (Tabel 7).

Tabel 6. Hasil Kuesioner SUS

Responden	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	4	4	4	4	4	4	4	4	4	4
2	4	3	4	1	4	3	3	4	4	3
3	4	3	3	4	4	4	4	4	4	4
4	4	3	4	3	4	1	4	3	4	3
5	4	4	3	4	3	4	4	4	4	4
6	4	2	3	2	3	3	3	2	3	2
7	3	3	3	1	3	3	3	3	3	3
8	4	3	3	0	2	3	4	3	3	2
9	4	4	4	4	4	4	4	4	4	4
10	3	3	4	3	4	4	3	4	4	4
11	3	3	3	3	3	3	3	3	4	2
12	3	4	3	3	3	3	4	4	4	4

Tabel 7. Perhitungan Skor SUS

Responden	Total Skor	SUS (Total Skor × 2,5)	Grade	Adjective Rating
1	40	100	A	Excellent
2	33	82,5	A	Excellent
3	38	95	A	Excellent
4	33	82,5	A	Excellent
5	38	95	A	Excellent
6	27	67,5	C	Okay
7	28	70	B	Good
8	27	67,5	C	Okay
9	40	100	A	Excellent
10	36	90	A	Excellent
11	30	75	B	Good
12	35	87,5	A	Excellent
Rata-rata skor SUS		84,375	A	Excellent

3. Uji Keamanan Sistem

Pengujian keamanan pada sistem yang dikembangkan dilakukan menggunakan pendekatan dasar dari Open Worldwide Application Security Project (OWASP) yang meliputi:

- * SQL Injection testing
- * Cross-Site Scripting (XSS)
- * Authentication and session management

Hasil yang diperoleh melalui uji tersebut adalah sistem berhasil menangani input tidak valid dan memiliki proteksi melalui Laravel ORM (Eloquent), CSRF Token, dan Middleware *authentication*.

4. Uji Performa

Pengujian performa dilakukan dengan simulasi beban menggunakan skenario *multi-user*. Sistem menunjukkan rata-rata response time senilai < 2 detik dengan *concurrent user* sebanyak 50 s.d. 100 *user*. Hal ini menunjukkan sistem cukup responsif untuk skala implementasi di jurusan.

3.6. Evaluasi Sistem

Setelah melewati langkah mengkodekan sistem lalu melakukan pengujian pada sistem, pada langkah ini memastikan kedua langkah tersebut berjalan sesuai harapan. Jika belum sesuai harapan, maka harus mengulang ke tahap mengkodekan sistem, lalu dilakukan pengujian kembali. Evaluasi sistem pada penelitian ini dilakukan melalui tujuh kali iterasi. Hasil pengujian yang ditampilkan merupakan hasil pengujian dari iterasi yang terakhir.

Hasil evaluasi ini menunjukkan bahwa sistem tidak hanya memiliki tingkat *usability* yang tinggi, tetapi juga memenuhi aspek kualitas perangkat lunak lainnya seperti keamanan dan kehandalan. Dibandingkan pendekatan konvensional berbasis Google Form, sistem ini memberikan peningkatan signifikan dalam:

- * Monitoring *real-time*
- * Integrasi data
- * Pengurangan redundansi data

Namun demikian, sistem masih memiliki keterbatasan pada belum diterapkannya arsitektur *microservices* secara penuh dan belum adanya integrasi langsung dengan sistem informasi akademik *production*. Hal ini dapat menjadi peluang untuk dilakukan penelitian lanjutan.

3.7. Analisis Efektivitas Sistem

Evaluasi tidak hanya difokuskan pada aspek *usability* melalui metode SUS, tetapi juga mencakup pengukuran efektivitas sistem berdasarkan perbandingan proses pengelolaan skripsi sebelum dan sesudah implementasi aplikasi. Tabel 8 menunjukkan perbandingan rata-rata waktu yang dibutuhkan untuk beberapa aktivitas utama.

