

Implementasi Metode *Iteratif*SDLC pada Sistem Informasi Skripsi dan Tugas Akhir Berbasis Web Fakultas Teknologi Informasi Universitas Merdeka Malang

Ardhillah Habibi Al-Fath ^{1*}, Nofrian Deny Hendrawan ²

¹ Program Studi Sistem Informasi Universitas Merdeka Malang

² Program Studi Sistem Informasi Universitas Merdeka Malang

Alamat: Jl. Terusan Dieng No. 62-64 Klojen, Pisang Candi, Sukun, Malang, Jawa Timur

Email: ardhillalahalfath@gmail.com¹, nofrian.hendrawan@unmer.ac.id²

* Penulis Korespondensi: Ardhillah Habibi Al-Fath

Abstract: The management of theses and final projects at the Faculty of Information Technology, Universitas Merdeka Malang, is still conducted manually through paper forms, WhatsApp messages, and electronic mail. This condition causes thesis-related data and process stages to be managed without an integrated system. This study aims to design and develop a web-based Thesis and Final Project Information System using the Software Development Life Cycle (SDLC) method with an iterative approach. The system was developed using the Laravel 11 framework, MariaDB database, and Docker as a containerization environment. The system consists of four main modules: title submission, academic supervision, proposal seminar, and final seminar, involving three user roles: students, lecturers, and administrative staff. Testing was conducted using the black-box testing method on six functional groups: authentication, title submission, academic supervision, proposal seminar, final seminar, and supervisor or examiner assignment. A total of 36 test scenarios were selected based on the main functional requirements, input validation, user access rights, and process flows of each module. Each scenario was tested by comparing the actual result with the expected result in a Docker environment by the researcher. All 36 scenarios produced results that were consistent with the expected outcomes, resulting in a success rate of 100%. This result indicates that the tested functions operated according to the specifications defined in the selected test scenarios. However, the percentage does not demonstrate that the system is generally effective, usable, or free from errors beyond the scope of the conducted testing.

Keywords: information system; thesis management; SDLC; iterative method; black box testing; Laravel; Docker

Abstrak: Pengelolaan skripsi dan tugas akhir di Fakultas Teknologi Informasi, Universitas Merdeka Malang masih dilakukan secara manual melalui formulir kertas, pesan WhatsApp, dan surat elektronik. Kondisi tersebut menyebabkan proses pengelolaan data dan pemantauan tahapan skripsi belum terintegrasi dalam satu sistem. Penelitian ini bertujuan merancang dan membangun Sistem Informasi Skripsi dan Tugas Akhir berbasis web menggunakan metode Software Development Life Cycle (SDLC) dengan pendekatan iteratif. Sistem dikembangkan menggunakan framework Laravel 11, basis data MariaDB, dan Docker sebagai lingkungan kontainerisasi. Sistem mencakup empat modul utama, yaitu pengajuan judul, bimbingan, seminar proposal, dan seminar hasil, serta melibatkan tiga peran pengguna, yaitu mahasiswa, dosen, dan administrasi. Pengujian dilakukan menggunakan metode black-box testing terhadap enam kelompok fungsi, yaitu autentikasi, pengajuan judul, bimbingan, seminar proposal, seminar hasil, dan plotting. Sebanyak 36 skenario pengujian dipilih berdasarkan kebutuhan fungsional utama, validasi masukan, hak akses pengguna, serta alur proses pada setiap modul. Setiap skenario diuji dengan membandingkan hasil aktual terhadap hasil yang diharapkan pada lingkungan pengujian Docker oleh peneliti. Seluruh 36 skenario menghasilkan keluaran yang sesuai dengan hasil yang diharapkan sehingga memperoleh tingkat keberhasilan sebesar 100%. Hasil tersebut menunjukkan bahwa fungsi-fungsi yang diuji dapat berjalan sesuai dengan spesifikasi pada skenario pengujian yang telah ditentukan. Persentase tersebut tidak dimaksudkan untuk menyatakan bahwa sistem telah terbukti efektif, mudah digunakan, atau bebas dari seluruh kesalahan di luar cakupan pengujian.

Kata kunci: sistem informasi; manajemen skripsi; SDLC; metode iteratif; black box testing; Laravel; Docker

Masuk: 09 Juli 2026

Revisi: 12 Agustus 2026

Diterima: 30 September 2026

Tersedia: 30 September 2026

Terbit: 30 September 2026



Hak cipta: © 2026 oleh penulis.

