

**PENGEMBANGAN FITUR MANAJEMEN LABORATORIUM PADA PORTAL
PRODI JURUSAN TEKNIK ELEKTRO MENGGUNAKAN METODE
EXTREME PROGRAMMING UNTUK Mendukung REGULASI
LABORATORIUM UNIVERSITAS LAMPUNG**

(Skripsi)

Oleh

**MUHAMMAD DANU SETA WIARDANA
NPM 2215061085**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2026**

**PENGEMBANGAN FITUR MANAJEMEN LABORATORIUM PADA PORTAL
PRODI JURUSAN TEKNIK ELEKTRO MENGGUNAKAN METODE
EXTREME PROGRAMMING UNTUK Mendukung REGULASI
LABORATORIUM UNIVERSITAS LAMPUNG**

Oleh

MUHAMMAD DANU SETA WIARDANA

Skripsi

**Sebagai Salah Satu Syarat Untuk Mencapai Gelar
SARJANA TEKNIK**

Pada

**Program Studi Teknik Informatika
Fakultas Teknik Universitas Lampung**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2026**

ABSTRAK

PENGEMBANGAN FITUR MANAJEMEN LABORATORIUM PADA PORTAL PRODI JURUSAN TEKNIK ELEKTRO MENGGUNAKAN METODE EXTREME PROGRAMMING UNTUK Mendukung REGULASI LABORATORIUM UNIVERSITAS LAMPUNG

Oleh

MUHAMMAD DANU SETA WIARDANA

Penelitian ini bertujuan mengembangkan fitur manajemen laboratorium pada Portal Program Studi Jurusan Teknik Elektro Universitas Lampung untuk mendukung implementasi regulasi laboratorium secara efisien dan terintegrasi. Metode yang digunakan adalah Extreme Programming (XP). Sebanyak 17 kebutuhan fungsional dan 7 kebutuhan non-fungsional diidentifikasi melalui kuesioner terhadap 123 responden dari lima peran pengguna, yaitu mahasiswa, asisten laboratorium, kepala laboratorium, Pranata Laboratorium Pendidikan (PLP), dan dosen. Seluruh kebutuhan diimplementasikan dalam 12 iterasi pengembangan yang menghasilkan modul inventaris alat, peminjaman dan pengembalian alat, penjadwalan praktikum, pengajuan bebas laboratorium, pengelolaan dokumen dan SOP, pengaduan laboratorium, serta pelaporan aktivitas. Pengujian dilakukan melalui unit testing dengan seluruh skenario uji dinyatakan lulus, User Acceptance Testing (UAT) terhadap 72 responden dengan 663 skenario uji dan tingkat keberhasilan 98,19%, serta evaluasi System Usability Scale (SUS) dengan skor rata-rata 88,08 berkategori Good dan Acceptable dengan nilai Cronbach's Alpha 0,886. Sistem telah berhasil diimplementasikan pada server Universitas Lampung dan telah melalui serangkaian pengujian yang menunjukkan tingkat penerimaan pengguna yang baik.

Kata kunci: *Extreme Programming, Sistem Informasi, Manajemen Laboratorium, Usability Testing, User Acceptance Testing, Universitas Lampung*

ABSTRACT

DEVELOPMENT OF LABORATORY MANAGEMENT FEATURES ON THE ELECTRICAL ENGINEERING DEPARTMENT STUDY PROGRAM PORTAL USING EXTREME PROGRAMMING METHOD TO SUPPORT LABORATORY REGULATIONS AT THE UNIVERSITY OF LAMPUNG

By

MUHAMMAD DANU SETA WIARDANA

This study aims to develop laboratory management features on the Study Program Portal of the Electrical Engineering Department at the University of Lampung to support the implementation of laboratory regulations efficiently and in an integrated manner. The method used is Extreme Programming (XP). A total of 17 functional requirements and 7 non-functional requirements were identified through a questionnaire distributed to 123 respondents from five user roles, namely students, laboratory assistants, laboratory heads, Laboratory Education Technicians (PLP), and lecturers. All requirements were implemented across 12 development iterations, resulting in modules for equipment inventory, equipment borrowing and return, practicum scheduling, laboratory clearance submission, document and SOP management, laboratory complaints, and activity reporting. Testing was conducted through unit testing with all test scenarios passing, User Acceptance Testing (UAT) involving 72 respondents with 663 test scenarios achieving a success rate of 98,19%, and a System Usability Scale (SUS) evaluation with an average score of 88.08 categorized as Good and Acceptable with a Cronbach's Alpha value of 0.886. The system has been successfully deployed on the Universitas Lampung server and has undergone a series of tests demonstrating a good level of user acceptance.

Keywords: Extreme Programming, Information System, Laboratory Management, Usability Testing, User Acceptance Testing, University of Lampung

Judul Skripsi

: **PENGEMBANGAN FITUR MANAJEMEN
LABORATORIUM PADA PORTAL PRODI
JURUSAN TEKNIK ELEKTRO
MENGUNAKAN METODE EXTREME
PROGRAMMING UNTUK Mendukung
REGULASI LABORATORIUM
UNIVERSITAS LAMPUNG**

Nama Mahasiswa

: **Muhammad Danu Setia Wiardana**

Nomor Pokok Mahasiswa

: **2215061085**

Program Studi

: **S1 Teknik Informatika**

Jurusan

: **Teknik Elektro**

Fakultas

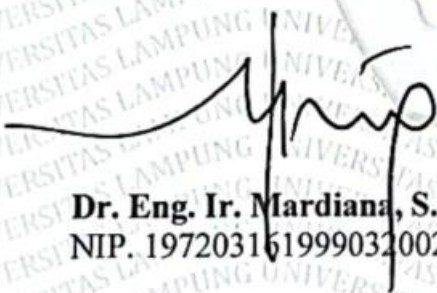
: **Teknik**

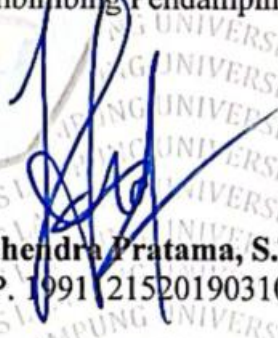
MENYETUJUI

1. Komisi Pembimbing

Pembimbing Utama

Pembimbing Pendamping



Dr. Eng. Ir. Mardiana, S.T., M.T., IPM.
NIP. 197203161999032002



Mahendra Pratama, S.T., M.Eng.
NIP. 199112152019031013

2. Mengetahui

Ketua Jurusan
Teknik Elektro

Ketua Program Studi
Teknik Informatika

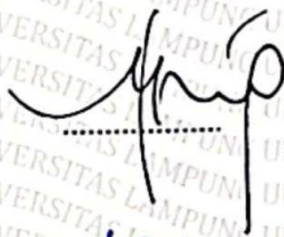

Herlinawati, S.T., M.T.
NIP. 197103141999032001


Yessi Mulyani, S.T., M.T.
NIP. 197312262000122001

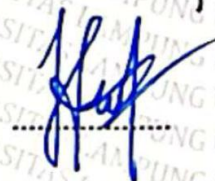
MENGESAHKAN

1. Tim Penguji

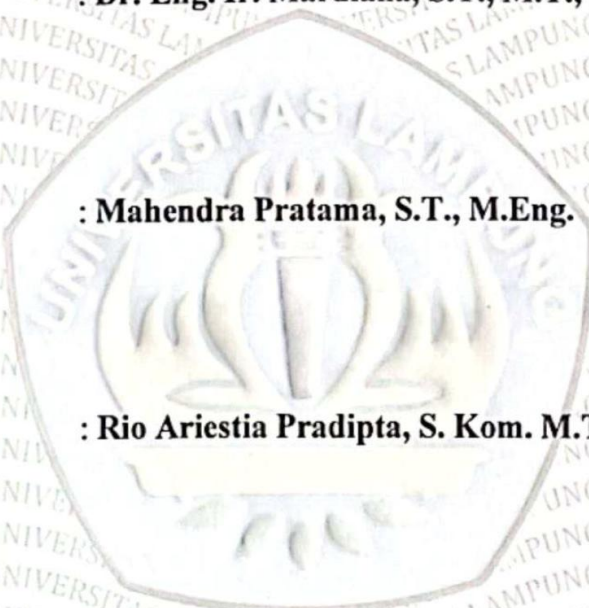
Ketua : Dr. Eng. Ir. Mardiana, S.T., M.T., IPM.



Sekretaris : Mahendra Pratama, S.T., M.Eng.



Penguji : Rio Ariestia Pradipta, S. Kom. M.T.I



2. Dekan Fakultas Teknik



Dr. H. Ahmad Herison, S.T., M.T.

NIP. 196910302000031001

Tanggal Lulus Ujian Skripsi : 13 Mei 2026

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan dengan sebenarnya bahwa skripsi saya yang berjudul “Pengembangan Fitur Manajemen Laboratorium Pada Portal Prodi Jurusan Teknik Elektro Menggunakan Metode Extreme Programming Untuk Mendukung Regulasi Laboratorium Universitas Lampung” merupakan hasil karya saya sendiri. Apabila pernyataan ini tidak benar dan terbukti bahwa skripsi ini merupakan salinan, maka saya bersedia menerima sanksi sesuai dengan ketentuan hukum atau akademik yang berlaku.

Bandar Lampung, 25 Mei 2026
Pembuat Pernyataan,



Muhammad Danu Seta Wiardana
NPM. 2215061085

RIWAYAT HIDUP



Penulis lahir di Bandar Lampung, tanggal 22 Juli 2004 dari orang tua Arifin dan Winahyu Anjani. Penulis telah menyelesaikan pendidikan formal di TK Kartika II-28 pada tahun 2010, SD Kartika II-5 tahun 2016, SMPN 1 Bandar Lampung pada tahun 2019, SMAN 2 Bandar Lampung pada tahun 2022. Pada tahun 2022, penulis diterima sebagai mahasiswa Program Studi Teknik Informatika Universitas Lampung melalui jalur SBMPTN. Selama menempuh pendidikan di Jurusan Teknik Elektro Universitas Lampung, penulis aktif berpartisipasi dalam berbagai kegiatan organisasi dan pengembangan diri. Penulis tergabung dalam Himpunan Mahasiswa Teknik Elektro (HIMATRO) Universitas Lampung, dengan peran sebagai Sekretaris Departemen Sosial dan Kewirausahaan pada periode 2023, serta dipercaya menjadi Kepala Departemen Sosial dan Kewirausahaan pada periode 2024. Selain itu, penulis juga merupakan anggota Generasi Baru Indonesia (GenBI), yaitu organisasi penerima beasiswa Bank Indonesia, dan berkontribusi dalam Divisi Kewirausahaan pada tahun 2024. Dalam bidang akademik, penulis aktif sebagai Asisten Laboratorium Teknik Komputer sejak tahun 2024 hingga 2025, membantu kegiatan praktikum dan pengembangan sistem pembelajaran di laboratorium. Penulis juga mengikuti program Studi Independen Bersertifikat (MSIB) Batch 7 di Dicoding Academy dengan fokus pengembangan *frontend* dan *backend web* pada tahun 2024. Sebagai upaya peningkatan kompetensi profesional, penulis turut mengikuti pelatihan bersertifikat Badan Nasional Sertifikasi Profesi (BNSP) pada bidang Digital Marketing di tahun yang sama.

MOTTO

"I don't have friends, I got family."

Dominic Toretto, Fast & Furious 7

"1327: 1 home, 3 cars, 2 garages, 7 friends"

Fast & Furious

"Youth. Courage. Success."

The Three Pillars of Manchester United

"Never be ashamed of trying. Effortlessness is a myth."

Taylor Swift

"Not today, maybe tomorrow."

Billie Eilish

"Well, maybe I'm not perfect, at least I'm working on it."

Paramore

"Help others, stay curious, work hard."

Muhammad Danu Seta Wiardana

PERSEMBAHAN

Alhamdulillahirobbilalamin

Segala puji bagi Allah SWT atas segala rahmat dan karunia-Nya sehingga skripsi ini dapat diselesaikan. *Shalawat* serta salam senantiasa dilimpahkan kepada Nabi Muhammad SAW.

Kupersembahkan Skripsi Ini Kepada

Kedua orang tuaku tercinta. Semoga Allah senantiasa memberikan perlindungan di dunia dan akhirat, *Aamiin*.

Seluruh dosen dan *civitas* akademik Jurusan Teknik Elektro Universitas Lampung, yang telah *membimbing* dengan ilmu, kesabaran, dan dedikasi tinggi.

Almamaterku tercinta, Universitas Lampung.

SANWACANA

Segala puji bagi Allah SWT atas limpahan rahmat, taufik, dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul “Pengembangan Fitur Manajemen Laboratorium Pada Portal Prodi Jurusan Teknik Elektro Menggunakan Metode Extreme Programming Untuk Mendukung Regulasi Laboratorium Universitas Lampung.” Skripsi ini disusun sebagai salah satu syarat meraih gelar Sarjana Teknik pada Program Studi Teknik Informatika, Universitas Lampung.

Dengan penuh rasa hormat dan penghargaan yang mendalam, penulis menyadari bahwa terselesaikannya karya ini tidak terlepas dari kontribusi, arahan, serta dukungan yang telah diberikan. Oleh karena itu, penulis menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Allah SWT, Tuhan Yang Maha Esa, atas segala limpahan rahmat, karunia, dan kekuatan yang diberikan sehingga skripsi ini dapat terselesaikan dengan baik.
2. Ibu Herlinawati, S.T., M.T., selaku Ketua Jurusan Teknik Elektro Universitas Lampung.
3. Ibu Yessi Mulyani, S.T., M.T., selaku Ketua Program Studi Teknik Informatika yang telah memberikan kelancaran selama proses penyusunan skripsi ini.
4. Ibu Dr. Eng. Ir. Mardiana, S.T., M.T., IPM., selaku Pembimbing Utama yang telah membimbing dan memberikan saran kepada penulis selama proses menyelesaikan penelitian dan penulisan skripsi.
5. Bapak Mahendra Pratama, S.T., M.Eng., selaku Pembimbing Pendamping dan juga Pembimbing Akademik yang telah membimbing, memberikan dukungan dan memberikan saran kepada penulis selama proses menyelesaikan penelitian dan penulisan skripsi.

6. Seluruh Dosen Program Studi Teknik Informatika yang telah memberikan banyak ilmu dan juga nasihat kepada penulis.
7. Kedua orang tua dan seluruh keluarga yang senantiasa memberikan kasih sayang, doa, semangat, serta pengorbanan tanpa henti.
8. Rekan seperjuangan Teknik Elektro Angkatan 2022 (RECTICAL) yang telah menemani dalam perjalanan akademik, baik suka maupun duka.
9. Asisten Laboratorium Teknik Komputer Angkatan 2022, yang telah memberikan dukungan moral, sehingga penulis dapat menyelesaikan penelitian dengan baik.

Penulis memohon maaf atas kekurangan dalam karya ini dan sangat terbuka untuk kritik serta saran membangun demi kesempurnaan di masa mendatang. Semoga skripsi ini bermanfaat untuk pengembangan ilmu dan peningkatan mutu laboratorium di lingkungan pendidikan tinggi.