Tabel 8. Perhitungan Skor SUS

Aktifitas	Sistem Manual	Sistem Usulan
Pengajuan Judul	3–5 hari	< 1 hari
Penetapan Pembimbing	5–7 hari	1–2 hari
Verifikasi Logbook	2–3 hari	Real-time
Pengajuan Proposal Seminar	2 hari	< 1 hari
Pengajuan Sidang Skripsi	2–3 hari	< 1 hari

Data yang diperoleh menunjukkan bahwa penerapan aplikasi memberikan dampak positif terhadap efisiensi administrasi skripsi, yang ditunjukkan oleh percepatan waktu

proses lebih dari 60% dibandingkan pendekatan sebelumnya yang masih bergantung pada Google Form dan pengelolaan dokumen secara manual.

Selain melakukan pengukuran usability menggunakan metode System Usability Scale (SUS), penelitian ini juga mengumpulkan masukan kualitatif dari pengguna terkait pengalaman mereka dalam menggunakan aplikasi yang dikembangkan. Berdasarkan hasil evaluasi terhadap 12 responden, sebanyak 91,7% menyatakan bahwa aplikasi membantu mempermudah kegiatan pemantauan progres skripsi. Selain itu, 83,3% responden menilai bahwa proses pengajuan berbagai tahapan skripsi dapat dilakukan lebih cepat dibandingkan mekanisme yang digunakan sebelumnya. Seluruh responden (100%) juga menyampaikan bahwa status pengajuan dapat dipantau dengan lebih mudah melalui fitur pemantauan secara real-time. Di sisi lain, 75% responden memberikan saran agar sistem pada pengembangan berikutnya dilengkapi dengan fitur notifikasi otomatis melalui media email maupun WhatsApp.

Berdasarkan tanggapan terbuka yang diberikan pengguna, beberapa manfaat yang paling sering disampaikan adalah meningkatnya kejelasan informasi mengenai status pengajuan, kemudahan dosen pembimbing dalam memonitor aktivitas bimbingan melalui logbook elektronik, serta pengelolaan dokumen yang menjadi lebih terstruktur dan sistematis. Temuan tersebut menunjukkan bahwa aplikasi yang dikembangkan tidak hanya memperoleh tingkat penerimaan yang tinggi dari pengguna, sebagaimana tercermin pada skor SUS sebesar 84,375, tetapi juga mampu memberikan kontribusi nyata terhadap peningkatan efektivitas dan efisiensi proses administrasi skripsi.

3.8. Penggunaan Sistem

Berdasarkan hasil pengujian yang telah dilakukan, aplikasi Pengelolaan Skripsi berbasis web ini telah memenuhi kebutuhan pengguna dan beroperasi dengan baik. Dengan demikian, aplikasi tersebut dinyatakan siap digunakan oleh Jurusan Teknik Informatika dan Komputer Politeknik Negeri Jakarta.

4. Kesimpulan

Penelitian ini tidak sekadar menghasilkan aplikasi manajemen skripsi berbasis *web*, melainkan juga mengusulkan desain arsitektur sistem yang modular dan *interoperable*. Berdasarkan hasil evaluasi, sistem menunjukkan tingkat *usability* yang tinggi (SUS = 84,375) serta memenuhi kriteria kualitas perangkat lunak dari aspek keamanan dan performa. Kontribusi utama terletak pada desain sistem yang bersifat generik dan adaptif, sehingga berpotensi untuk diintegrasikan dengan berbagai sistem akademik yang ada. Dengan demikian, penelitian ini melampaui pendekatan penelitian terdahulu yang cenderung berfokus pada implementasi aplikasi tanpa memperhatikan aspek arsitektur sistem dan interoperabilitas data.