Diserahkan untuk kemungkinan

publikasi akses terbuka

berdasarkan syarat dan ketentuan

lisensi Creative Commons

Attribution (CC BY SA) (

<https://creativecommons.org/licenses/by-sa/4.0/>)

1. Pendahuluan

Skripsi dan tugas akhir merupakan karya ilmiah mahasiswa sebagai syarat kelulusan utama untuk mendapat gelar program sarjana. Pengelolaan selama pengerjaan skripsi harus dilakukan dengan tertib, transparan dan efisien untuk memastikan kualitas akademik yang optimal. Namun pada kenyataannya perguruan tinggi di Indonesia masih belum bisa mengelola skripsi secara efisien karena keterbatasan sistem informasi pendukung [1].

Fakultas Teknologi Informasi Universitas Merdeka Malang masih mengelola proses skripsi dan tugas akhir, mulai dari pengajuan judul, bimbingan, penjadwalan seminar, hingga pemantauan progres mahasiswa melalui surat, pesan WhatsApp, dan formulir kertas. Kondisi tersebut menyebabkan informasi belum terpusat dan menyulitkan koordinasi antara mahasiswa, dosen, dan administrasi.

Beberapa penelitian terdahulu telah mengembangkan sistem untuk mendukung pengelolaan tugas akhir, tetapi masih memiliki ruang lingkup yang terbatas. Dewi dkk. [2] mengembangkan sistem digitalisasi manajemen tugas akhir berbasis web yang berfokus pada pengelolaan dan pemantauan tugas akhir, tetapi belum menjelaskan integrasi tahapan pengajuan judul, bimbingan, seminar proposal, dan seminar hasil secara menyeluruh dalam satu alur sistem. Haykhal [3] mengembangkan sistem pengajuan judul tugas akhir berbasis web, tetapi cakupannya terbatas pada proses pengajuan judul dan belum mencakup proses bimbingan serta tahapan seminar. Sementara itu, Ramadhan dkk. [4] menerapkan metode Agile pada pengembangan aplikasi berbasis Laravel, tetapi objek penelitiannya adalah pengelolaan data magang sehingga alur bisnis dan kebutuhan sistemnya berbeda dari pengelolaan skripsi dan tugas akhir. Berdasarkan keterbatasan tersebut, masih terdapat celah penelitian berupa kebutuhan terhadap sistem yang mengintegrasikan rangkaian proses skripsi dan tugas akhir secara berurutan, mulai dari pengajuan judul hingga seminar hasil, dengan pembagian hak akses bagi mahasiswa, dosen, dan administrasi. Penelitian ini mengisi celah tersebut melalui pengembangan sistem informasi yang disesuaikan dengan kebutuhan proses akademik di FTI Universitas Merdeka Malang.

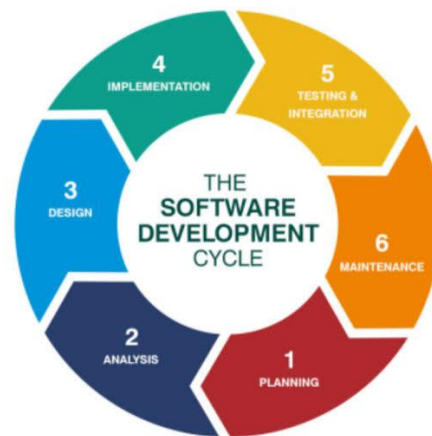
Penelitian ini bertujuan merancang dan membangun Sistem Informasi Skripsi dan Tugas Akhir di Fakultas Teknologi Informasi Universitas Merdeka Malang menggunakan metode SDLC dengan pendekatan iteratif. Sistem yang dikembangkan mengintegrasikan proses pengajuan judul, bimbingan, seminar proposal, dan seminar hasil bagi tiga peran pengguna, yaitu mahasiswa, dosen, dan administrasi. Kontribusi penelitian ini tidak hanya berupa aplikasi, tetapi juga berupa rancangan alur digital dan pembagian fungsi berdasarkan peran pengguna yang dapat dijadikan referensi untuk pengembangan sistem akademik dengan proses persetujuan, penjadwalan, dan pemantauan bertahap. Meskipun demikian, kontribusi tersebut terbatas pada konteks proses akademik di FTI Universitas Merdeka Malang sehingga penerapannya pada institusi lain memerlukan penyesuaian terhadap kebijakan dan kebutuhan setempat.

2. Tinjauan Literatur

2.1. Sistem Informasi

Sistem Informasi adalah kumpulan komponen yang bekerja sama untuk menyimpan, mengelola dan menampilkan informasi atau data yang diperlukan untuk mendukung pengambilan keputusan dalam suatu organisasi [1]. Dalam konteks akademik, sistem informasi bertepatan penting dalam mengintegrasikan berbagai proses administratif kepada seluruh sivitas yang berperan dalam pengelolaan data supaya memperoleh informasi yang akurat dan tepat waktu [5].