Bandar Lampung, 25 Mei 2026

Penulis,



Muhammad Danu Seta Wiardana

DAFTAR ISI

	Halaman
DAFTAR ISI.....	iv
DAFTAR TABEL.....	ix
DAFTAR GAMBAR	xi
I. PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	4
1.3 Tujuan Penelitian.....	4
1.4 Manfaat Penelitian.....	5
1.5 Batasan Masalah.....	5
1.6 Sistematika Penulisan.....	6
II. TINJAUAN PUSTAKA.....	7
2.1 Dasar Teori	7
2.1.1 <i>Agile</i>	7
2.1.2 <i>Extreme Programming (XP)</i>	8
2.1.3 Sistem Informasi Manajemen Laboratorium	10
2.1.4 <i>Front End</i>	10
2.1.5 <i>Back End</i>	11
2.1.6 <i>Use Case Diagram</i>	11
2.1.7 <i>Activity Diagram</i>	13
2.1.8 <i>Usability Testing</i>	14
2.1.9 <i>Unit Testing</i>	18
2.1.10 <i>User Acceptance Testing (UAT)</i>	18
2.1.11 ISO 17025:2017	19
2.1.12 Renstra Universitas Lampung 2025-2029.....	21

2.1.13	PermenPANRB No. 7 Tahun 2019 & Permendikbudristek.....	22
2.2	Pengembangan Terdahulu	22
2.2.1	Analisis Pengembangan Terdahulu.....	22
2.2.2	Analisis Gap dan Posisi Penelitian.....	23
2.3	<i>Tools</i> yang Digunakan.....	24
2.3.1	<i>Laravel Framework</i>	24
2.3.2	<i>Application Programming Interface (API)</i>	25
2.3.3	<i>PostgreSQL</i>	25
2.3.4	<i>Draw.io</i>	26
2.3.5	<i>Hypertext Preprocessor (PHP)</i>	26
2.3.6	<i>Figma</i>	27
III.	METODOLOGI PENELITIAN.....	29
3.1	Waktu dan Tempat Penelitian	29
3.2	Alat dan Bahan Penelitian	30
3.2.1	Alat Penelitian.....	30
3.2.2	Bahan Penelitian.....	30
3.3	Metode dan Tahapan Penelitian	31
3.3.1	<i>Planning</i>	34
3.3.2	<i>Design</i>	35
3.3.3	<i>Coding</i>	36
3.3.4	<i>Testing</i>	37
3.3.5	<i>Deployment dan Maintenance</i>	38
IV.	HASIL DAN PEMBAHASAN.....	40
4.1	Pengembangan Portal Prodi Terdahulu	40
4.2	Gambaran Umum Pengembangan Sistem.....	42
4.2.1	Pengumpulan Data Kuesioner.....	43
4.2.2	Proses Analisis Kuesioner dan Identifikasi Kebutuhan	44
4.2.3	Output: Identifikasi Kebutuhan Fungsional dan Non-Fungsional	46
4.2.4	Release Planning dan Iteration Plan.....	50
4.2.5	<i>Schema</i> Basis Data dan <i>Use Case</i> Manajemen Lab	53
4.3	Iterasi 1	55
4.3.1	<i>Planning</i>	55

4.3.2	<i>Design</i>	56
4.3.3	<i>Coding</i>	57
4.3.4	<i>Testing</i>	60
4.4	Iterasi 2	61
4.4.1	<i>Planning</i>	61
4.4.2	<i>Design</i>	62
4.4.3	<i>Coding</i>	63
4.4.4	<i>Testing</i>	67
4.5	Iterasi 3	68
4.5.1	<i>Planning</i>	68
4.5.2	<i>Design</i>	69
4.5.3	<i>Coding</i>	70
4.5.4	<i>Testing</i>	71
4.6	Iterasi 4	72
4.6.1	<i>Planning</i>	72
4.6.2	<i>Design</i>	73
4.6.3	<i>Coding</i>	74
4.6.4	<i>Testing</i>	77
4.7	Iterasi 5	78
4.7.1	<i>Planning</i>	78
4.7.2	<i>Design</i>	79
4.7.3	<i>Coding</i>	80
4.7.4	<i>Testing</i>	82
4.8	Iterasi 6	83
4.8.1	<i>Planning</i>	83
4.8.2	<i>Design</i>	84
4.8.3	<i>Coding</i>	86
4.8.4	<i>Testing</i>	88
4.9	Iterasi 7	89
4.9.1	<i>Planning</i>	89
4.9.2	<i>Design</i>	90
4.9.3	<i>Coding</i>	91

4.9.4	<i>Testing</i>	93
4.10	Iterasi 8	94
4.10.1	<i>Planning</i>	94
4.10.2	<i>Design</i>	95
4.10.3	<i>Coding</i>	97
4.10.4	<i>Testing</i>	99
4.11	Iterasi 9	100
4.11.1	<i>Planning</i>	100
4.11.2	<i>Design</i>	101
4.11.3	<i>Coding</i>	102
4.11.4	<i>Testing</i>	104
4.12	Iterasi 10	105
4.12.1	<i>Planning</i>	105
4.12.2	<i>Design</i>	106
4.12.3	<i>Coding</i>	107
4.12.4	<i>Testing</i>	111
4.13	Iterasi 11	112
4.13.1	<i>Planning</i>	112
4.13.2	<i>Design</i>	114
4.13.3	<i>Coding</i>	115
4.13.4	<i>Testing</i>	119
4.14	Iterasi 12	120
4.14.1	<i>Planning</i>	120
4.14.2	<i>Design</i>	121
4.14.3	<i>Coding</i>	123
4.14.4	<i>Testing</i>	126
4.15	Rangkuman Iterasi Pengembangan.....	127
4.16	<i>User Acceptance Testing</i>	128
4.16.1	Rekap Hasil UAT Keseluruhan.....	132
4.17	Evaluasi <i>Usability</i> dengan <i>System Usability Scale</i>	133
4.17.1	Uji Reliabilitas dengan Metode <i>Cronbach's Alpha</i>	136
4.18	<i>Deployment</i> dan <i>Maintenance</i>	138

V. KESIMPULAN DAN SARAN.....	141
5.1 Kesimpulan.....	141
5.2 Saran.....	142
DAFTAR PUSTAKA	144
LAMPIRAN.....	148
Lampiran A Dokumentasi Iterasi Pengembangan Sistem.....	149
Lampiran B Hasil Survei Analisis Kebutuhan.....	150
Lampiran C <i>Design</i> Tampilan Antarmuka.....	151
Lampiran D Hasil <i>User Acceptance Testing</i> dan <i>System Usability Scale</i>	152
Lampiran E Data dan Hasil <i>Unit Testing</i>	153
Lampiran F Dokumen Penelitian Terdahulu.....	154

DAFTAR TABEL

Tabel	Halaman
Tabel 2.1 Komponen dan Simbol <i>Use Case</i> Diagram	12
Tabel 2.2 Komponen dan Simbol <i>Activity</i> Diagram	13
Tabel 2.3 Komponen dan Simbol <i>Activity</i> Diagram Lanjutan	14
Tabel 3.1 Waktu Pelaksanaan Penelitian	29
Tabel 3.2 Perangkat Keras	30
Tabel 3.3 Perangkat Lunak	30
Tabel 3.4 Ringkasan Metode XP	31
Tabel 3.5 Tahap <i>Planning</i>	35
Tabel 3.6 Tahap <i>Design</i>	36
Tabel 3.7 Tahap <i>Coding</i>	37
Tabel 3.8 Tahap <i>Testing</i>	38
Tabel 4.1 Ringkasan Kebutuhan Fungsional	46
Tabel 4.2 Ringkasan Kebutuhan Fungsional Lanjutan	47
Tabel 4.3 Ringkasan Kebutuhan Non-Fungsional	48
Tabel 4.4 Iterasi XP Plan (dengan 4 Tahapan per Iterasi)	51
Tabel 4.5 Iterasi XP Plan (dengan 4 Tahapan per Iterasi) Lanjutan	52
Tabel 4.6 <i>User Stories</i> Iterasi 1	55
Tabel 4.7 <i>Testing</i> Iterasi ke-1	60
Tabel 4.8 <i>User Stories</i> Iterasi 2	61
Tabel 4.9 <i>Testing</i> Iterasi ke-2	67
Tabel 4.10 <i>User Stories</i> Iterasi 3	68
Tabel 4.11 <i>Testing</i> Iterasi ke-3	71
Tabel 4.12 <i>User Stories</i> Iterasi 4	72
Tabel 4.13 <i>Testing</i> Iterasi ke-4	77
Tabel 4.14 <i>User Stories</i> Iterasi 5	78
Tabel 4.15 <i>Testing</i> Iterasi ke-5	82
Tabel 4.16 <i>User Stories</i> Iterasi 6	83
Tabel 4.17 <i>Testing</i> Iterasi ke-6	88
Tabel 4.18 <i>User Stories</i> Iterasi 7	89
Tabel 4.19 <i>Testing</i> Iterasi ke-7	93
Tabel 4.20 <i>User Stories</i> Iterasi 8	94

Tabel 4.21 <i>Testing</i> Iterasi ke-8.....	99
Tabel 4.22 <i>User Stories</i> Iterasi 9.....	100
Tabel 4.23 <i>Testing</i> Iterasi ke-9.....	104
Tabel 4.24 <i>User Stories</i> Iterasi 10.....	105
Tabel 4.25 <i>Testing</i> Iterasi ke-10.....	111
Tabel 4.26 <i>User Stories</i> Iterasi 11.....	112
Tabel 4.27 <i>User Stories</i> Iterasi 11 Lanjutan	113
Tabel 4.28 <i>Testing</i> Iterasi ke-11.....	119
Tabel 4.29 <i>User Stories</i> Iterasi 12.....	120
Tabel 4.30 <i>Testing</i> Iterasi ke-12.....	126
Tabel 4.31 Rangkuman Iterasi Pengembangan.....	127
Tabel 4.32 Rangkuman Iterasi Pengembangan Lanjutan.....	128
Tabel 4.33 Hasil UAT Mahasiswa.....	129
Tabel 4.34 Hasil UAT Asisten Laboratorium.....	130
Tabel 4.35 Hasil UAT Kepala Laboratorium.....	131
Tabel 4.36 Hasil UAT PLP	132
Tabel 4.37 Hasil UAT Keseluruhan.....	133
Tabel 4.38 Pernyataan Kuesioner SUS	134
Tabel 4.39 Skala Penilaian SUS	134
Tabel 4.40 Distribusi Grade SUS Keseluruhan.....	135
Tabel 4.41 Spesifikasi Server.....	139

DAFTAR GAMBAR

Gambar	Halaman
Gambar 2.1 Fase pada <i>Extreme Programming</i> (XP)	8
Gambar 2.2 Penilaian SUS.....	17
Gambar 3.1 Tahapan Penelitian	32
Gambar 4.1 Antarmuka Dashboard.....	41
Gambar 4.2 Antarmuka Pendaftaran Seminar	42
Gambar 4.3 Use Case Diagram Manajemen Laboratorium	49
Gambar 4.4 <i>Schema</i> Manajemen Laboratorium	53
Gambar 4.5 <i>Activity Diagram</i> Login.....	56
Gambar 4.6 Desain Halaman Daftar Laboratorium	57
Gambar 4.7 Halaman Daftar Laboratorium	57
Gambar 4.8 Kode Halaman Daftar Laboratorium.....	58
Gambar 4.9 <i>Activity Diagram</i> Pengelolaan Inventaris Alat.....	62
Gambar 4.10 Desain Halaman Daftar Inventaris Alat	63
Gambar 4.11 Halaman Inventaris Alat.....	63
Gambar 4.12 Kode Halaman Inventaris Alat.....	65
Gambar 4.13 <i>Activity Diagram</i> Pengembalian Alat.....	69
Gambar 4.14 Desain Halaman Daftar Peminjaman Alat	70
Gambar 4.15 Halaman Daftar Peminjaman Alat	70
Gambar 4.16 <i>Activity Diagram</i> Pengelolaan Data Praktikum	73
Gambar 4.17 Desain Halaman Daftar Praktikum	74
Gambar 4.18 Halaman Daftar Praktikum.....	74
Gambar 4.19 Kode Halaman Daftar Praktikum.....	75
Gambar 4.20 <i>Activity Diagram</i> Pengelolaan Jadwal Praktikum	79
Gambar 4.21 Desain Halaman Tambah Pertemuan Praktikum	80
Gambar 4.22 Halaman Jadwal Pertemuan Praktikum.....	80
Gambar 4.23 Kode Halaman Jadwal Pertemuan Praktikum.....	81
Gambar 4.24 <i>Activity Diagram</i> Pengajuan Bebas Laboratorium.....	84
Gambar 4.25 Desain Halaman Kalender Kegiatan	85
Gambar 4.26 Halaman Kalender Kegiatan	86
Gambar 4.27 Kode Halaman Kalender Kegiatan.....	87
Gambar 4.28 <i>Activity Diagram</i> Pengajuan Bebas Laboratorium.....	90

Gambar 4.29 Desain Halaman Daftar Bebas Laboratorium	91
Gambar 4.30 Halaman Daftar Bebas Laboratorium	91
Gambar 4.31 Kode Halaman Daftar Bebas Laboratorium.....	92
Gambar 4.32 <i>Activity Diagram</i> Cetak Surat Bebas Laboratorium.....	95
Gambar 4.33 Desain Halaman Dokumen dan SOP	96
Gambar 4.34 Surat Bebas Laboratorium.....	97
Gambar 4.35 Halaman Dokumen dan SOP.....	98
Gambar 4.36 Desain Halaman Dokumen dan SOP	101
Gambar 4.37 Desain Halaman Tambah Dokumen dan SOP	101
Gambar 4.38 Halaman Dokumen dan SOP.....	102
Gambar 4.39 Kode Halaman Dokumen dan SOP	103
Gambar 4.40 <i>Activity Diagram</i> Ajukan Pengaduan Laboratorium.....	106
Gambar 4.41 Desain Halaman Ajukan Pengaduan Laboratorium.....	107
Gambar 4.42 Halaman Ajukan Pengaduan Laboratorium	107
Gambar 4.43 Kode Halaman Ajukan Pengaduan Laboratorium	109
Gambar 4.44 <i>Activity Diagram</i> Pendaftaran Kegiatan Akademik.....	114
Gambar 4.45 Desain Halaman Pendaftaran Kegiatan Akademik	115
Gambar 4.46 Halaman Pendaftaran Kegiatan Akademik	115
Gambar 4.47 Kode Halaman Pendaftaran Kegiatan Akademik.....	117
Gambar 4.48 <i>Activity Diagram</i> Laporan Aktivitas	121
Gambar 4.49 Desain Halaman Laporan Aktivitas	122
Gambar 4.50 Halaman Laporan Aktivitas	123
Gambar 4.51 Kode Halaman Laporan Aktivitas.....	124
Gambar 4.52 Hasil Uji Reliabilitas	137
Gambar 4.53 Dashboard Portainer	138
Gambar 4.54 Dashboard Adminer	139

I. PENDAHULUAN

1.1 Latar Belakang

Laboratorium merupakan sarana penunjang utama dalam proses pembelajaran dan penelitian di perguruan tinggi yang memerlukan sistem pengelolaan efisien, akurat, dan terstruktur untuk mendukung kelancaran kegiatan akademik [1]. Dalam pendidikan, keberadaan laboratorium memiliki peran strategis mengingat 70% kegiatan pembelajaran berbentuk praktikum, sehingga pengelolaan yang baik menjadi kunci keberhasilan implementasi Tri Dharma Perguruan Tinggi [2]. Berdasarkan Peraturan Menteri Pendidikan dan Kebudayaan No. 49 Tahun 2014, perguruan tinggi wajib memenuhi delapan standar nasional pendidikan yang mencakup standar sarana dan prasarana pembelajaran, sehingga diperlukan sistem pengelolaan laboratorium yang tidak hanya efisien tetapi juga mampu menyediakan dokumentasi lengkap untuk keperluan akreditasi dan penjaminan mutu [1].