Secara keseluruhan, hasil evaluasi memperlihatkan bahwa sistem yang dikembangkan mampu meningkatkan kualitas pengelolaan skripsi melalui percepatan proses administrasi lebih dari 60% dan pengurangan kesalahan administrasi sebesar 77,8%. Selain itu, nilai SUS sebesar 84,375 yang termasuk kategori Grade A (Excellent) menunjukkan tingkat kepuasan dan penerimaan pengguna yang tinggi. Dengan demikian, aplikasi yang diusulkan terbukti tidak hanya memenuhi kebutuhan fungsional pengguna, tetapi juga memberikan nilai tambah dalam bentuk peningkatan efisiensi operasional pada pengelolaan skripsi di perguruan tinggi.

Ucapan Terima Kasih: Kepada semua civitas akademika Politeknik Negeri Jakarta khususnya Jurusan Teknik Informatika dan Komputer yang telah mendukung terlaksananya penelitian ini. Ucapan terima kasih juga kami sampaikan kepada Unit UPATIK di Politeknik Negeri Jakarta yang telah memberikan fasilitas server untuk *men-deploy* aplikasi ini.

Referensi

- [1] A. Mahalia, "Skripsi: Pengertian, Jenis, Unsur dan Tahapan dalam Menyusunnya," Skripsi: Pengertian, Jenis, Unsur dan Tahapan dalam Menyusunnya. Accessed: Feb. 20, 2024. <https://www.detik.com/jatim/berita/d-6905827/skripsi-pengertian-jenis-unsur-dan-tahapan-dalam-menyusunnya>
- [2] A. Kasim, L. Hadjaratie, and R. H. Dai, "Rancang Bangun Sistem Informasi Skripsi dan Kerja Praktik Berbasis Web," *Jambura J. Inform.*, vol. 2, no. 2, pp. 95–107, Oct. 2020, <https://doi.org/10.37905/jji.v2i2.5331>.
- [3] I. Ilhamsyah, A. P. Kharisma, and N. Yudistira, "Pengembangan Aplikasi Manajemen Tugas berbasis Android (Studi pada CV. Cheleron Production)" *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 12, pp. 5221–5227, 2021. <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/10208>
- [4] O. C. E. Kainde, "Perancangan Aplikasi Penyedia Informasi Kerja Siswa SMK Berbasis Web Menggunakan Framework Laravel," *JATISI J. Tek. Inform. Dan Sist. Inf.*, vol. 9, no. 3, pp. 2191–2204, Sept. 2022, <https://doi.org/10.35957/jatisi.v9i3.2333>.
- [5] R. E. Saputra, Z. Zulhalim, I. Ibrahim, S. Waluyo, and A. S. Rini, "Perancangan Aplikasi Student Get Student Berbasis Web Menggunakan Framework Laravel Pada STIE & STMIK Jayakarta," *J. Manajemen Inform. Jayakarta*, vol. 1, no. 3, p. 196, July 2021, <https://doi.org/10.52362/jmijayakarta.v1i3.158>.
- [6] I. Ilhamsyah, A. P. Kharisma, and N. Yudistira, "Pengembangan Aplikasi Manajemen Tugas berbasis Android (Studi pada CV. Cheleron Production)" *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 12, pp. 5221–5227, 2021. <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/10208>
- [7] S. Pratasik and I. Rianto, "Pengembangan Aplikasi E-DUK Dalam Pengelolaan SDM Menggunakan Metode Agile Development," *Cogito Smart J.*, vol. 6, no. 2, pp. 204–216, Dec. 2020, <https://doi.org/10.31154/cogito.v6i2.267.204-216>.
- [8] A. Mahdiyan and P. Rosyani, "Rancang Bangun Aplikasi Pengajuan Cuti Karyawan Berbasis Android Pada PT. Surya Toto Indonesia TBK," *Biner J. Ilm. Inform. Dan Komput.*, vol. 1, no. 1, pp. 10–16, Jan. 2022, <https://doi.org/10.32699/biner.v1i1.2418>.