2.2. Software Development Life Cycle(SDLC)



Gambar 1. Software Development Life Cycle

SDLC merupakan kerangka pengembangan perangkat lunak berbasis best practice yang terdiri dari tahapan Planning, Requirements, Analysis, System Design, Implementation, Testing, Deployment dan Maintenance [6] [7]. Tujuan utama SDLC adalah menghasilkan perangkat lunak berkualitas tinggi yang sesuai dengan kebutuhan organisasi melalui pengembangan yang terkontrol dan adaptif pada setiap fase pengembangan [8].

2.3. Metode Iteratif

Metode iteratif merupakan pendekatan pengembangan perangkat lunak yang membagi proses pengembangan ke dalam beberapa pengulangan. Setiap iterasi mencakup perencanaan, analisis kebutuhan, perancangan, implementasi, pengujian, dan evaluasi terhadap bagian sistem yang dikembangkan. Hasil evaluasi pada satu iterasi digunakan sebagai masukan untuk perbaikan dan penyempurnaan pada iterasi berikutnya [9] [10] [11]. Dalam penelitian ini, pendekatan iteratif diterapkan pada setiap modul sistem sehingga pengembangan dapat disesuaikan secara bertahap dengan kebutuhan pengguna.

2.4. Docker

Docker adalah platform kontainerisasi open-source yang memungkinkan pengembang membangun, mengemas dan menjalankan aplikasi secara konsisten di berbagai lingkungan server tanpa masalah depedensi [12] [13]. Dengan menggunakan kontainer, tim pengembang dapat meminimalisir resiko ketidakcocokan depedensi yang sering menjadi hambatan dalam perilis perangkat lunak [14]. Dalam penelitian ini, Docker digunakan untuk proses deployment sistem ke server produksi, memastikan konsistensi lingkungan antara tahap pengembangan dan produksi.

3. Metode

Penelitian ini menggunakan metode kualitatif dengan desain penelitian tindakan (action research). Lokasi penelitian ini adalah Fakultas Teknologi Informasi Universitas Merdeka Malang. Pengumpulan data dilakukan melalui wawancara terstruktur dengan Kepala Program Studi dan Wakil Dekan untuk mengidentifikasi kebutuhan sistem yang sesuai.

3.1. Alur Pengembangan Iteratif

Pengembangan sistem dilakukan secara iteratif melalui tahapan berikut pada setiap iterasi:

- Perencanaan (Planning): Identifikasi masalah, studi literatur, dan penentuan tujuan pengembangan pada iterasi.
- Analisis Kebutuhan (Requirements Analysis): Wawancara dengan pemangku kepentingan dan analisis kebutuhan fungsional sistem pada bagian yang akan dikembangkan.

- Desain Sistem (System Design): Perancangan arsitektur sistem, basis data, use case diagram, sequence diagram, dan activity diagram sesuai hasil analisis pada iterasi berjalan.
- Implementasi (Implementation): Menerapkan rancangan menjadi fungsi atau modul sistem yang dapat digunakan.
- Pengujian dan Evaluasi (Testing and Evaluation): Menguji fungsi yang telah diimplementasikan, mengevaluasi hasilnya, serta menggunakan temuan evaluasi sebagai masukan untuk iterasi berikutnya.

3.2. Pemformatan Komponen Matematika

Sistem dirancang dengan tiga peran yaitu mahasiswa, dosen dan administrasi (Kaprodi/TU). Perancangan mencakup use case diagram, activity diagram untuk setiap modul, sequence diagram yang menggambarkan interaksi antar komponen, dan Entity Relationship Diagram (ERD) dengan 9 tabel utama yaitu users, thesis, thesisdocument, mentorship, thesis_schedule, thesis_examiner, seminar_proposal, seminar_hasil, dan revisions.

Sistem diimplementasikan menggunakan framework laravel 11 dengan fitur autentikasi berbasis Google OAuth, basis data MariaDB dan dideploy menggunakan Docker container untuk memastikan konsistensi depedensi selama pengembangan dan deployment di server. Penggunaan UUID sebagai primary key diterapkan sebagai praktik keamanan by design untuk mencegah serangan enumerasi ID.

3.3. Metode Pengujian

Pengujian dilakukan menggunakan metode Blackbox testing untuk memastikan modul-modul yang telah ditentukan mampu berjalan sesuai dengan kebutuhan spesifikasi pengguna. Blackbox testing bekerja dengan Pengujian dibagi menjadi enam modul yang terdiri dari modul pengajuan judul, bimbingan, seminar proposal, seminar hasil, plotting dan autentikasi.