Pentingnya pengelolaan laboratorium yang profesional dan terstandar semakin diperkuat dengan terbitnya Peraturan Menteri Pendayagunaan Aparatur Negara dan Reformasi Birokrasi (PermenPANRB) No. 7 Tahun 2019 tentang Jabatan Fungsional Pranata Laboratorium Pendidikan yang mengatur secara komprehensif ruang lingkup tugas pengelolaan laboratorium meliputi perencanaan, pengoperasian peralatan dan penggunaan bahan, pemeliharaan, pengevaluasian sistem kerja, hingga pengembangan kegiatan laboratorium [3]. Regulasi ini mewajibkan adanya penyusunan Standar Operasional Prosedur (SOP), penerapan Kesehatan dan Keselamatan Kerja (K3), pelaporan kegiatan, serta sistem penilaian kinerja yang akuntabel untuk menjamin mutu laboratorium [3]. Kerangka organisasi pengelolaan laboratorium di Universitas Lampung diatur melalui

Peraturan Menteri Pendidikan, Kebudayaan, Riset, dan Teknologi No. 49 Tahun 2024 tentang Organisasi dan Tata Kerja Universitas Lampung, yang menggariskan laboratorium sebagai perangkat penunjang dengan kepemimpinan kepala laboratorium dan tim Pranata Laboratorium Pendidikan (PLP) [4]. Sejalan dengan kebijakan nasional dan kerangka organisasi tersebut, Universitas Lampung dalam Rencana Strategis (Renstra) 2025-2029 telah menetapkan target strategis penguatan tata kelola laboratorium yang mencakup pengelolaan laboratorium yang adaptif terhadap standar ISO 17025:2017 dengan target 15 laboratorium terakreditasi pada tahun 2029, digitalisasi sistem pengelolaan, serta optimalisasi laboratorium sebagai *revenue generating* unit [5]. Renstra Unila juga menekankan pentingnya penyusunan Peraturan Rektor tentang standar, prosedur, dan kriteria minimal sarana prasarana laboratorium, serta pengangkatan kepala laboratorium pada setiap unit sebagai bagian dari upaya pencapaian akreditasi unggul dan daya saing kelas dunia [5].

Sejalan dengan visi digitalisasi universitas, saat ini Jurusan Teknik Elektro Universitas Lampung telah memiliki sistem informasi berbasis web yang dikenal sebagai "Portal Prodi" yang berfungsi sebagai platform utama pengelolaan administrasi akademik di lingkungan jurusan, mencakup Program Studi S1 Teknik Elektro, S1 Teknik Informatika, dan S2 Teknik Elektro. Portal Prodi ini telah memfasilitasi berbagai layanan akademik seperti pendaftaran seminar (kerja praktik, usul, hasil, komprehensif) dan menjadi pusat data aktivitas mahasiswa. Meskipun Portal Prodi telah terimplementasi dengan baik, fitur khusus untuk manajemen operasional laboratorium yang menjadi sarana vital bagi Jurusan Teknik Elektro masih belum tersedia. Hal ini menyebabkan pengelolaan laboratorium masih berjalan secara manual dan terpisah dari ekosistem digital jurusan, di mana pencatatan peminjaman alat masih menggunakan *logbook* fisik yang rentan rusak atau hilang, serta data inventaris laboratorium tidak terintegrasi dengan data akademik mahasiswa maupun dosen yang sudah ada di Portal Prodi. Akibatnya, sinkronisasi data menjadi sulit dilakukan dan pemantauan aktivitas laboratorium oleh Kepala Laboratorium maupun Ketua Jurusan menjadi tidak efisien dan tidak *real-time*. [6].

Oleh karena itu, Sistem Informasi Manajemen Laboratorium (SIMLAB) dikembangkan untuk meningkatkan efisiensi operasional, keakuratan data, dan analisis berbasis data *real-time* di laboratorium [7]. Penelitian sebelumnya menunjukkan bahwa implementasi sistem informasi dapat menghilangkan kesalahan pencatatan manual, mempercepat proses pembuatan laporan, serta meningkatkan akuntabilitas dan transparansi dalam pengelolaan aset laboratorium. SIMLAB juga dapat membantu dalam pengelolaan inventaris dengan memanfaatkan teknologi seperti QR Code untuk mempermudah identifikasi dan pelacakan peralatan, serta mengimplementasikan *Role-Based Access Control* (RBAC) untuk memastikan setiap pengguna hanya memiliki akses sesuai dengan peran dan tanggung jawabnya [8].

Dengan keterbatasan sumber daya manusia dalam pengelolaan laboratorium, diperlukan sistem yang dapat mengotomatisasi proses-proses administratif sehingga laboran dapat lebih fokus pada tugas-tugas teknis yang memerlukan keahlian khusus [9]. Pemilihan *framework Laravel* dinilai sangat relevan untuk pengembangan sistem informasi laboratorium, karena dukungan arsitektur MVC memudahkan pemisahan logika bisnis pengelolaan data laboratorium (inventaris, jadwal praktikum, peminjaman alat) dari tampilan pengguna serta pengendali akses, pengembangan fitur sistem informasi laboratorium yang terstruktur, aman, dan mudah diintegrasikan dengan data administrasi jurusan.

Berdasarkan hal tersebut, pengembangan Sistem Informasi Manajemen Laboratorium di Jurusan Teknik Elektro Universitas Lampung menjadi kebutuhan mendesak yang harus direalisasikan. Sistem ini diharapkan dapat mengelola berbagai layanan laboratorium seperti praktikum, penelitian, pengabdian masyarakat, dan pengujian, serta menyediakan fitur inventarisasi alat dan bahan, prosedur dan jadwal aktivitas, unduhan dokumen regulasi, dan pengajuan bebas laboratorium. Dengan demikian, sistem dapat mendukung pencapaian visi dan misi jurusan dalam menghasilkan lulusan yang kompeten dan berdaya saing tinggi melalui peningkatan kualitas layanan laboratorium secara menyeluruh.

1.2 Rumusan Masalah

Adapun rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana mengidentifikasi kebutuhan fungsional sistem informasi manajemen laboratorium di Jurusan Teknik Elektro Universitas Lampung?
2. Bagaimana merancang dan membangun sistem informasi manajemen laboratorium yang dapat mengelola inventaris alat dan bahan secara efisien?
3. Bagaimana mengimplementasikan fitur penjadwalan dan monitoring aktivitas laboratorium dalam sistem informasi?
4. Bagaimana membangun mekanisme peminjaman dan pengembalian alat laboratorium yang terintegrasi dalam sistem?
5. Bagaimana melakukan pengujian menyeluruh terhadap sistem yang dikembangkan untuk memastikan kualitas, keamanan, dan *usability*?

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Mengimplementasikan metode *Extreme Programming* (XP) dalam pengembangan sistem informasi manajemen laboratorium pada Portal Prodi Jurusan Teknik Elektro Universitas Lampung melalui penerapan yang sesuai dengan konteks penelitian, seperti iterasi pengembangan, perancangan sederhana, *refactoring*, dan pengujian berkelanjutan, dengan penyesuaian pada penerapan pair programming.
2. Mengidentifikasi kebutuhan fungsional dan non-fungsional sistem informasi manajemen laboratorium di Jurusan Teknik Elektro Universitas Lampung melalui analisis kebutuhan pengguna dengan berbagai peran yaitu mahasiswa, asisten laboratorium, kepala laboratorium, Pranata Laboratorium Pendidikan (PLP), dan dosen.
3. Menguji sistem yang dikembangkan menggunakan *unit testing* pada setiap iterasi pengembangan untuk memastikan seluruh fungsionalitas sistem berjalan sesuai dengan spesifikasi kebutuhan yang telah ditetapkan.
4. Mengevaluasi sistem menggunakan metode UAT untuk memastikan fungsionalitas berjalan dengan baik dan *usability testing* dengan *System*

Usability Scale (SUS) untuk mengukur tingkat kepuasan dan kemudahan penggunaan sistem oleh pengguna.

1.4 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Penelitian ini berkontribusi pada pengembangan sistem informasi manajemen laboratorium yang dapat meningkatkan efisiensi pengelolaan laboratorium di Jurusan Teknik Elektro Universitas Lampung.
2. Hasil penelitian ini dapat menjadi dasar bagi pengembangan lebih lanjut di masa depan.

1.5 Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah:

1. Penelitian dilakukan di lingkup Jurusan Teknik Elektro Universitas Lampung.
2. Sistem yang dikembangkan mencakup modul pengelolaan inventaris alat dan bahan, penjadwalan, peminjaman alat, bebas laboratorium, dan pelaporan.
3. Penelitian tidak mencakup aspek integrasi dengan sistem keuangan atau sistem informasi akademik Universitas.
4. Implementasi sistem dibatasi pada lingkungan laboratorium Jurusan Teknik Elektro.
5. Sistem menerapkan struktur peran yang diturunkan dari *role* dasar masing-masing pengguna, yaitu Asisten Laboratorium berasal dari *role* Mahasiswa, Kepala Laboratorium berasal dari *role* Dosen, dan Pranata Laboratorium Pendidikan (PLP) berasal dari *role* Tenaga Pendidik. Peningkatan peran ini hanya dilakukan di lingkungan internal laboratorium dan tidak mencakup proses kepegawaian atau penetapan jabatan di tingkat fakultas maupun universitas.

1.6 Sistematika Penulisan

Adapun struktur penulisan yang diterapkan pada penelitian ini sebagai berikut:

BAB I: PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan laporan.

BAB II: TINJAUAN PUSTAKA

Bab ini berisi tentang penelitian-penelitian sebelumnya dan teori-teori yang mendukung sebagai referensi dalam penelitian ini.

BAB III: METODOLOGI PENELITIAN

Bab ini berisi tentang waktu dan tempat penelitian, alat dan bahan yang digunakan dalam penelitian, dan prosedur serta analisis kebutuhan perangkat lunak.

BAB IV: HASIL DAN PEMBAHASAN

Bab ini berisi pembahasan mengenai penelitian yang dilakukan.

BAB V: KESIMPULAN DAN SARAN

Bab ini memuat kesimpulan berdasarkan hasil penelitian dan saran yang diharapkan untuk pengembangan penelitian selanjutnya.

DAFTAR PUSTAKA

LAMPIRAN

II. TINJAUAN PUSTAKA

2.1 Dasar Teori

2.1.1 *Agile*

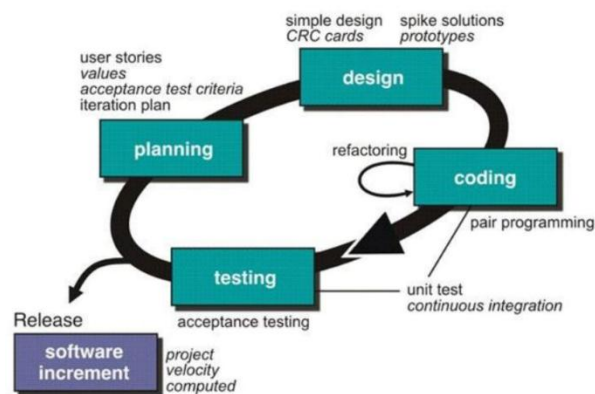
Metodologi *agile* adalah pendekatan pengembangan *software* yang muncul sebagai respons terhadap kebutuhan tim *development* untuk lebih fleksibel dalam menghadapi perubahan kebutuhan *user* yang terus berkembang [10]. Berbagai metodologi *agile* seperti *extreme Programming (XP)*, *Scrum*, *Lean Software Development*, *Feature-Driven Development (FDD)*, dan *Crystal methodologies* dikembangkan untuk memberikan kemampuan kepada tim agar bisa beradaptasi dengan cepat terhadap perubahan *requirement* selama proses *development* [10]. Pendekatan *agile development* telah diperkenalkan sebagai upaya untuk membuat rekayasa perangkat lunak yang fleksibel dan efisien [11]. *Agile software development* merupakan metodologi pengembangan perangkat lunak yang berbasis pada pengembangan iteratif, persyaratan setiap tahapan dan solusi yang ditawarkan berkembang dengan pendekatan kolaborasi antar tim yang terorganisir [11].

Karakteristik utama dari metodologi *agile* adalah penekanan pada kolaborasi tim, iterasi pengembangan yang singkat, keterlibatan *customer* yang aktif dalam proses *development*, dan kesediaan untuk menerima bahwa perubahan adalah bagian normal dari proses pengembangan *software* [10]. Metodologi pengembangan sistem merupakan kerangka yang menjadi pijakan dalam perancangan dan pengembangan perangkat lunak dengan tujuan menghasilkan aplikasi yang sesuai dengan kebutuhan bisnis sebuah organisasi [11].

2.1.2 *Extreme Programming (XP)*

Extreme Programming (XP) merupakan salah satu metodologi pengembangan perangkat lunak berbasis *Agile* yang menekankan pada kualitas kode dan kepuasan pengguna melalui iterasi pengembangan yang cepat dan fleksibel [12]. XP dikembangkan oleh Kent Beck bersama Ward Cunningham dan Ron Jeffries pada tahun 1996 dan berfokus pada pengkodean (*coding*) yang menjadi aktivitas utama dalam semua tahapan pada siklus pengembangan perangkat lunak [11]. Metode XP merupakan metode yang responsif terhadap perubahan dengan iterasi yang bisa dilakukan berulang kali sesuai kebutuhan, serta memiliki kemampuan untuk menyesuaikan kebutuhan pengembangan sistem dengan *requirement* yang berubah dengan perubahan *requirement* yang sangat cepat [11].

XP memiliki lima nilai utama yaitu *communication*, *simplicity*, *feedback*, *courage*, dan *respect* yang menjadi fondasi dalam implementasinya [12]. Tahapan pengembangan perangkat lunak dengan XP meliputi empat fase utama yaitu *planning* (perencanaan), *design* (perancangan), *coding* (pengkodean) dan *testing* (pengujian) [11]. Pada tahap perencanaan dimulai dengan pemahaman konteks bisnis dari aplikasi dan mendefinisikan fungsionalitas keseluruhan yang akan dikembangkan. Tahap perancangan fokus pada *design* aplikasi secara sederhana dengan menggunakan CRC (*Class Responsibility Collaborator*). Tahap pengkodean merupakan penerjemahan dari perancangan dalam bahasa pemrograman, dan tahap *testing* dilakukan untuk menguji sistem yang telah dibangun agar dapat menemukan kesalahan-kesalahan [11].