- [9] Iqbal, "Tahapan Metode Prototype, Pengertian Metode Prototype dan Contohnya." <https://nextgen.co.id/tahapan-metode-prototype-pengertian-metode-prototype-dan-contohnya/>
- [10] D. Rizky, "Mengenai Prototyping," Medium. Accessed: Dec. 25, 2025. <https://medium.com/dot-intern/sdlc-metode-prototype-8f50322b14bf>
- [11] F. P. E. Putra, R. W. Efendi, A. B. Tamam, and W. A. Pramadi, "Tren dan Praktik Terbaik dalam Pengembangan Web Berbasis API : Kajian Literatur terhadap Framework Laravel dan React," *Infomatek*, vol. 27, no. 1, pp. 165–178, June 2025. <https://doi.org/10.23969/infomatek.v27i1.25122>.
- [12] F. Sinlae, E. Irwanda, Z. Maulana, and V. Eka Syahputra, "Penggunaan Framework Laravel dalam Membangun Aplikasi Website Berbasis PHP," *J. Siber Multi Disiplin*, vol. 2, no. 2, pp. 119–132, July 2024, <https://doi.org/10.38035/jsmd.v2i2.186>.
- [13] Budianto, A. Wibawa, and R. M. Said, "Pengenalan Laravel Sebagai Framework Open Source Dalam Membangun Sebuah Aplikasi Berbasis Web Di Smk Islamiyah Ciputat," *J. Perad. Masy.*, vol. 5, no. 3, pp. 128–138, Mei 2025. <https://doi.org/10.55182/jpm.v5i3.548>
- [14] "Laravel development trends 2024," Esokia. Accessed: Dec. 25, 2025. [Online]. Available: <https://esokia.com/en/blog/laravel-development-trends-2024>
- [15] "Advantages and disadvantages of Laravel Framework for web Development," DDI Development. Accessed: Dec. 24, 2025. [Online]. Available: <https://ddi-dev.com/blog/programming/pros-and-cons-of-laravel-framework-for-web-app-development/>
- [16] M. Miftakhul, F. Zahro, R. Hakim, and M. S. Akbar, "Pengujian Black Box Testing Pada Sistem Website Pemesanan Online Toko Ayam Krispy," *J. MEDIA Akad. JMA*, vol. 3, no. 5, Mei 2025. <https://jurnal.mediaakademik.com/index.php/jma/article/view/1908>
- [17] A. Samdono, A. P. Sari, and F. P. Aditiawan, "pengujian black box pada sistem informasi stok dan penjualan berbasis website menggunakan metode equivalence partitioning (studi kasus: CV. Algani Karya Mandiri)," *JATI J. Mhs. Tek. Inform.*, vol. 8, no. 1, Feb. 2024. <https://doi.org/10.36040/jati.v8i1.8893>
- [18] F. G. Sembodo, G. F. Fitriana, and N. A. Prasetyo, "Evaluasi Usability Website Shopee Menggunakan System Usability Scale (SUS)," *J. Appl. Inform. Comput. JAIC*, vol. 5, no. 2, pp. 146–150, Desember 2021.
- [19] A. Saputra, "Penerapan Usability pada Aplikasi PENTAS Dengan Menggunakan Metode System Usability Scale (SUS)," *JTIM J. Teknol. Inf. Dan Multimed.*, vol. 1, no. 3, pp. 206–212, Nov. 2019, <https://doi.org/10.35746/jtim.v1i3.50>.
- [20] N. S. Utami, K. S. Suhaimi, T. Kasnalestari, G. R. Banurea, M. I. Robani, and D. Rahmadana, "Pengujian Usability Testing Menggunakan Metode System Usability Scale untuk Menilai Aplikasi SIKUU," *JTIM J. Teknol. Inf. Dan Multimed.*, vol. 7, no. 4, pp. 689–697, Oct. 2025, <https://doi.org/10.35746/jtim.v7i4.738>.