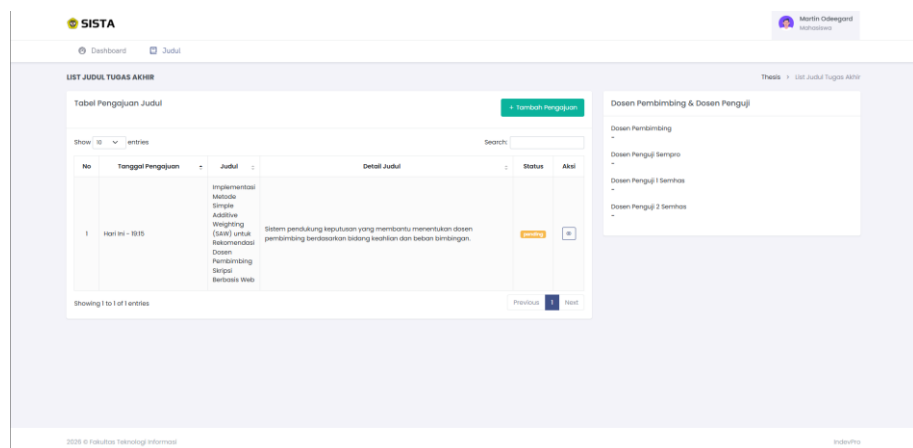
4. Hasil dan Pembahasan

4.1. Implementasi Sistem

Sistem Informasi Skripsi dan Tugas Akhir berhasil dikembangkan menggunakan framework Laravel 11 dengan basis data MariaDB dan dideploy menggunakan Docker. Sistem mencakup empat modul fungsional yang memfasilitasi 3 role yaitu mahasiswa, dosen dan administrasi.

Halaman login menggunakan autentikasi Google OAuth, sehingga pengguna hanya dapat menggunakan akun email yang sudah terdaftar pada siakad. Setelah berhasil login pengguna akan diarahkan ke dashboard sesuai dengan role masing-masing. Dashboard mahasiswa menampilkan data personal yang berhubungan dengan skripsi yaitu IPK, BTA, status dan semester, dashboard dosen menampilkan ringkasan jumlah skripsi yang dibimbing, pengajuan bimbingan dan seminar. Dashboard administrasi sama dengan dosen tetapi data yang diambil dari semua mahasiswa yang sedang aktif mengerjakan skripsi.

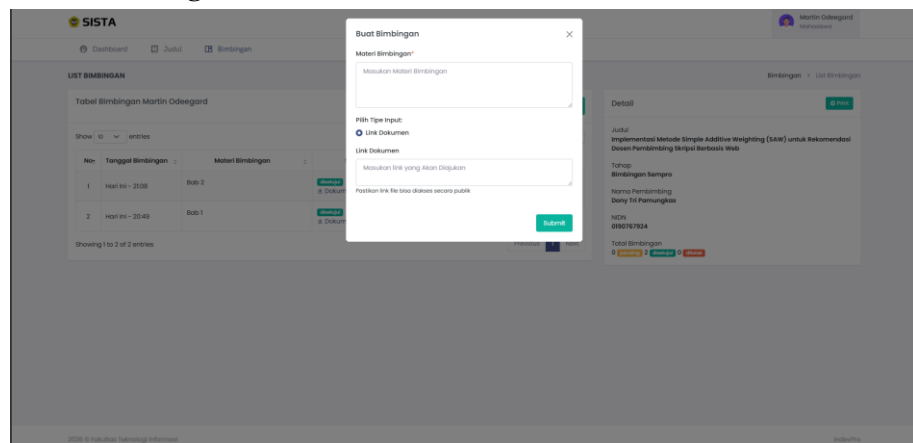
4.2. Modul Pengajuan Judul



Gambar 2. Halaman Pengajuan Judul Mahasiswa

Modul pengajuan judul memfasilitasi interaksi antara mahasiswa mengajukan judul dan administrasi memproses judul untuk menyetujui atau menolak judul yang diajukan. Mahasiswa hanya bisa mengajukan judul ketika syarat-syarat administrasi sudah terpenuhi, mulai dari syarat minimal SKS dan IPK, pembayaran BTA yang harus sudah lunas dan semester yang ditempuh. Administrasi bertugas untuk memproses judul sekaligus melakukan plotting pembimbing untuk judul yang tepat. Dosen hanya bisa melihat judul yang sudah diarahkan oleh administrasi untuk menjadi dosen pembimbing.