Gambar 2.1 Fase pada *Extreme Programming (XP)*

Sumber : Pengembangan Sistem *Extreme Programming* [11]

Gambar 2.1 menunjukkan empat fase utama dalam siklus *Extreme Programming* (XP) yang berjalan secara iteratif dan siklis. Fase pertama adalah *Planning*, di mana tim melakukan identifikasi *user stories* dari kebutuhan aplikasi, mendefinisikan *values* yang menjadi pengembangan, menetapkan *acceptance test criteria* untuk setiap *user story* agar jelas kapan suatu fitur dianggap selesai, dan membuat *iteration plan* yang mengorganisir *user stories* menjadi beberapa iterasi pengembangan. Dari fase *Planning*, sistem masuk ke fase *Design*, di mana tim melakukan perancangan aplikasi dengan prinsip *simple design* (tidak membuat desain yang berlebihan), mengidentifikasi *spike solutions* untuk mengatasi *technical challenges* yang mungkin muncul, membuat CRC (*Class Responsibility Collaborator*) cards untuk mendefinisikan *class* dan *responsibility* dalam domain *logic*, serta membuat *prototypes* untuk memvalidasi desain sebelum implementation. Setelah *design* selesai, sistem masuk ke fase *Coding*, di mana implementasi kode dilakukan berdasarkan *design* yang telah dibuat, *refactoring* dilakukan secara berkelanjutan untuk meningkatkan kualitas dan kebersihan kode agar mudah diperbaiki, dan *pair programming* dipraktikkan di mana dua developer bekerja sama pada satu komputer untuk meningkatkan kualitas dan *knowledge sharing*. Hasil dari *coding phase* kemudian dimasukkan ke fase *Testing*, di mana *unit testing* dilakukan untuk memastikan setiap unit kode berfungsi dengan benar dan menghasilkan hasil yang diharapkan, *acceptance testing* dilakukan untuk memverifikasi bahwa implementasi memenuhi *acceptance criteria* yang telah ditetapkan di phase *Planning*, dan *continuous integration* memastikan bahwa code yang di-*commit* langsung terintegrasi dan di-*test* secara otomatis sehingga *issue* dapat dideteksi lebih awal. Setelah keempat fase ini selesai, sistem menghasilkan *software increment* (rilis) yang berisi fitur-fitur yang telah dikerjakan dalam iterasi tersebut. Siklus ini kemudian berulang kembali ke fase *Planning* untuk iterasi berikutnya, dengan *user stories* baru atau kelanjutan dari *stories* sebelumnya, sehingga sistem terus berkembang secara incremental dengan *feedback* di setiap akhir iterasi.

2.1.3 Sistem Informasi Manajemen Laboratorium

Sistem Informasi Manajemen Laboratorium (*Laboratory Information Management System/LIMS*) adalah alat yang dirancang untuk meningkatkan efisiensi laboratorium modern melalui pengelolaan informasi, data sampel, dan proses kerja secara terintegrasi [7]. LIMS merupakan sistem komputerisasi yang mengumpulkan, *memproses*, dan menyimpan informasi yang dihasilkan laboratorium [7].

Implementasi SIMLAB dapat meningkatkan akurasi data, mengurangi waktu pencatatan, memperbaiki alur kerja laboratorium, serta membuat manajemen data lebih efisien dan efektif, informasi lebih akurat dan cepat, data lebih aman, mempermudah pengintegrasian data, dan meningkatkan akuntabilitas serta transparansi [13]. Sistem ini dirancang untuk mengotomatisasi proses administratif laboratorium sehingga staf dapat lebih fokus pada kegiatan teknis dan ilmiah, sekaligus membantu dalam pemenuhan standar kualitas dan regulasi laboratorium

2.1.4 *Front End*

Frontend development merupakan bagian penting dalam pengembangan aplikasi *web* modern yang berfokus pada penyediaan pengalaman pengguna yang optimal serta memastikan interaksi yang lancar dengan sistem. Seorang *frontend developer* bertugas menerjemahkan rancangan visual menjadi antarmuka yang interaktif dan berfungsi sebagaimana mestinya. Teknologi utama yang digunakan meliputi *HyperText Markup Language* (HTML), *Cascading Style Sheets* (CSS), dan *JavaScript*, yang memungkinkan tampilan *web* bersifat dinamis, menarik, serta mudah di akses pada berbagai perangkat dan ukuran layar. *Frontend* adalah proses pengembangan *website* yang berjalan di sisi pengguna (*client-side*) menggunakan HTML, CSS, dan *JavaScript/jQuery*, di mana interaksi langsung terjadi di browser tanpa campur tangan server [14].

Aspek responsivitas dan performa menjadi prioritas utama dalam pengembangan *frontend*. Pengembang harus memastikan bahwa aplikasi *web* dapat beradaptasi dengan berbagai resolusi layar tanpa mengurangi kenyamanan dan

kecepatan akses. Untuk meningkatkan performa, digunakan berbagai teknik *optimisasi* seperti *lazy loading* dan *code splitting* yang membantu mempercepat waktu pemuatan halaman. Selain itu, penerapan desain responsif dan adaptif telah menjadi standar pengembangan *web* modern untuk menjaga konsistensi tampilan serta fungsionalitas antarmuka di berbagai perangkat.

2.1.5 *Back End*

Backend development berfokus pada pengelolaan logika server, basis data, serta infrastruktur yang menunjang kinerja aplikasi *web*. Seorang *backend developer* bertanggung jawab memastikan keamanan, skalabilitas, dan keandalan sistem melalui penggunaan bahasa pemrograman sisi server dan sistem manajemen basis data. Aspek keamanan menjadi prioritas utama dengan penerapan praktik seperti enkripsi data, *autentikasi* yang kuat, serta pencegahan serangan *SQL injection* dan *cross-site scripting (XSS)*.




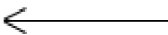
Selain keamanan, skalabilitas dan efisiensi juga menjadi fokus dalam pengembangan *backend*. Sistem harus mampu menangani peningkatan beban dan pertumbuhan data tanpa mengurangi performa. Teknik seperti *caching*, *load balancing*, dan optimisasi basis data digunakan untuk mempercepat akses serta mengurangi beban server. Manajemen API (*Application Programming Interface*) menjadi komponen penting untuk menghubungkan *frontend* dan *backend* secara aman dan efisien. *Backend* meliputi bagian aplikasi *web* yang berjalan di server (*server-side*), bertugas menangani pemrosesan data, penyimpanan *database*, serta logika bisnis yang hasilnya dikirim kepada *frontend* melalui respons server [14].

2.1.6 *Use Case Diagram*

Use case diagram merupakan sebuah diagram yang menunjukkan hubungan antara aktor dan *use case* dalam sebuah sistem. Diagram ini digunakan untuk analisis dan desain sistem, menggambarkan interaksi antara aktor (baik manusia maupun organisasi) dengan sistem yang dikembangkan. Dalam *use case diagram*, digambarkan elemen-elemen utama yaitu *actor*, *use case*, serta *relationship* di

antara keduanya yang terdiri dari asosiasi, generalisasi, dan dependensi. *Use case* digambarkan dalam bentuk elips horizontal, sementara aktor direpresentasikan sebagai peran yang berinteraksi serta *saling* bertukar informasi dengan sistem. Hubungan yang tercipta dapat berupa asosiasi langsung antara aktor dan *use case*, asosiasi antar *use case*, serta generalisasi baik antar aktor maupun *use case* [15]. Berikut ini adalah tabel simbol yang digunakan dalam *Use Case Diagram* (Tabel 2.1).

Tabel 2.1 Komponen dan Simbol *Use Case Diagram*


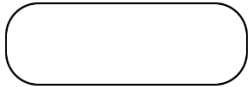
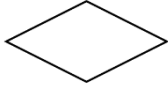


Nama	Fungsi	Simbol
<i>System Boundary</i>	Batas sistem yang menunjukkan sejauh mana sistem berinteraksi dengan aktor	
<i>Use Case</i>	Fungsionalitas atau layanan utama yang disediakan sistem untuk aktor.	
Aktor	Peran yang berinteraksi dengan sistem, berupa manusia, sistem lain, atau perangkat	
Generalisasi	Hubungan hierarki antar aktor atau <i>use case</i> yang menunjukkan pewarisan sifat atau perilaku.	-----
Asosiasi	Hubungan langsung antara aktor dan <i>use case</i> yang menunjukkan adanya interaksi.	
<i>Include</i>	Hubungan di mana satu <i>use case</i> selalu memanggil <i>use case</i> lain sebagai bagian dari prosesnya.	<<include>> ----->
<i>Extend</i>	Menunjukkan bahwa suatu <i>use case</i> dapat diperluas dengan fungsionalitas <i>use case</i> lain.	<<extend>> ----->

2.1.7 Activity Diagram



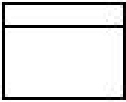
Activity diagram adalah diagram UML yang menggambarkan konsep aliran data, kontrol, dan aksi-aksi terstruktur yang didesain dengan baik dalam suatu sistem. Diagram ini merepresentasikan urutan aktivitas, baik secara linier maupun paralel, dalam suatu proses sistem. Komponen-komponen utama *activity* diagram meliputi *activity node* (menunjukkan proses-proses utama), *activity edge* (menghubungkan aktivitas), *initial state* (simbol awal proses), *decision* (pengambilan keputusan), *fork* dan *join* (pengelolaan proses paralel), serta *final state* (akhir proses) [15].

Activity diagram dijelaskan sebagai alat untuk menjelaskan urutan kegiatan dalam suatu proses pada sistem, baik secara berurutan maupun paralel. Berikut ini adalah tabel simbol yang digunakan dalam *activity* diagram (Tabel 2.2).

Tabel 2.2 Komponen dan Simbol *Activity* Diagram

Nama	Fungsi	Simbol
Status Awal (<i>Initial State</i>)	Titik awal dimulainya proses atau aktivitas dalam sistem.	
Aktivitas (<i>Activity</i>)	Tindakan atau proses yang dilakukan dalam sistem.	
Percabangan (<i>Decision</i>)	Titik pengambilan keputusan yang menentukan alur proses berdasarkan kondisi tertentu.	
Arah (<i>Action Flow</i>)	Arah atau urutan perpindahan dari satu aktivitas ke aktivitas lain.	
Penggabungan (<i>Merge</i>)	Titik di mana beberapa alur bercabang digabung kembali menjadi satu.	

Tabel 2.3 Komponen dan Simbol *Activity Diagram* Lanjutan

Pemisahan (<i>Fork</i>)	Titik di mana satu alur proses dibagi menjadi beberapa aktivitas yang berjalan secara paralel.	
Status Akhir (<i>Final State</i>)	Titik berakhirnya seluruh proses dalam diagram aktivitas.	
<i>Swimlane</i>	Pembagian area dalam diagram untuk menunjukkan siapa (aktor atau bagian) yang bertanggung jawab terhadap setiap aktivitas.	

2.1.8 *Usability Testing*

Usability testing adalah suatu metode pengujian yang krusial dalam proses rekayasa perangkat digital, bertujuan mengidentifikasi kemudahan, efektivitas, dan kepuasan pengguna ketika berinteraksi dengan suatu produk [16]. Pengujian ini dapat dirancang dengan berbagai mode, yaitu pada sesi terkontrol (*moderated*), tes dipandu oleh seorang fasilitator yang mengarahkan responden secara langsung, sedangkan sesi tak terkontrol (*unmoderated*) membebaskan responden menjawab tanpa pengawasan, cukup mengikuti instruksi tertulis. Bahkan, skema pengujian dapat dilakukan secara daring (*remote*) melalui platform digital, atau bertatap muka (*in-person*), tergantung pada kebutuhan dan konteks pengembangan produk. Tujuan pelaksanaan *usability testing* pun sangat fleksibel, di tahap awal, sering dilakukan pengujian eksplorasi untuk menggali kebutuhan dan perilaku pengguna, ketika produk sudah lebih matang, tes penilaian (*assessment*) digunakan mengecek performa sistem yang telah berjalan. Sementara, untuk membedakan produk/desain yang bersaing, pengujian komparasi dimanfaatkan agar pengembang memperoleh data preferensi dan kemudahan penggunaan secara empiris. Prinsip utama dalam pengujian *usability* meliputi *learnability* (kemudahan pengguna dalam pertama kali menggunakan aplikasi), *efficiency* (kecepatan pengguna mencapai tujuan), *memorability* (kemudahan ketika pengguna kembali menggunakan aplikasi setelah

jeda), *error* (kesalahan yang dilakukan pengguna), dan *satisfaction* (kepuasan pengguna terhadap aplikasi). Pengujian dapat dilakukan dengan pendekatan *expert-based*, *automated*, atau *user-based testing* (melibatkan pengguna nyata menyelesaikan skenario tugas). Agar hasilnya komprehensif dan objektif, *usability testing* biasanya menggunakan serangkaian tugas yang harus diselesaikan oleh pengguna untuk mengamati pengalaman serta hambatan yang dialami. Indikator kuantitatif yang biasa dipakai untuk menganalisis hasil *usability testing* meliputi persentase keberhasilan tugas (*Task Success Rate*), waktu penyelesaian tugas (*Task Completion Time*), jumlah kesalahan saat menjalankan tugas (*Error Rate*), serta penilaian terhadap kepuasan pengguna (*user satisfaction*). Secara kuantitatif, hasil pengujian *usability* dianalisis menggunakan sejumlah metrik yang telah baku secara internasional. *Task Success Rate* dihitung dengan persamaan:

$$\text{Success Rate} = \frac{(S + (PS \times 0,5))}{\text{Total Task}} \times 100\%$$

S adalah jumlah tugas yang sempurna, PS adalah tugas yang separuh berhasil, semua dihitung relatif terhadap total tugas [17]. *Time Based Efficiency* (TBE) merupakan tingkat kecepatan pengguna menggunakan aplikasi saat menyelesaikan tugas, dapat diukur melalui:

$$\text{Time Based Efficiency} = \frac{\sum_{j=1}^R \sum_{i=1}^N \frac{n_{ij}}{t_{ij}}}{N \times R}$$

R adalah jumlah responden, dan N adalah jumlah tugas. Nilai n_{ij} menandakan keberhasilan tugas, apabila individu berhasil menyelesaikan tugas tertentu maka nilainya 1, kalau gagal atau tidak selesai, nilainya otomatis 0. Sementara itu, t_{ij} adalah waktu yang diperlukan oleh masing-masing pengguna untuk menyelesaikan tugas spesifik. Jika tugasnya tidak selesai, waktu tetap dihitung sampai pengguna memutuskan berhenti mencoba atau menyerah di tengah jalan. *Error Rate* sebagai indikator kualitas diperoleh dengan membagi total kesalahan yang terjadi dengan peluang keseluruhan *error*, dihitung dengan persamaan:

$$\text{Error Rate} = \frac{\text{Total Defects}}{\text{Total Opportunities}}$$

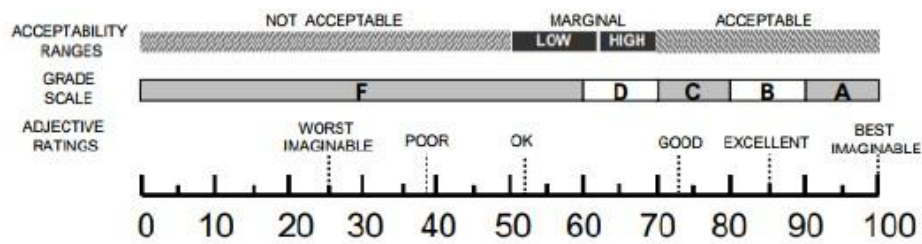
Total *Defects* adalah jumlah kesalahan atau *error* yang dicatat selama proses pengujian, sedangkan Total *Opportunities* adalah jumlah maksimum kesempatan terjadinya kesalahan, biasanya dihitung dari banyaknya titik rawan *error* dalam suatu tugas dikalikan dengan total peserta yang terlibat. *System Usability Scale* (SUS) adalah kuesioner sederhana yang dirancang untuk mengukur tingkat kemudahan dan kualitas suatu sistem, aplikasi, atau produk digital dari sudut pandang pengguna [17]. SUS terdiri dari sepuluh pertanyaan dengan skala *Likert* lima poin (sangat setuju–sangat tidak setuju) yang mencakup aspek-aspek fungsionalitas, kompleksitas, integrasi fitur, hingga kenyamanan dan rasa percaya diri pengguna saat menggunakan sistem. Pertanyaan bernomor ganjil berisi pernyataan positif, sedangkan item genap adalah pernyataan negatif, sehingga skema skoringnya berbeda, yaitu skor jawaban positif (ganjil) dikurangi satu, dan skor jawaban negatif (genap) adalah lima dikurangi skor responden. Nilai total dari semua tanggapan pada sepuluh pernyataan dijumlahkan dan dikalikan 2,5, sehingga didapat skor akhir SUS dengan rentang 0–100. Persamaan Skor Responden dapat dilihat pada persamaan berikut ini:

$$\text{Skor } R = ((P1 - 1) + (5 - P2) + (P3 - 1) + (5 - P4) + (P5 - 1) + (5 - P6) + (P7 - 1) + (5 - P8) + (P9 - 1) + (5 - P10)) * 2,5$$

Skor R adalah nilai SUS tiap responden, P1...P10 adalah nilai skala yang telah diberikan oleh responden pada tiap pernyataan. Setelah itu, persamaan Skor rerata SUS dapat dilihat pada berikut ini:

$$\text{Skor Rerata SUS} = \sum_{i=1}^n \frac{x_i}{n}$$

X_i adalah jumlah nilai skor dari responden, dan n adalah jumlah responden. Nilai SUS kemudian diinterpretasi secara bertingkat, melalui tiga kategori, yaitu tingkat *acceptability* (*not acceptable*, *marginal*, *acceptable*), *grade scale* (A sampai F), serta *adjective rating* (*worst imaginable*, *poor*, *OK*, *good*, *best imaginable*). SUS mudah diterapkan karena fleksibel untuk aplikasi apa saja dan sudah tervalidasi oleh banyak penelitian *usability*, sehingga sering dipilih sebagai standar analisis kepuasan pengguna [17].