4.3. Modul Bimbingan

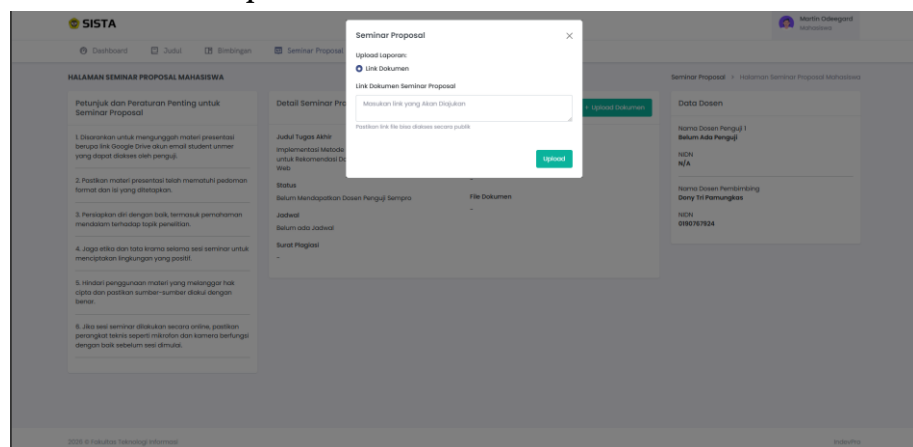


Gambar 3. Halaman Bimbingan Mahasiswa

Modul bimbingan memfasilitasi interaksi antara mahasiswa dan dosen pembimbing. Menu bimbingan pada mahasiswa baru bisa ditampilkan ketika judul sudah disetujui oleh administrasi, mahasiswa bisa mengajukan bimbingan melalui form yang sudah disediakan, mahasiswa mengisi topik dan link publik draft skripsi yang akan diajukan. Bimbingan yang sudah diajukan akan muncul pada halaman bimbingan administrasi dan dosen yang sudah ditugaskan. Hanya dosen pembimbing yang ditugaskan yang bisa memproses bimbingan, dosen pembimbing bisa memutuskan untuk menyetujui atau menolak bimbingan. Meskipun administrasi bisa melihat semua bimbingan namun administrasi hanya bisa melihat.

Jumlah bimbingan yang diajukan mahasiswa akan otomatis dihitung oleh sistem, jika mahasiswa sudah memiliki 3 bimbingan yang disetujui maka pada halaman bimbingan dosen pembimbing akan muncul tombol lanjut seminar proposal, jika mahasiswa sudah memiliki 8 bimbingan disetujui maka akan muncul tombol lanjut seminar hasil pada halaman dosen pembimbing.

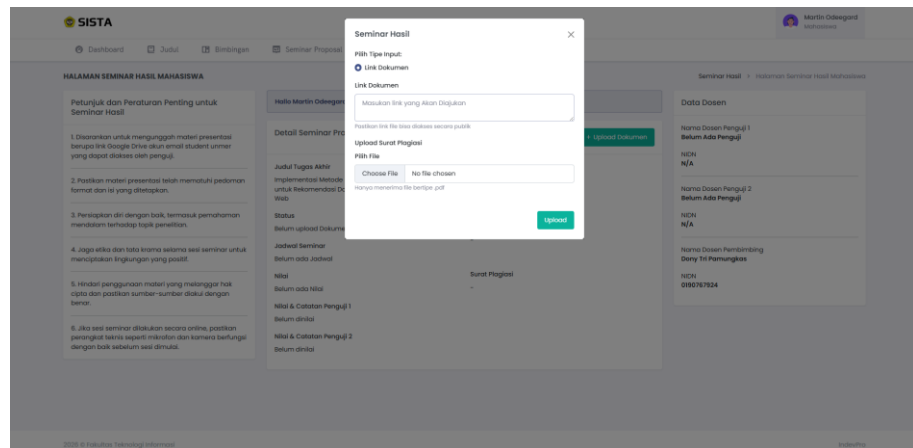
4.4. Modul Seminar Proposal



Gambar 4. Halaman Seminar Proposal Mahasiswa

Menu seminar proposal pada halaman mahasiswa baru akan ditampilkan ketika dosen pembimbing sudah menyetujui ke tahap tersebut. Mahasiswa mengajukan file proposal skripsi dalam bentuk link publik. Mahasiswa juga bisa melihat pedoman dan arahan untuk melakukan seminar proposal. Administrasi kemudian melakukan plotting pengujian 1, pengujian 2 dan menentukan jadwal seminar. Setelah melewati jadwal yang ditentukan dosen pengujian bisa melakukan penilaian, dosen pengujian diminta untuk mengisi revisi. Dosen pengujian bisa menyelesaikan status seminar proposal setelah mahasiswa menyelesaikan revisi dan bisa melanjutkan bimbingan selanjutnya untuk ke tahap seminar hasil.

4.5. Modul Seminar Hasil



Gambar 5. Halaman Seminar Hasil Mahasiswa

Menu seminar hasil baru akan ditampilkan pada halaman mahasiswa ketika dosen pembimbing sudah menyetujui ke tahap tersebut. Tidak jauh berbeda dari tahap seminar proposal mahasiswa juga diharuskan memasukkan link publik yang mengarah pada file skripsi, pada tahap seminar hasil mahasiswa juga diminta untuk memasukkan file hasil cek plagiasi. Kemudian administrasi melakukan penjadwalan seminar hasil, setelah melewati jadwal seminar hasil dosen penguji bisa melakukan penilaian akhir pada skripsi mahasiswa, setelah revisi terakhir disetujui maka mahasiswa akan ditandai oleh sistem bahwa sudah menyelesaikan skripsi.

4.6. Hasil Pengujian Blackbox Testing

Pengujian sistem dilakukan dengan menggunakan metode Blackbox Testing yang berfokus pada fungsi-fungsi utama pada sistem tanpa memperhatikan kode didalam sistem. Pengujian mencakup enam modul fungsional yaitu pengajuan judul, bimbingan, seminar proposal, seminar hasil, plotting dan autentikasi. Tbl 1 menyajikan rangkuman singkat dari hasil pengujian Blackbox pada seluruh modul sistem.

Tabel 1. Ringkasan Hasil Pengujian Blackbox

No	Modul	Skenario	Berhasil	Gagal	Presentase
1	Autentikasi	Login valid dan tidak valid,	5	0	100%
2	Pengajuan Judul	Pengajuan judul valid dan tidak valid, menerima dan menolak judul, melihat status	5	0	100%
3	Bimbingan	Pengajuan valid dan tidak valid, persetujuan diterima dan ditolak, hak akses role, melihat status dan riwayat	8	0	100%
4	Seminar Proposal	Pendaftaran valid dan tidak valid, penjadwalan valid dan tidak valid, melihat informasi seminar, input nilai seminar	6	0	100%

5	Seminar Hasil	Pendaftaran valid dan tidak valid, penjadwalan valid dan tidak valid, melihat informasi seminar, input nilai seminar	6	0	100%
6	Plotting	Plotting pembimbing valid dan tidak valid, plotting penguji valid dan tidak valid	6	0	100%

Berdasarkan hasil dari pengujian Blackbox pada Tbl 1, 36 skenario dari 6 modul berhasil dijalankan dengan sempurna dan memiliki presentase keberhasilan 100%. Hal ini menunjukkan bahwa sistem berhasil berjalan sesuai dengan spesifikasi yang sudah ditentukan.

Pengujian sistem dilakukan untuk memeriksa kesesuaian keluaran aktual dengan keluaran yang diharapkan pada skenario pengujian yang telah ditentukan. Skenario dipilih berdasarkan kebutuhan fungsional utama, hak akses setiap peran, validasi masukan, dan alur proses pada enam kelompok fungsi. Oleh karena itu, hasil keberhasilan 100% hanya berlaku pada 36 skenario yang diuji dan tidak dapat digeneralisasikan sebagai ukuran efektivitas, usability, keamanan, atau kualitas sistem secara keseluruhan.

4.7 Hasil Implementasi Docker

```
FROM alpine:latest

# Setup document root
WORKDIR /var/www/html

# Install packages and remove default server definition
RUN apk add --no-cache \
    curl \
    nginx \
    nodejs \
    npm \
    php83 \
    php83-ctype \
    php83-curl \
    php83-dom \
    php83-fileinfo \
    php83-fpm \
    php83-gd \
    php83-intl \
    php83-json \
    php83-mbstring \
    php83-mysqli \
    php83-openssl \
    php83-pdo_mysql \
    php83-pdo_pgsql \
    php83-phar \
    php83-session \
    php83-tokenizer \
    php83-xml \
    php83-xmlreader \
    php83-xmlwriter \
    php83-zip \
    supervisor

# Configure nginx
COPY config/nginx.conf /etc/nginx/nginx.conf
```

Gambar 6. DockerFile

Docker diimplementasikan sebagai kontainer untuk mawadahi depedensi yang dibutuhkan untuk project baik selama di lokal maupun server. Dengan menggunakan docker seluruh dependency aplikasi seperti PHP, Nginx dan library Laravel bisa dikemas dan digunakan dalam sebuah kontainer sehingga tidak bergantung pada konfigurasi sistem operasi host.