Gambar 2.2 Penilaian SUS

Sumber : Artikel Evaluasi *Usability* [17]

Dalam pengukuran menggunakan kuesioner seperti SUS, validitas dan reliabilitas instrumen sangat penting untuk memastikan data yang diperoleh akurat dan konsisten. Reliabilitas mengacu pada sejauh mana hasil pengukuran tetap konsisten apabila dilakukan pengukuran ulang terhadap subjek yang sama. Metode yang paling umum digunakan untuk mengukur reliabilitas kuesioner dengan skala Likert adalah koefisien *Cronbach's Alpha* [18]. Nilai *Cronbach's Alpha* berkisar antara 0 hingga 1, di mana nilai yang semakin mendekati 1 menunjukkan tingkat reliabilitas yang semakin tinggi. Instrumen penelitian dikatakan reliabel jika memiliki koefisien *Cronbach's Alpha* lebih besar dari 0,60 ($> 0,60$). Rumus perhitungan *Cronbach's Alpha* adalah sebagai berikut:

$$r_1 = \left(\frac{k}{k-1} \right) \left(1 - \frac{\sum s_i^2}{s_t^2} \right)$$

Pada rumus tersebut dapat dilihat bahwa r_{11} merupakan koefisien reliabilitas, n adalah banyaknya soal, S_i^2 adalah variasi skor soal ke i dan S_t^2 adalah variasi skor total. Suatu variabel dikatakan reliabel, apabila hasil $\alpha > 0,60$, sedangkan jika hasil $\alpha < 0,60$ maka variabel tersebut dikatakan tidak reliabel [18]. Untuk mempermudah analisis data kuesioner SUS dalam penelitian ini, perhitungan nilai reliabilitas *Cronbach's Alpha* akan dilakukan menggunakan bantuan perangkat lunak statistik SPSS (*Statistical Package for the Social Sciences*).

2.1.9 *Unit Testing*

Unit testing adalah metode pengujian perangkat lunak di mana unit terkecil dari kode program diuji secara terisolasi untuk memastikan bahwa unit tersebut bekerja sesuai dengan yang diharapkan. Tujuan utama *unit testing* adalah untuk memverifikasi kebenaran logika kode pada level terendah sebelum kode tersebut diintegrasikan dengan bagian lain dari sistem. *Unit testing* juga memberikan manfaat jangka panjang berupa perlindungan terhadap regresi (*regression protection*), ketahanan terhadap *refactoring* (*resistance to refactoring*), dan umpan balik yang cepat (*fast feedback*) [19].

Unit Testing yang baik harus memiliki empat pilar utama, yaitu perlindungan terhadap regresi, artinya tes harus mampu mendeteksi *bug* yang muncul akibat perubahan kode, ketahanan terhadap *refactoring*, artinya tes tidak boleh gagal hanya karena struktur internal kode diubah selama perilaku eksternalnya tetap sama, umpan balik yang cepat, artinya tes harus dieksekusi dengan cepat agar pengembang tidak perlu menunggu lama untuk mengetahui hasil perubahan kode; dan (4) *maintainability*, artinya tes harus mudah dibaca dan dipelihara [19]. *Unit Testing* sering dilakukan secara otomatis menggunakan *framework* xUnit (seperti PHPUnit untuk PHP) dan mengikuti pola AAA (*Arrange, Act, Assert*) di mana setiap tes disusun dengan mempersiapkan kondisi awal (*Arrange*), mengeksekusi tindakan yang diuji (*Act*), dan memverifikasi hasil yang diharapkan (*Assert*).

2.1.10 *User Acceptance Testing (UAT)*

User Acceptance Testing (UAT) adalah fase pengujian formal yang dilakukan untuk menentukan apakah sistem memenuhi kebutuhan bisnis dan kriteria penerimaan yang telah ditetapkan, sehingga dapat diterima oleh pengguna. Berbeda dengan *unit testing* yang berfokus pada spesifikasi teknis dan deteksi bug perangkat lunak, UAT berfokus pada validasi apakah sistem mampu mendukung proses bisnis operasional dalam skenario dunia nyata (*real-world scenarios*) [20]. UAT menjawab pertanyaan mendasar: "Apakah sistem ini layak untuk digunakan (*fit for purpose*)?" bukan sekadar "Apakah sistem ini bekerja sesuai spesifikasi?" .

UAT memiliki peran strategis sebagai gerbang terakhir (*final gateway*) sebelum sistem dirilis ke lingkungan produksi. UAT berfungsi sebagai manajemen risiko untuk menghindari kegagalan sistem yang mahal setelah peluncuran, membangun kepercayaan pengguna (*confidence building*), memvalidasi kesiapan proses bisnis, dan menilai kesiapan sistem untuk layanan operasional. Proses UAT melibatkan pengujian fungsionalitas sistem proses bisnis *end-to-end*, di mana pengguna akhir (*end-users*) menjalankan skenario nyata untuk memastikan sistem memberikan output yang diharapkan tanpa kendala [20].

UAT didasarkan pada *acceptance criteria* (kriteria penerimaan) yang didefinisikan dari kebutuhan bisnis (*business requirements*). Kriteria ini menjadi tolak ukur objektif untuk keputusan *Go/No-Go* dalam pelepasan sistem. Jika *unit testing* menjamin kode dibangun dengan benar (*built the code right*), maka UAT menjamin bahwa sistem yang dibangun adalah sistem yang benar sesuai kebutuhan pengguna (*built the right system*) [20].

2.1.11 ISO 17025:2017

ISO/IEC 17025:2017 adalah standar internasional yang menetapkan persyaratan umum kompetensi untuk laboratorium pengujian dan kalibrasi [21]. Meskipun standar ini dirancang khusus untuk laboratorium pengujian dan kalibrasi komersial, prinsip-prinsip fundamental yang terkandung di dalamnya dapat diadaptasi dan diterapkan pada laboratorium pendidikan untuk meningkatkan kualitas pengelolaan dan mendukung proses akreditasi program studi. Standar ini dirancang untuk mempromosikan kepercayaan terhadap hasil laboratorium dengan mengatur berbagai aspek penting. ISO 17025 menuntut laboratorium untuk memastikan ketidakberpihakan dalam setiap kegiatan, menjaga kerahasiaan informasi yang diperoleh, dan mendokumentasikan seluruh proses secara sistematis. Standar ini juga mewajibkan laboratorium untuk merencanakan tindakan memitigasi risiko dan memanfaatkan peluang untuk peningkatan mutu secara berkelanjutan [21].

Dalam laboratorium pendidikan di Jurusan Teknik Elektro, beberapa prinsip dan komponen ISO 17025 yang relevan untuk diterapkan dalam sistem informasi manajemen laboratorium meliputi:

Klausul 4: Ketidakberpihakan dan Kerahasiaan. Laboratorium pendidikan harus menjaga objektivitas dalam setiap kegiatan dan memastikan tidak ada konflik kepentingan yang memengaruhi operasional. Sistem informasi harus mendukung hal ini dengan menyediakan audit trail yang transparan untuk setiap transaksi, implementasi *role-based access control* untuk mencegah penyalahgunaan wewenang, dan proteksi data pribadi pengguna sesuai prinsip kerahasiaan. Dalam implementasi sistem, ini diwujudkan melalui *logging* otomatis untuk setiap aktivitas kritis seperti peminjaman alat, perubahan data inventaris, dan persetujuan kegiatan laboratorium.

Klausul 5: Persyaratan Struktur. Laboratorium harus memiliki struktur organisasi yang jelas dengan pembagian tanggung jawab dan wewenang yang terdefinisi. Sistem informasi mendukung ini dengan mengimplementasikan hierarki *user roles* yang mencerminkan struktur organisasi laboratorium, seperti mahasiswa, asisten laboratorium, dosen, kepala laboratorium, dan PLP. Setiap *role* memiliki akses dan tanggung jawab yang berbeda sesuai dengan tugas dan fungsinya dalam organisasi laboratorium. Sistem juga menyediakan fitur untuk mendokumentasikan struktur organisasi dan deskripsi setiap posisi.

Klausul 6.4: Persyaratan Sumber Daya atau Peralatan. Standar mensyaratkan laboratorium untuk mengelola peralatan dengan baik termasuk, pemeliharaan, dan dokumentasi lengkap. Sistem informasi mengimplementasikan ini melalui modul inventaris dengan fitur QR Code untuk identifikasi cepat, jadwal perbaikan, riwayat perbaikan dan, serta pelacakan lokasi dan penggunaan alat.

Klausul 7.4: Penanganan Barang. Laboratorium harus memiliki prosedur untuk penanganan, penyimpanan, dan identifikasi barang. Sistem informasi harus terdapat fitur *tracking* penyimpanan alat, kondisi penyimpanan, dan status ketersediaan alat. Sistem juga mencatat instruksi untuk alat-alat yang memerlukan perlakuan khusus.

Klausul 7.5: Catatan Teknis. Sistem mencatat setiap praktikum atau aktivitas laboratorium dengan tanggal, waktu, mahasiswa yang terlibat, alat dan bahan yang digunakan, hasil. Sistem memastikan data tidak bisa dihapus bisa diarsip dengan *backup* untuk memastikan data aman dan dapat diakses di masa depan.

Klausul 7.8: Pelaporan Hasil. Laboratorium harus memiliki format standar untuk pelaporan kegiatan. Sistem informasi menyediakan *template* laporan untuk berbagai keperluan seperti laporan praktikum, laporan peminjaman alat, laporan *maintenance*. Sistem juga menyediakan fitur export data untuk memudahkan pembuatan laporan manajemen.

Klausul 7.9: Pengaduan. Laboratorium harus memiliki mekanisme penanganan pengaduan. Sistem informasi menyediakan modul *ticketing* untuk pengaduan atau keluhan terkait fasilitas laboratorium, kerusakan alat, atau layanan laboratorium. Setiap pengaduan tercatat dengan baik dan dipantau hingga terselesaikan dengan dokumentasi lengkap mengenai tindakan yang diambil.

Klausul 8.3: Pengendalian Dokumen. Laboratorium harus mengendalikan semua dokumen yang berkaitan dengan operasional. Sistem informasi menyediakan *repository* untuk menyimpan dan mengelola dokumen seperti SOP praktikum, panduan keselamatan kerja, instruksi kerja alat, dan formulir-formulir standar.

2.1.12 Renstra Universitas Lampung 2025-2029

Renstra Unila 2025-2029 adalah dokumen perencanaan strategis yang memuat arah kebijakan, tujuan, dan target pengembangan universitas selama lima tahun mendatang. Dalam kerangka Renstra, penguatan tata kelola laboratorium menjadi salah satu prioritas, dengan visi mencapai laboratorium berstandar internasional melalui akreditasi ISO 17025, digitalisasi pengelolaan, dan optimalisasi sarana untuk mendukung pembelajaran berbasis *project* dan *case-method*. Renstra juga mensyaratkan penyusunan SOP, pemenuhan standar minimal sarana dan prasarana, dan pengangkatan kepala laboratorium yang kompeten sebagai bagian dari upaya pencapaian akreditasi unggul dan daya saing kelas dunia [5]. Oleh karena itu, Renstra menjadi acuan utama dalam implementasi sistem manajemen laboratorium yang modern, adaptif, dan terintegrasi, serta memastikan laboratorium mampu memberikan kontribusi signifikan terhadap pencapaian indikator kinerja utama universitas.

2.1.13 PermenPANRB No. 7 Tahun 2019 & Permendikbudristek No. 49 Tahun 2024

Peraturan Menteri Pendayagunaan Aparatur Negara dan Reformasi Birokrasi (PermenPANRB) No. 7 Tahun 2019 mengatur Jabatan Fungsional Pranata Laboratorium Pendidikan (PLP) yang bertanggung jawab sebagai pelaksana teknis pengelolaan laboratorium pendidikan. Regulasi ini mewajibkan laboratorium untuk memiliki SOP, menerapkan Kesehatan dan Keselamatan Kerja (K3), serta dokumentasi aktivitas dan penilaian kinerja untuk menjamin mutu laboratorium. Selain itu, Permendikbudristek No. 49 Tahun 2024 mengatur organisasi, struktur, dan tata kelola laboratorium di perguruan tinggi, termasuk pembentukan unit laboratorium, pengangkatan kepala laboratorium sesuai bidang ilmu, dan integrasi laboratorium dengan Unit Penunjang Akademik [3]. Regulasi ini memperkuat landasan hukum dan organisasi laboratorium, sehingga pengelolaan dapat berjalan sesuai standar nasional maupun internasional, selaras dengan misi universitas dalam Renstra.

2.2 Pengembangan Terdahulu

2.2.1 Analisis Pengembangan Terdahulu

Berdasarkan hasil telaah terhadap berbagai penelitian terdahulu, ditemukan sejumlah karya yang relevan dengan pengembangan sistem informasi laboratorium. Purwanto *et al.* (2022) mengembangkan Sistem Informasi Manajemen Pembelajaran untuk kegiatan praktikum laboratorium melalui forum diskusi dan studi literatur. Sistem tersebut membantu pembelajaran mandiri dan pengelolaan fasilitas laboratorium, namun masih berfokus pada aspek pembelajaran dan belum mencakup inventaris serta regulasi laboratorium secara komprehensif [1]. Selanjutnya, Putra dan Suartana (2021) mengembangkan *Smart and Interactive Laboratory Management System (SI-LMS)* menggunakan metode *Waterfall* dengan teknologi *PHP*, *MySQL*, dan *SVM*. Sistem ini memiliki tingkat akurasi tinggi dalam memberikan rekomendasi penggantian inventaris berbasis *machine learning*,

namun belum mengimplementasikan mekanisme *role-based access control* (RBAC) [22].

Pratama *et al.* (2024) merancang Modul Aktivitas Laboratorium pada Portal Jurusan Teknik Elektro *Unila* dengan metode RAD, menghasilkan skor daya tarik dan kejelasan tinggi, tetapi masih terbatas pada modul aktivitas tanpa mencakup fitur peminjaman dan pengajuan bebas laboratorium [23]. Zuhdi *et al.* (2017) mengembangkan *Management Information Systems of Laboratory* menggunakan *Laravel* dengan metode *Waterfall*, menghasilkan sistem yang efektif melalui pengujian *blackbox* [24]. Jaya *et al.* (2023) dan Rabiah *et al.* (2022) memanfaatkan QR Code dalam sistem inventarisasi laboratorium untuk meningkatkan keamanan dan keteraturan data, namun keduanya hanya berfokus pada inventaris tanpa integrasi penjadwalan atau regulasi laboratorium [6].

Selanjutnya, Willyansah *et al.* (2025) mengembangkan Sistem Monitoring Laboratorium Komputer untuk mempermudah pemantauan kondisi perangkat, tetapi belum mencakup manajemen kegiatan laboratorium secara menyeluruh [9]. Saputra dan Hardiani (2025) mengintegrasikan SIMLAB dengan *Vue.js* dan *CodeIgniter* yang menyederhanakan proses kerja di laboratorium patologi [7].

2.2.2 Analisis Gap dan Posisi Penelitian

Berdasarkan telaah terhadap penelitian-penelitian terdahulu tersebut, dapat diidentifikasi beberapa *research gap*. Pertama, belum ada penelitian yang mengintegrasikan secara komprehensif fitur inventaris, penjadwalan, peminjaman, regulasi laboratorium, dan pengajuan bebas laboratorium dalam satu sistem yang utuh, khususnya untuk laboratorium teknik elektro. Kedua, sebagian besar penelitian masih berfokus pada aspek administratif dan inventarisasi tanpa mengintegrasikan regulasi dan dokumentasi prosedur operasional standar (SOP) sesuai ketentuan perguruan tinggi. Ketiga, belum ditemukan penerapan sistem manajemen laboratorium yang mengimplementasikan *role-based access control* secara menyeluruh untuk berbagai tingkat pengguna seperti mahasiswa, asisten

laboratorium, kepala laboratorium, dan PLP. Keempat, belum ada penelitian yang secara khusus menyesuaikan sistem dengan karakteristik serta regulasi laboratorium teknik elektro Universitas Lampung.

Posisi dan kontribusi penelitian ini terletak pada beberapa aspek kebaruan. Pertama, penelitian ini memiliki konteks spesifik yaitu pengembangan sistem manajemen laboratorium Teknik Elektro Universitas Lampung yang disesuaikan dengan kebutuhan dan prosedur institusional. Kedua, sistem ini mengintegrasikan regulasi laboratorium termasuk dokumentasi SOP dan standar keselamatan kerja dalam satu platform digital. Ketiga, penelitian ini menerapkan *role-based management* komprehensif dengan empat tingkat pengguna utama, yaitu mahasiswa biasa, asisten laboratorium, kepala laboratorium, dan PLP, masing-masing dengan hak akses yang berbeda. Keempat, sistem yang dikembangkan mencakup fitur layanan laboratorium secara menyeluruh, meliputi praktikum, penelitian, pengabdian masyarakat, pengujian alat, hingga pengajuan bebas laboratorium. Terakhir, dari sisi teknologi, penelitian ini menggunakan *framework Laravel*, yang memberikan keunggulan dalam keamanan, skalabilitas, serta kemudahan pemeliharaan untuk pengembangan jangka panjang.

2.3 Tools yang Digunakan

2.3.1 Laravel Framework

Laravel merupakan *framework PHP* dengan pendekatan *Model-View-Controller* (MVC) yang menyediakan struktur pengembangan aplikasi *web* yang elegan, efisien, dan terorganisir. Arsitektur MVC memisahkan logika bisnis (*Model*), tampilan (*View*), dan pengendali alur aplikasi (*Controller*), sehingga memudahkan proses pengembangan, pengujian, serta pemeliharaan sistem [24]. *Framework* ini memiliki dokumentasi lengkap, komunitas pengembang yang luas, serta fitur-fitur modern untuk mendukung pengembangan *web*. Keunggulan *Laravel* mencakup struktur kode yang bersih, keamanan *built-in* terhadap ancaman umum seperti *SQL injection* dan *cross-site scripting* (XSS), kemudahan dalam

pengelolaan basis data, serta integrasi fleksibel dengan berbagai *library* dan *tools* pendukung [24]. *Laravel* juga menonjol dalam aspek efisiensi, kemudahan pembelajaran, serta kelengkapan dokumentasi. *Framework* ini mendukung pengembangan cepat dengan dukungan *template engine Blade*, ORM *Eloquent*, dan sistem *middleware* yang kuat. Dengan fleksibilitas tinggi dan komunitas aktif, *Laravel* menjadi salah satu *framework PHP* paling populer untuk pengembangan aplikasi *web* modern [25].

2.3.2 *Application Programming Interface (API)*

API (Application Programming Interface) adalah sekumpulan metode atau fungsi yang disediakan oleh sebuah program, *library*, atau platform, yang memungkinkan aplikasi lain untuk berkomunikasi serta memanfaatkan fungsionalitas yang tersedia, seperti *jQuery API*, *Google Maps API*, maupun *REST API* pada sistem *backend* [14].

API juga sebagai antarmuka untuk berinteraksi dengan aplikasi *web*. Berbagai jenis *API* memiliki aturan, fungsi, dan tujuan yang berbeda, tetapi semuanya menggunakan format tertentu sebagaimana *JSON* atau *YAML* untuk pertukaran data antara aplikasi. *API* seringkali menggunakan mekanisme autentikasi dan otorisasi agar pengguna hanya dapat *mengakses* sumber daya sesuai dengan hak akses mereka.

Secara teknis, anatomi *API web* mencakup cara kerja *API* seperti *RESTful* dan *GraphQL*, di mana spesifikasi *API* biasanya ditulis menggunakan format yang mudah dipahami manusia maupun perangkat lunak. Pada pengembangan aplikasi modern, *API* menjadi komponen penting dalam integrasi sistem, memungkinkan pengembang untuk *mengakses* data dan layanan eksternal secara aman melalui protokol dan autentikasi yang telah ditentukan [14].

2.3.3 *PostgreSQL*

PostgreSQL merupakan sistem manajemen basis data relasional yang memiliki arsitektur objek yang kompleks dan ekstensif. Setiap layanan *PostgreSQL*

menampung banyak basis data individual, di mana setiap basis data memiliki skema sebagai tingkat organisasi berikutnya untuk menyimpan objek-objek seperti tabel, *View*, dan fungsi. *PostgreSQL* secara otomatis membuat skema bernama *public*. Tabel dalam *PostgreSQL* memiliki kemampuan khusus, yaitu *inheritance* dan secara otomatis menciptakan tipe data kustom, *View* yang dapat diperbarui termasuk *materialized Views* untuk meningkatkan performa *query* [26].

PostgreSQL juga dilengkapi dengan fitur-fitur *advanced* seperti *extension* yang membuat pengembang untuk mengemas berbagai komponen. *PostgreSQL* memiliki tingkat ekstensibilitas tinggi karena *PostgreSQL* dibangun dengan struktur yang dapat mereplikasi dirinya sendiri dan pengaturan sistem disimpan dalam tabel yang dapat *di-query* dan dimodifikasi [26].

2.3.4 Draw.io

Draw.io adalah aplikasi pembuatan diagram *online* yang gratis dan *open-source* dengan antarmuka *drag-and-drop* yang intuitif. Platform ini mendukung pembuatan berbagai jenis diagram profesional seperti *flowchart*, ERD (*Entity Relationship Diagram*), UML, *network diagram*, *wireframe*, dan *mind maps*. Bagi pengguna *casual*, draw.io memberikan *tools* ERD yang cukup baik. Tidak hanya *tools* ERD, draw.io juga berfungsi sebagai *flowchart maker* atau *algoritma*. Draw.io sudah mendukung penyimpanan berbasis *cloud* seperti *drive*, *OneDrive*, maupun *non-cloud*, serta penyimpanan lokal di perangkat [27]. Keunggulan utama draw.io adalah sifatnya yang gratis sepenuhnya tanpa biaya berlangganan, tidak memerlukan registrasi, dapat diakses melalui *web* browser atau aplikasi desktop, mendukung *export* dalam berbagai format (PNG, JPEG, SVG, PDF, XML), serta memungkinkan kolaborasi untuk berbagi dan bekerja bersama pada diagram.

2.3.5 Hypertext Preprocessor (PHP)

PHP (*Hypertext Preprocessor*) merupakan bahasa pemrograman skrip yang dirancang khusus untuk pengembangan *Web* dan dapat disematkan langsung ke dalam HTML [28]. *PHP* telah menjadi salah satu bahasa pemrograman yang paling banyak digunakan dalam pengembangan *Web* karena berbagai keunggulannya.

Popularitas *PHP* terletak pada aspek ekonomis, skalabilitas, kesederhanaan, dan kompatibilitasnya yang tinggi [28]. *PHP* kompatibel dengan semua sistem operasi utama termasuk *Windows*, *Unix*, *Linux*, dan *MacOS*, serta didukung oleh hampir semua penyedia *hosting* dengan kecepatan pemrosesan yang tinggi [28]. Dalam konteks pengembangan aplikasi modern, *PHP* sering dikombinasikan dengan *framework* untuk meningkatkan efisiensi dan kecepatan penulisan kode. *Framework* seperti *CodeIgniter*, *CakePHP*, dan *Symfony* telah terbukti memfasilitasi skalabilitas dan pemeliharaan jangka panjang sambil mematuhi standar pengembangan dan menjaga kode tetap terorganisir [28]. *PHP* juga mendukung berbagai *database* seperti *MySQL*, *Oracle*, *MongoDB*, dan *PostgreSQL*, menjadikannya pilihan yang fleksibel untuk berbagai jenis proyek *Web* [28]. Karakteristik *open source* dan lisensi gratis *PHP* membuatnya sangat diminati oleh pengembang, baik untuk proyek skala kecil maupun *enterprise* seperti *Facebook*, *Wikipedia*, *WordPress*, dan *Yahoo* yang menggunakan *PHP* sebagai teknologi utama mereka [28].

2.3.6 Figma

Figma adalah platform desain berbasis *cloud* yang digunakan untuk pembuatan *user interface* (UI) dan *prototype* aplikasi secara kolaboratif. Figma memungkinkan desainer untuk membuat *wireframe*, *mockup*, dan *prototype* interaktif yang dapat diakses dan diedit secara *real-time* oleh tim. Keunggulan utama Figma adalah sifatnya yang berbasis web sehingga tidak memerlukan instalasi *software* khusus, mendukung kolaborasi *real-time* di mana *multiple users* dapat mengedit desain secara bersamaan, menyediakan *component library* yang memudahkan konsistensi desain, dan memiliki fitur *prototyping* untuk membuat *interactive mockup* yang dapat di-*test* sebelum implementasi.

Dalam penelitian ini, Figma digunakan untuk merancang *user interface* dan *user experience* (UI/UX) sistem informasi manajemen laboratorium. Proses desain dimulai dengan pembuatan *wireframe* untuk mendefinisikan struktur dan layout setiap halaman, dilanjutkan dengan *high-fidelity mockup* yang menampilkan desain visual lengkap dengan warna, *typography*, dan elemen grafis. Figma juga

digunakan untuk membuat *prototype* interaktif yang memungkinkan *stakeholder* untuk melihat dan merasakan flow aplikasi sebelum masuk ke tahap *coding*. *Feedback* terhadap *prototype* kemudian digunakan untuk memperbaiki desain sebelum diimplementasikan, sehingga mengurangi risiko *rework* di tahap *development*. Figma juga menyediakan fitur *inspect* yang memudahkan untuk melihat spesifikasi teknis seperti ukuran, *spacing*, dan kode CSS dari setiap elemen desain, sehingga mempercepat proses implementasi *frontend*.

III. METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian ini dilakukan pada Laboratorium Teknik Komputer, yang berada di bawah Program Studi Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung dan dilakukan pada bulan November 2025 sampai dengan April 2026. Pembagian waktu penelitian dapat dilihat pada tabel 3.1.

Tabel 3.1 Waktu Pelaksanaan Penelitian

No	Kegiatan Penelitian	2025			2026		
		Nov	Des	Jan	Feb	Mar	Apr
1	<i>Planning</i>						
2	<i>Design</i>						
3	<i>Coding</i>						
4	<i>Testing</i>						
5	<i>Deployment</i>						
6	<i>Maintenance</i>						
7	Penulisan Karya Ilmiah						

3.2 Alat dan Bahan Penelitian

3.2.1 Alat Penelitian

Adapun alat yang digunakan dalam penelitian dibagi menjadi dua jenis, yaitu perangkat keras (Tabel 3.2) dan perangkat lunak (Tabel 3.3).

Tabel 3.2 Perangkat Keras

No	Perangkat Keras	Spesifikasi	Deskripsi
1	Laptop	AMD Ryzen 5 5625U, RAM 16GB, Windows 11	Perangkat utama pengembangan

Tabel 3.3 Perangkat Lunak

No	Perangkat Lunak	Versi	Deskripsi
1	<i>Google Chrome</i> (Browser)	Versi 128	<i>Tools</i> implementasi dan <i>testing website</i>
2	Visual Studio Code	Versi 1.82	<i>Text</i> editor untuk pengembangan kode
3	<i>Laravel</i>	Versi 10	Framework <i>backend</i> PHP
4	<i>PHP</i>	Versi 8.1	Bahasa pemrograman
5	<i>PostgreSQL</i>	Versi 15	<i>Database</i> untuk penyimpanan data

3.2.2 Bahan Penelitian

Penelitian ini memanfaatkan berbagai sumber referensi seperti buku, jurnal, skripsi, dan karya ilmiah untuk memperkuat landasan teori, pemahaman, serta metode yang digunakan dalam pengembangan fitur manajemen laboratorium pada portal program studi Jurusan Teknik Elektro. Sumber-sumber tersebut mencakup

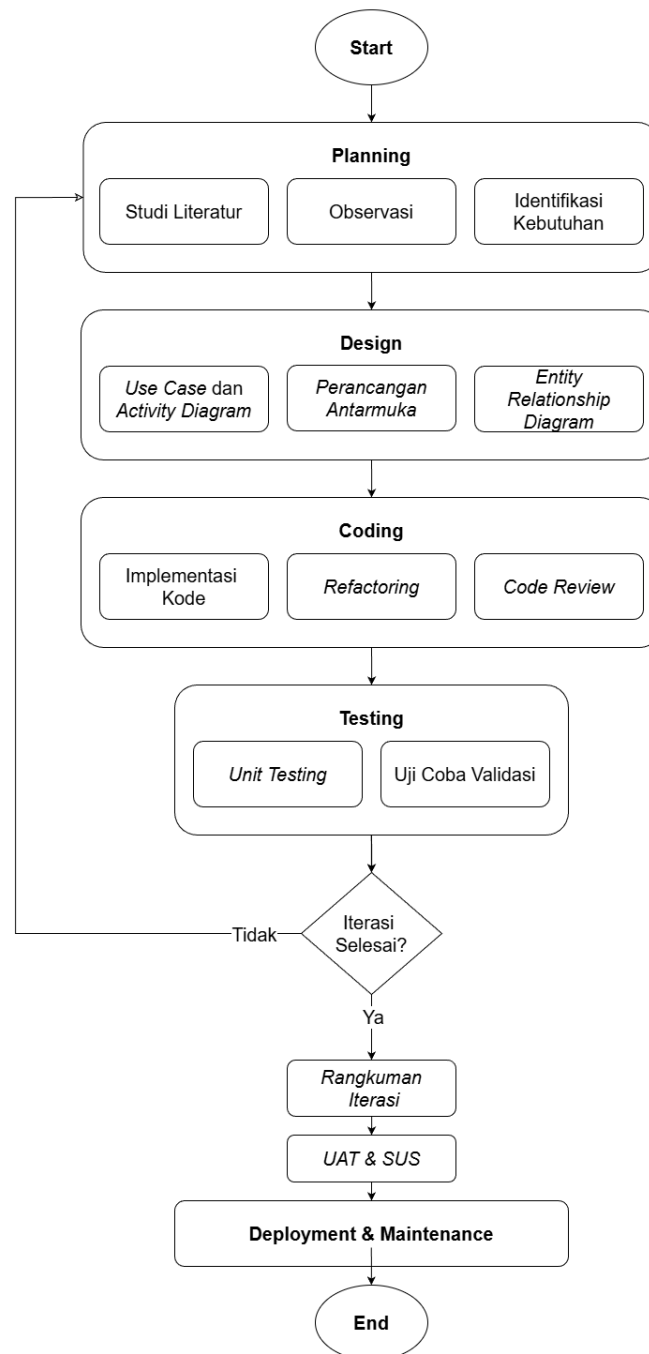
studi terkait pengembangan sistem informasi, analisis terhadap penelitian terdahulu sebagai acuan dalam perancangan fitur lanjutan, serta referensi metodologis yang digunakan sebagai dasar dalam analisis dan pembahasan hasil pengembangan sistem berbasis *framework Laravel* guna mendukung penerapan regulasi laboratorium di Universitas Lampung.