5. Perbandingan

Dibandingkan dengan penelitian sejenis, sistem yang dibangun dalam penelitian ini memiliki beberapa keunggulan. Haykhal [3] hanya mencakup modul pengajuan judul tanpa bimbingan dan seminar, sementara Miftakhul Ula & Sumiati [15] berfokus pada pengajuan judul dengan fitur deteksi kemiripan judul. Penelitian Dewi dkk. [2] mencakup manajemen tugas

akhir tetapi tidak mengintegrasikan proses seminar bertahap. Penelitian ini mengintegrasikan seluruh alur proses skripsi dan pengajuan judul hingga seminar hasil dalam satu platform, dilengkapi dengan mekanisme threshold bimbingan otomatis (3x untuk seminar proposal, 8x untuk seminar hasil) dan deployment menggunakan Docker untuk portabilitas yang lebih baik. Fitur autentikasi menggunakan Google OAuth juga mempermudah proses login tanpa perlu manajemen akun terpisah.

6. Kesimpulan

Penelitian ini berhasil merancang dan membangun Sistem Informasi Skripsi dan Tugas Akhir berbasis web di Fakultas Teknologi Informasi Universitas Merdeka Malang dengan menerapkan metode pengembangan SDLC dengan pendekatan iteratif. Sistem dikembangkan menggunakan framework Laravel 11, basis data MariaDb dan dideploy menggunakan docker.

Sistem berhasil mengintegrasikan empat modul utama yaitu pengajuan judul, bimbingan, seminar proposal dan seminar hasil dalam satu platform yang bisa diakses oleh tiga role yaitu dosen, mahasiswa dan administrasi. Pengujian berhasil dijalankan dengan presentase keberhasilan 100% menggunakan pengujian blackbox testing, terdapat 36 skenario dari 6 modul yang diuji. Hal ini membuktikan bahwa sistem berhasil berjalan sesuai dengan spesifikasi kebutuhan pengguna yang sebelumnya sudah ditentukan.

Untuk pengembangan selanjutnya, disarankan untuk menambahkan fitur keamanan login tambahan yaitu adanya penggunaan Personal Identification Number(PIN) sebagai lapisan keamanan tambahan, penambahan notifikasi real-time untuk memastikan informasi tersampaikan dengan cepat, serta analisis kinerja dan skalabilitas lanjutan untuk memastikan stabilitas sistem pada skala penggunaan yang besar.

Kontribusi Penulis: Ardhillah Habibi Al-Fath: konseptualisasi, perancangan sistem, implementasi modul bimbingan, pengujian, penulisan naskah. Nofrian Deny Hendrawan: supervisi dan validasi penelitian.

Pendanaan: Penelitian ini tidak menerima pendanaan eksternal.

Pernyataan Ketersediaan Data: Tidak ada data baru yang dibuat atau dianalisis selama penelitian ini. Berbagi data tidak berlaku untuk artikel ini.

Ucapan Terima Kasih: Penulis menyadari keterbatasan kemampuan tanpa adanya bantuan dari orang-orang di lingkungan penulis. Rasa terimakasih penulis ucapkan kepada semua yang terlibat pada penulisan penelitian ini baik secara langsung maupun secara tidak langsung.

Konflik Kepentingan: Penulis menyatakan tidak ada konflik kepentingan

Referensi

- [1] M. F. Adham, "Analisis Implementasi Sistem Informasi: Studi Literatur," *Jurnal Teknologi Sistem Informasi*, vol. 5, no. 1, pp. 264–275, Jun. 2024, doi: 10.35957/jtsi.v5i1.7815.
- [2] A. N. Dewi, S. Supriyono, and S. Muzid, "DIGITALISASI MANAJEMEN TUGAS AKHIR DI PROGRAM STUDI TEKNIK MESIN DENGAN SISTEM INFORMASI WEB-BASED," *Just IT : Jurnal Sistem Informasi, Teknologi Informasi, dan Komputer*, Sep. 2023, doi: 10.24853/justit.14.1.%p.
- [3] F. Haykhal, "RANCANG BANGUN SISTEM INFORMASI TAKU UNTUK PENGAJUAN JUDUL TUGAS AKHIR BERBASIS WEB DI LINGKUP JURUSAN TEKNIK INFORMATIKA," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 13, no. 3S1, Oct. 2025, doi: 10.23960/jitet.v13i3S1.8016.
- [4] E. R. Ramadhan, K. Prihandani, and A. Voutama, "Penerapan Metode Agile Pada Development Aplikasi Pengelolaan Data Magang Berbasis Web Menggunakan Framework Laravel," *Jurnal Ilmiah Wahana Pendidikan*, vol. 9, no. 7, pp. 144–154, Apr. 2023, doi: 10.5281/zenodo.7812416.