3.3 Metode dan Tahapan Penelitian

Pengembangan fitur manajemen laboratorium pada penelitian ini menggunakan metode XP, salah satu metodologi *Agile* yang menekankan kualitas kode dan kepuasan pengguna melalui iterasi pengembangan yang singkat dan fleksibel. Tahapan XP yang diterapkan meliputi *planning*, *design*, *coding*, dan *testing* sebagaimana dijelaskan pada Bab II. Pengembangan dipecah menjadi 12 iterasi pendek dengan durasi 1–2 minggu per iterasi yang dilaksanakan di dalam periode penelitian selama enam bulan. Setiap iterasi menghasilkan peningkatan fungsionalitas yang dapat diuji dan dievaluasi, sehingga dapat dimasukkan ke dalam perencanaan iterasi berikutnya.

Tabel 3.4 Ringkasan Metode XP

Tahapan	Input	Output
<i>Planning</i>	Analisis Kebutuhan, Hasil observasi	<i>Product Backlog, Iteration Plan, User Stories</i>
<i>Design</i>	<i>User Stories, Iteration Plan</i>	<i>Activity Diagram, Antarmuka</i>
<i>Coding</i>	<i>Design sistem, User Stories</i>	Kode program (<i>Controller, Model, View</i>), <i>Repository</i>
<i>Testing</i>	<i>Test scenarios</i>	<i>Test reports, Code coverage report</i>



Gambar 3.1 Tahapan Penelitian

Berdasarkan Gambar 3.1 Tahapan Penelitian, terdapat tahapan *planning* dilakukan studi literatur terhadap berbagai sumber akademik untuk memahami konsep SIMLAB, XP, ISO 17025, RBAC, teknologi yang digunakan, *usability testing*, observasi dilakukan di laboratorium selama 4 minggu, dan identifikasi kebutuhan dilakukan melalui kuesioner kepada berbagai pengguna (PLP, Kepala Lab, Asisten Lab, Mahasiswa, Dosen) untuk mengumpulkan kebutuhan fungsional

dan non-fungsional *s*, fase *planning* ini berbeda dengan XP standar (Gambar 2.1) yang fokus pada *business context* understanding dan *user stories definition* karena penelitian ini memperluas *scope* dengan menambahkan studi literatur dan observasi lapangan untuk memahami regulasi laboratorium kompleks (Renstra Unila, ISO 17025, PermenPANRB, Permendikbudristek) dan kebutuhan melalui observasi dan kuesioner. Setelah fase *planning* selesai, masuk ke tahapan *design* di mana dibuat *use case diagram* dan *activity diagram* untuk menggambarkan alur proses bisnis sistem, tahapan *design* ini juga berbeda dengan XP standar yang menggunakan *CRC cards* sebagai *tool* utama. Namun penelitian ini menggunakan kombinasi *tools* yang lebih luas yaitu *use case diagram*, *activity diagram*, dan Tampilan Antarmuka.

Kemudian tahapan *coding* dimulai implementasi kode dilakukan berdasarkan *design* yang telah dibuat, *refactoring* dilakukan untuk meningkatkan kualitas kode, dan *code review* dilakukan untuk memastikan kode sesuai dengan standar yang ditetapkan. Namun berbeda dengan XP standar yang menerapkan *pair programming* secara konsisten, penelitian ini tidak menerapkan *pair programming* karena penelitian dilakukan oleh seorang *researcher individual*, meskipun tetap mempertahankan prinsip-prinsip XP lainnya seperti *continuous refactoring* untuk *code quality improvement* dan *code review* melalui *self-review* dan *feedback* yang menggantikan fungsi *pair programming* dalam *quality assurance*.

Setelah *coding* selesai, dilanjutkan dengan tahapan *testing* yang mencakup *unit testing* untuk mengetes *individual units* dari kode, dan uji coba validasi (*functional testing*) untuk memverifikasi bahwa sistem memenuhi *requirement* yang telah didefinisikan. Setelah keempat tahapan ini selesai, dilakukan pengecekan apakah iterasi sudah selesai semua dengan target 12 iterasi total dengan durasi 1-2 minggu per iterasi selama 6 bulan penelitian, jika tidak semua iterasi selesai, maka siklus kembali ke tahapan *planning* untuk iterasi berikutnya dengan kebutuhan yang baru, sehingga memastikan bahwa setiap *incremental release* telah melalui empat tahapan. Jika semua 12 iterasi sudah selesai, dilakukan rangkuman iterasi yang merekap seluruh hasil pengembangan dari setiap iterasi, mencakup fitur yang telah diimplementasikan, hasil *testing*, serta catatan *refactoring* dan perbaikan yang dilakukan sepanjang proses pengembangan. Rangkuman ini berfungsi sebagai dokumentasi menyeluruh sebelum sistem dibawa ke tahap evaluasi pengguna.

Setelah rangkuman iterasi selesai, dilakukan *User Acceptance Testing & SUS* yang melibatkan pengguna nyata untuk mengukur kemudahan penggunaan sistem menggunakan *System Usability Scale* dan *user satisfaction*, karena aspek *usability* dan *user experience* menjadi prioritas penting dalam pengembangan sistem yang akan digunakan oleh berbagai tipe pengguna dengan background berbeda (mahasiswa, asisten lab, kepala lab, PLP). Setelah UAT & SUS selesai, sistem siap masuk ke tahapan *deployment & maintenance* untuk diimplementasikan di lingkungan production laboratorium dan dilakukan monitoring serta maintenance. Dengan adaptasi ini terhadap XP standar, penelitian tetap mempertahankan *essence* dari XP yaitu iterasi pendek, *quality emphasis*, dan *continuous feedback*, namun disesuaikan dengan penelitian akademik individual, sistem informasi laboratorium yang dipengaruhi oleh *multiple regulatory frameworks* (Renstra, ISO 17025, PermenPANRB, Permendikbudristek).

3.3.1 *Planning*

Tahap *planning* dilakukan dengan mengidentifikasi dan merencanakan kebutuhan pengembangan melalui kuesioner kepada PLP, Kepala Laboratorium, Asisten Laboratorium, Mahasiswa, serta Dosen. Perencanaan sistem diarahkan untuk mendukung dokumentasi kegiatan, pengelolaan sumber daya, dan monitoring risiko, sambil mengadopsi prinsip ISO 17025 yang relevan pada aspek penjaminan mutu, struktur organisasi, ketidakberpihakan, kerahasiaan data, serta pengelolaan inventaris laboratorium.

Untuk mengumpulkan kebutuhan, melakukan studi literatur terhadap berbagai sumber akademik tentang SIMLAB dan *Laboratory Information Management System*, *Extreme Programming* sebagai *framework* pengembangan, ISO 17025 dan *management system* untuk *laboratory*, RBAC untuk implementasi permission, *Frontend & Backend Development* dengan Laravel dan PostgreSQL, *Usability Testing*. Setelah itu dilakukan analisis mendalam terhadap ISO 17025-2017, Renstra Universitas Lampung 2025-2029, PermenPANRB No. 7 Tahun 2019 tentang Pranata Laboratorium Pendidikan, dan Permendikbudristek No. 49 Tahun

2024 tentang Organisasi dan Tata Kerja Universitas Lampung untuk memastikan sistem memenuhi semua requirement regulasi yang ada.

Kebutuhan fungsional dan non-fungsional diidentifikasi kemudian disusun menjadi *user stories* dengan format "*As a [user], I want [feature], so that [benefit]*", diprioritaskan dengan kategori *High (must have)*, *Medium (should have)*, dan *Low (nice to have)*.

Seluruh *user stories* yang telah dikumpulkan dan diprioritaskan kemudian dijadikan dasar dalam proses *release planning*. *Release planning* dilakukan dengan mengorganisir seluruh *user stories* tersebut ke dalam 12 iterasi pengembangan yang dilaksanakan selama enam bulan dari November 2025 hingga April 2026, dengan durasi 1–2 minggu per iterasi. Setiap iterasi dirancang untuk menghasilkan *increment* fungsionalitas yang dapat diuji dan dievaluasi, sehingga dapat dimasukkan ke dalam perencanaan iterasi berikutnya. Output dari *release planning* ini adalah *iteration plan* yang menjadi acuan pelaksanaan pengembangan sistem secara keseluruhan. Perencanaan ini menyesuaikan ruang lingkup laboratorium yang merupakan laboratorium pendidikan (bukan laboratorium pengujian), sehingga prinsip-prinsip ISO 17025 diterapkan hanya pada aspek yang relevan dengan tata kelola, dokumentasi, keamanan data, dan manajemen sumber daya laboratorium.

Tabel 3.5 Tahap *Planning*

Input	Proses	Output
Kebutuhan	Kuesioner dan observasi	Daftar kebutuhan fungsional dan non-fungsional
Proses bisnis laboratorium	Analisis dan prioritasasi	<i>User stories</i> dengan prioritas
<i>User stories</i>	Estimasi <i>story point</i>	<i>Iteration plan</i> 12 iterasi

3.3.2 Design

Dalam tahap desain, sistem dirancang agar selaras dengan regulasi yang berlaku, termasuk Renstra Unila dan Permen terkait, sehingga digitalisasi layanan, otomatisasi dokumentasi dapat tercapai. Prinsip-prinsip ISO 17025 diintegrasikan dalam desain melalui fitur utama seperti dokumentasi aktivitas, pencatatan inventaris alat, pelaporan kegiatan. Desain sistem dilakukan secara sederhana dan incremental dengan prinsip YAGNI (*You Aren't Gonna Need It*), hanya mengembangkan fitur yang diperlukan pada setiap iterasi pengembangan. Perancangan arsitektur sistem menggunakan pola *Model-View-Controller* (MVC) dari Laravel dengan API *endpoints* menggunakan prinsip RESTful. *Refactoring* juga disusun dengan identifikasi kode dari iterasi sebelumnya yang perlu diperbaiki demi menjaga struktur kode tetap optimal seiring perubahan kebutuhan.

Perancangan *User Interface* (UI/UX) juga dilakukan untuk memastikan sistem *user-friendly* dengan membuat *wireframe* untuk setiap halaman utama, mengidentifikasi komponen UI yang akan digunakan, menerapkan *responsive design* untuk berbagai ukuran layar. Perancangan UML Diagram mencakup *Use Case Diagram* untuk menunjukkan interaksi aktor dengan sistem, *Activity Diagram* untuk alur proses bisnis.

Tabel 3.6 Tahap *Design*

Input	Proses	Output
Analisis kebutuhan	Perancangan Antarmuka	Tampilan antarmuka
<i>Requirement</i> sistem	Pemodelan UML	<i>Use case diagram, Activity diagram.</i>

3.3.3 Coding

Tahap *coding* mengimplementasikan *user stories* menjadi kode program yang dapat dijalankan dengan menerapkan standar pengkodean konsisten dan praktik pengembangan terbaik. Standar pengkodean mengikuti konvensi *Laravel* dan PSR (*PHP Standard Recommendation*) dengan penerapan *naming convention* yang jelas untuk *variabel, function, dan class*, serta implementasi prinsip *clean code* untuk keterbacaan dan *maintainability*. *Version control* menggunakan *Bitbucket*

dengan melakukan *commit* dan *push code* secara berkala ke *repository* untuk integrasi kode baru dengan kode eksisting guna menghindari konflik. Implementasi teknis dilakukan dengan *Laravel* yang mencakup pembuatan *Controller* untuk *handle business logic*, *Model* untuk interaksi dengan *database* menggunakan *Eloquent ORM*, serta validasi input untuk keamanan data. *Frontend* dibangun menggunakan *Blade Template Engine*, sedangkan *PostgreSQL* digunakan untuk penyimpanan data. *Refactoring* dilakukan secara berkala untuk mengeliminasi *code duplication* mengikuti prinsip DRY (*Don't Repeat Yourself*), menyederhanakan logika kompleks menjadi fungsi *reusable*. Dokumentasi kode meliputi dokumentasi *API endpoints*, *inline documentation* untuk fungsi-fungsi penting.

Tabel 3.7 Tahap Coding

Input	Proses	Output
<i>Design sistem</i>	Setup environment Laravel	Struktur project Laravel
<i>User stories</i>	Implementasi <i>Controller</i>	<i>File Controller</i> dengan <i>business logic</i>
Struktur <i>database</i>	Pembuatan <i>Model</i>	<i>Eloquent Models</i> dengan relasi
<i>Wireframe/Mockup</i>	Pembuatan <i>View</i>	<i>Blade templates</i>
<i>API endpoints design</i>	Implementasi <i>Routes</i>	<i>File routes</i> dengan <i>API endpoints</i>
<i>Business logic</i>	Implementasi validasi	<i>Validation rules</i> dan <i>error handling</i>
Kode kompleks	<i>Refactoring</i>	<i>Code clean</i> dan <i>reusable</i>
Semua implementasi	<i>Commit & push</i>	<i>Repository code</i> dengan <i>version history</i>

3.3.4 Testing

Tahap *testing* dilakukan secara berkala dan otomatis pada setiap iterasi untuk memastikan kualitas sistem sesuai standar ISO 17025 terkait validitas hasil dan keamanan data. Dengan menggunakan prinsip "*All tests must pass*" sebelum kode diintegrasikan atau di-*deploy* ke *production*. *Testing* mencakup *unit testing* menggunakan *PHPUnit* dengan target *code coverage* minimal 70% untuk memastikan setiap fungsi berjalan dengan benar, *validation testing* untuk memastikan data *input* dan alur proses memenuhi aturan bisnis yang ditetapkan, *usability testing* dengan sampel pengguna menggunakan metrik *System Usability Scale* (SUS) untuk mengukur aspek *learnability*, *efficiency*, *memorability*, *errors*, dan *satisfaction*, *performance testing* untuk mengukur kecepatan *response time* dan melakukan *load testing* terhadap sistem, serta *security testing* untuk mendeteksi kerentanan terhadap *SQL Injection*, *XSS*, dan *CSRF* pada implementasi *Laravel Sanctum* dan integrasi *Unila SSO*. Setiap bug yang ditemukan dicatat dan diprioritaskan berdasarkan *severity*, kemudian diperbaiki sebelum fitur dianggap selesai.

Tabel 3.8 Tahap *Testing*

Input	Proses	Output
Kode program (<i>Controller, Model, View</i>)	<i>Unit testing</i> dengan <i>PHPUnit, Faker, Mockery</i>	<i>Test report, Code coverage report</i>
<i>API endpoints</i>	<i>Integration testing</i>	<i>API test results</i>
Fitur yang sudah diimplementasi	<i>Functional testing</i>	<i>Test scenarios report</i>

3.3.5 *Deployment dan Maintenance*

Tahap *deployment* dilakukan setelah semua iterasi selesai dengan konfigurasi *server production*, setup dan *migration database PostgreSQL*, *upload* kode melalui *FTP* atau *Git* dengan konfigurasi *environment variables* untuk koneksi *database* dan *Unila SSO*, *build frontend assets* menggunakan *Vite* untuk *production*, *testing* final untuk memastikan semua fitur berfungsi, kemudian

handover sistem kepada pengguna dengan *training* singkat. Tahap *maintenance* berfokus pada *monitoring* performa sistem, penanganan *bug* yang muncul, pembaruan fitur berdasarkan *feedback* pengguna, *backup database PostgreSQL* secara rutin, serta *monitoring* berkelanjutan untuk mengidentifikasi area yang perlu diperbaiki, dan jika terdapat kebutuhan pengembangan fitur baru, proses akan mengikuti kembali tahapan XP dari *planning* hingga *deployment*.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian dan pengembangan yang telah dilakukan pada Portal Prodi Manajemen Lab di Jurusan Teknik Elektro Universitas Lampung, berikut adalah kesimpulan yang disusun untuk menjawab rumusan masalah penelitian:

1. Proses perancangan dan pembangunan Portal Laboratorium Prodi berhasil dilakukan melalui penerapan metode XP yang mendukung pengembangan secara bertahap, meliputi iterasi, perancangan sederhana, refactoring, dan pengujian berkelanjutan. Pada penelitian ini *pair programming* tidak dapat diterapkan secara penuh karena proses pengembangan dilakukan oleh satu peneliti
2. Sebanyak 17 (tujuh belas) kebutuhan fungsional dan 7 (tujuh) kebutuhan non-fungsional sistem informasi manajemen laboratorium berhasil diidentifikasi melalui analisis kebutuhan pengguna yang melibatkan lima peran, yaitu mahasiswa, asisten laboratorium, kepala laboratorium, PLP, dan dosen. Seluruh kebutuhan fungsional tersebut berhasil diimplementasikan ke dalam sistem dan diunggah pada server Universitas Lampung.
3. Pengujian *unit testing* yang dilaksanakan pada setiap iterasi pengembangan selama 12 iterasi dengan total 69 skenario uji menunjukkan bahwa seluruh skenario berhasil dijalankan dengan status lulus. Hal ini membuktikan bahwa logika dan fungsionalitas sistem yang dibangun telah berjalan sesuai dengan spesifikasi kebutuhan yang telah ditetapkan.

4. Hasil pengujian menunjukkan bahwa Portal Laboratorium Prodi telah memenuhi standar kualitas fungsional dan kegunaan yang baik. Pengujian *User Acceptance Testing* (UAT) yang melibatkan 72 responden menghasilkan tingkat keberhasilan sebesar 98,19% dari total 663 skenario uji, yang membuktikan bahwa sistem mampu memenuhi kebutuhan pengguna akhir secara fungsional. Evaluasi *usability* menggunakan *System Usability Scale* (SUS) menghasilkan rata-rata skor 88,08 yang menempatkan sistem pada kategori grade B dengan adjective rating "Good" dan tingkat acceptability Acceptable, membuktikan bahwa sistem yang dibangun memiliki tingkat kemudahan penggunaan yang baik.

5.2 Saran

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan, terdapat beberapa saran untuk pengembangan Portal Laboratorium Prodi ke depannya:

1. Menambahkan notifikasi berbasis WhatsApp guna meningkatkan responsivitas terhadap aktivitas yang memerlukan tindak lanjut, seperti persetujuan peminjaman alat laboratorium, pengingat batas waktu pengembalian alat, serta pembaruan status pengajuan surat bebas laboratorium. Saran ini didasarkan pada hasil analisis kuesioner yang menunjukkan bahwa sebanyak 84,8% (78 dari 92) responden Asisten Laboratorium dan seluruh responden PLP menyatakan WhatsApp sebagai preferensi utama notifikasi, mengingat mayoritas pengguna 80% menggunakan smartphone saat berada di laboratorium.
2. Antarmuka pengguna (*user interface*) Portal Prodi secara keseluruhan disarankan untuk diperbarui agar memenuhi standar desain antarmuka yang lebih modern. Hal ini didukung oleh hasil evaluasi SUS di mana terdapat 3 responden (4,3%) yang memberikan grade D (Poor) dan 3 responden (4,3%) grade F (Awful), yang mengindikasikan masih terdapat aspek tampilan dan kemudahan navigasi yang perlu ditingkatkan. Bagi pengembang berikutnya, disarankan untuk memprioritaskan pembaruan antarmuka pengguna sebelum melanjutkan pengembangan fitur-fitur baru.

3. Sebagai pengembangan lanjutan, dapat dipertimbangkan pembangunan aplikasi berbasis *mobile* untuk memperluas aksesibilitas sistem. Hal ini sejalan dengan temuan kuesioner bahwa 81% mahasiswa (17 dari 21) dan 80% asisten laboratorium menggunakan *smartphone* sebagai perangkat utama saat mengakses laboratorium, sehingga aksesibilitas *mobile* menjadi kebutuhan nyata pengguna. Namun, pembaruan antarmuka web sebagaimana disebutkan pada poin sebelumnya hendaknya diprioritaskan terlebih dahulu.
4. Pengembangan selanjutnya disarankan untuk melengkapi pengujian sistem dengan *performance testing* dan *stress testing* guna mengukur kemampuan sistem dalam menangani beban pengguna secara bersamaan. Pada penelitian ini, pengujian dibatasi pada *unit testing*, UAT, dan SUS, sehingga belum mencakup evaluasi performa sistem ketika diakses oleh seluruh civitas Jurusan Teknik Elektro secara simultan.

DAFTAR PUSTAKA

DAFTAR PUSTAKA

- [1] P. Purwanto, S. Sudarwanto, and S. Sukaswanto, "Sistem Informasi Manajemen Pembelajaran Pada Pembelajaran Praktikum Laboratorium Di Jurusan Pendidikan Teknik Otomotif," *J. Pendidik. Vokasi Otomotif*, vol. 5, no. 1, pp. 110–119, Jan. 2023, doi: 10.21831/jpvo.v5i1.57772.
- [2] N. N. Rabiah, L. Lindawati, and S. Sarjana, "Web-Based Laboratory Inventory Application Using QR Code and RFID in Telecommunication Engineering Laboratories/Workshops," *Sinkron*, vol. 7, no. 4, pp. 2248–2261, Oct. 2022, doi: 10.33395/sinkron.v7i4.11624.
- [3] Kementerian Pendayagunaan Aparatur Negara dan Reformasi Birokrasi Republik Indonesia, "Peraturan Menteri Pendayagunaan Aparatur Negara dan Reformasi Birokrasi Nomor 7 Tahun 2019 tentang Jabatan Fungsional Pranata Laboratorium Pendidikan," Jakarta, Indonesia, 2019.
- [4] Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia, "Peraturan Menteri Pendidikan, Kebudayaan, Riset, dan Teknologi Nomor 49 Tahun 2024 tentang Organisasi dan Tata Kerja Universitas Lampung," Berita Negara Republik Indonesia Tahun 2024 Nomor 633, Jakarta, Indonesia, 2024.
- [5] Universitas Lampung, "Rencana Strategis Universitas Lampung 2025-2029," Bandar Lampung, Indonesia, 2024.
- [6] D. K. Jaya, L. E. Susilowati, and H. R. Akhdiyati, "Sistem Informasi Quick Response Code (Qr Code) Sederhana Untuk Mengoptimalkan Inventarisasi Peminjaman Dan Penggunaan Alat-Alat Laboratorium Di Jurusan Ilmu Tanah, Universitas Mataram," *J. Abdi Insani*, vol. 10, no. 2, pp. 771–783, May 2023, doi: 10.29303/abdiinsani.v10i2.951.
- [7] D. E. Saputra and T. Hardiani, "Integrasi fitur sistem informasi manajemen laboratorium menggunakan framework Vue.js dan CodeIgniter," in *Pros. Semnas LPPM Unisa Yogyakarta*, vol. 3, Feb. 2025, pp. 234–241.
- [8] A. K. Nasich, S. A. Wicaksono, and M. C. Saputra, "Implementasi Role Based Access Control (RBAC) dalam Sistem Informasi Manajemen Pelanggan dan Pembayaran Air Berbasis Web (studi pada PT Tirta Wangi Sejahtera)" *J. pengemb Teknol. Inf. Ilmu Komputer.*, vol. 9, no.9, 2025.
- [9] W. Willyansah, F. Ayu, and M. Muhammad, "Implementasi Sistem Informasi Monitoring Laboratorium Komputer Berbasis Web Menggunakan Metode Waterfall," *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 7, no. 1, pp. 166–171, Jan. 2025, doi: 10.47233/jteksis.v7i1.1753.
- [10] T. Dingsøyr, S. Nerur, V. Baliyepally, and N. B. Moe, "A decade of agile methodologies: Towards explaining agile software development," *J. Syst.*

- Softw.*, vol. 85, no. 6, pp. 1213–1221, June 2012, doi: 10.1016/j.jss.2012.02.033.
- [11] R. I. Borman, A. T. Priandika, and A. R. Edison, “Implementasi Metode Pengembangan Sistem Extreme Programming (XP) pada Aplikasi Investasi Peternakan,” *J. Sist. Dan Teknol. Inf. Justin*, vol. 8, no. 3, p. 272, July 2020, doi: 10.26418/justin.v8i3.40273.
- [12] A. Shrivastava, I. Jaggi, N. Katoch, D. Gupta, and S. Gupta, “A Systematic Review on Extreme Programming,” *J. Phys. Conf. Ser.*, vol. 1969, no. 1, p. 012046, July 2021, doi: 10.1088/1742-6596/1969/1/012046.
- [13] W. Yustika, N. T. Siregar, and V. A. Barus, “Peranan Sistem Database Di Dalam Sistem Informasi Manajemen Pada UINSU (Universitas Islam Negeri Sumatera Utara),” *Surplus J. Ekon. Dan Bisnis*, vol. 1, no. 2, pp. 188–196, June 2023, doi: 10.71456/sur.v1i2.428.
- [14] J. Duckett, *Front-End Back-End Development with HTML, CSS, JavaScript, jQuery, PHP, and MySQL*, 1st ed. Wiley Publishing, 2022.
- [15] T. Arianti, A. Fa’izi, S. Adam, and M. Wulandari, “Perancangan Sistem Informasi Perpustakaan Menggunakan Diagram UML (Unified Modelling Language),” *J. Ilm. Komput. Terap. Inf.*, vol. 1, no. 1, pp. 19–25, 2022.
- [16] D. Liu, R. G. Bias, M. Lease, and R. Kuipers, “Crowdsourcing for usability testing,” *Proc. Am. Soc. Inf. Sci. Technol.*, vol. 49, no. 1, pp. 1–10, 2012, doi: 10.1002/meet.14504901100.
- [17] Veni Manik, C. Hetty Primasari, Yohanes Priadi Wibisono, and Aloysius Bagas Pradipta Irianto, “Evaluasi Usability pada Aplikasi Mobile ACC.ONE menggunakan System Usability Scale (SUS) dan Usability Testing,” *J. Sains Dan Inform.*, vol. 7, no. 1, pp. 1–10, Mar. 2021, doi: 10.34128/jsi.v7i1.286.
- [18] “T. Teni and A. Yudianto, “Pengaruh Motivasi Belajar Terhadap Hasil Belajar Siswa Kelas VII SMP Negeri 2 Kedokanbunder Kabupaten Indramayu,” *Jurnal Pendidikan Indonesia*, vol. 2, no. 1, pp. 104-114, Jan. 2021.”
- [19] V. Khorikov, *Unit Testing Principles, Practices, and Patterns*. New York: Manning Publications Co. LLC, 2020.
- [20] B. Hambling and P. van Goethem, “User Acceptance Testing: A Step-by-Step Guide”. Swindon, UK: BCS, The Chartered Institute for IT, 2013.
- [21] “Badan Standardisasi Nasional, SNI ISO/IEC 17025:2017 – Persyaratan umum kompetensi laboratorium pengujian dan kalibrasi. Jakarta: BSN, 2018.”
- [22] R. E. Putra and I. Made Suartana, “Development of Smart and Interactive Laboratory Management System (SI-LMS),” in *2021 Fourth International Conference on Vocational Education and Electrical Engineering (ICVEE)*, Surabaya, Indonesia: IEEE, Oct. 2021, pp. 1–5. doi: 10.1109/ICVEE54186.2021.9649702.
- [23] M. Pratama, T. Yulianti, H. D. Septama, and M. Komarudin, “Pengembangan Modul Aktivitas Laboratorium pada Sistem Informasi Portal Jurusan Teknik Elektro Universitas Lampung,” in *Pros. Semnas Teknik Elektro Unila*, 2025.
- [24] M. I. H. Zuhdi, S. Subiyanto, and S. Sukamta, “Management information systems of laboratory using laravel framework:case study at electrical engineering of Universitas Negeri Semarang,” *J. Pendidik. Vokasi*, vol. 7, no. 2, p. 158, July 2017, doi: 10.21831/jpv.v7i2.13317.

- [25] M. Laaziri, K. Benmoussa, S. Khoulji, K. Mohamed Larbi, and A. E. Yamami, "A comparative study of laravel and symfony PHP frameworks," *Int. J. Electr. Comput. Eng. IJECE*, vol. 9, no. 1, p. 704, Feb. 2019, doi: 10.11591/ijece.v9i1.pp704-712.
- [26] R.O. Obe and L.S. Hsu, *PostgreSQL: Up and Running*, 3rd ed. Sebastopol, CA: O'Reilly Media, 2018.
- [27] D. R. Anamisa and F. A. Mufarroha, *Dasar Pemrograman WEB Teori dan Implementasi: HTML, CSS, Javascript, Bootstrap, Codelgniter*, 1st ed. Malang: Media Nusa Creative, 2020.
- [28] S. Sotnik, V. Manakov, and V. Lyashenko, "Overview: PHP and MySQL Features for Creating Modern Web Projects ," *Int. J. Artif. Intell. Appl. Sci. Res.*, vol. 7, no. 1, pp. 1–7, Jan. 2023.