-
- [5] A. Angelina, C. Yandhika, C. L. Hartanto, M. Graciela, and A. Farisi, "Sebuah Tinjauan Literatur Sistematis tentang Metode Pengembangan Perangkat Lunak Sistem Informasi Berbasis Web," *Jurnal Teknologi Sistem Informasi*, vol. 5, no. 1, pp. 181–192, May 2024, doi: 10.35957/jtsi.v5i1.6619.
- [6] T. Pricillia and Zulfachmi, "Perbandingan Metode Pengembangan Perangkat Lunak (Waterfall, Prototype, RAD)," *Jurnal Bangkit Indonesia*, vol. 10, no. 1, pp. 6–12, Mar. 2021, doi: 10.52771/bangkitindonesia.v10i1.153.
- [7] N. Nilma, "PENERAPAN MODEL SOFTWARE DEVELOPMENT LIFE CYCLE PADA RANCANG BANGUN SISTEM PAYROLL PERUSAHAAN," *Jurnal Publikasi Teknik Informatika*, vol. 1, no. 2, pp. 61–69, Jun. 2022, doi: 10.55606/jupti.v1i2.334.
- [8] F. Anisa, F. S. Harahap, H. A. Khosyi, and I. P. Sari, "Pengembangan Software Menggunakan Model SDLC Guna Mencapai Keselarasan dengan Kebutuhan Pengguna," *Journal Of Informatics And Business*, vol. 1, no. 4, pp. 229–232, Jan. 2024, doi: 10.47233/jibs.v1i1.522.
- [9] P. R. Perkasa and E. Mailoa, "ADOPSI DEVSECOPS UNTUK MENDUKUNG METODE AGILE MENGGUNAKAN TRIVY SEBAGAI SECURITY SCANNER DOCKER IMAGE DAN DOCKERFILE," *Jurnal Indonesia : Manajemen Informatika dan Komunikasi*, vol. 4, no. 3, pp. 856–863, Sep. 2023, doi: 10.35870/jimik.v4i3.291.
- [10] C. Ramadhan, M. A. Senubekti, and D. Amalia, "Penerapan Metodologi Agile dalam Pengembangan Perangkat Lunak," *Router: Jurnal Teknik Informatika dan Terapan*, vol. 3, no. 2, pp. 10–15, Apr. 2025, doi: 10.62951/router.v3i2.411.
- [11] R. T. Aldisa and M. A. Abdullah, "Penerapan Agile Development Methodology dalam Sistem Penjualan Buku dengan Fitur Kategori dan Pencarian," *bits*, vol. 3, no. 4, pp. 547–553, Mar. 2022, doi: 10.47065/bits.v3i4.1434.
- [12] A. R. Ekaputra and A. S. Affandi, "Pemanfaatan layanan cloud computing dan docker container untuk meningkatkan kinerja aplikasi web," *Journal of Information System and Application Development*, vol. 1, no. 2, pp. 141–150, Sep. 2023, doi: 10.26905/jisad.v1i2.11084.
- [13] Y. Fathurrahman, Pizaini, Novriyanto, and T. Darmizal, "IMPLEMENTASI ARSITEKTUR MICROSERVICES DAN TEKNOLOGI DOCKER PADA APLIKASI TRACER ALUMNI," *Jurnal Sistem Informasi dan Teknologi*, vol. 6, no. 2, May 2024, Accessed: Apr. 27, 2026. [Online]. Available: <https://journalversa.com/s/index.php/jsit/article/view/2766>
- [14] M. R. Alamsyah, H. I. Wahanani, and M. Idhom, "PENERAPAN DOCKER UNTUK MEMBANGUN INFRASTRUKTUR PRIVATE CLOUD STORAGE BERBASIS IAAS," *JIFOSI*, vol. 2, no. 2, 2021, doi: 10.33005/jifosi.v2i2.95.
- [15] Miftakhul Ula and Sumiati Sumiati, "Perancangan Sistem Informasi Untuk Pengajuan Judul Tugas Akhir Pada Prodi Teknik Industri Berbasis Web Aplikasi," *jusiik-widyakarya*, vol. 2, no. 1, pp. 180–191, Dec. 2023, doi: 10.59581/jusiik-widyakarya.v2i1.2378.