

**PENGEMBANGAN *BACKEND* PADA SISTEM *MONITORING*  
SKRIPSI DI JURUSAN PERIKANAN DAN KELAUTAN  
UNIVERSITAS LAMPUNG**

**(Skripsi)**

**Oleh**

**DIMAS HIDAYANTO NAIM**

**NPM 2115061035**



**FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG**

**2026**

**PENGEMBANGAN *BACKEND* PADA SISTEM *MONITORING*  
SKRIPSI DI JURUSAN PERIKANAN DAN KELAUTAN  
UNIVERSITAS LAMPUNG**

**Oleh**

**DIMAS HIDAYANTO NAIM**

**Skripsi**

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

**Pada**

**Jurusan Teknik Elektro  
Fakultas Teknik Universitas Lampung**



**FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2026**

## **ABSTRAK**

### **PENGEMBANGAN *BACKEND* PADA SISTEM *MONITORING* SKRIPSI DI JURUSAN PERIKANAN DAN KELAUTAN UNIVERSITAS LAMPUNG**

**Oleh**

**DIMAS HIDAYANTO NAIM**

Pemantauan progres skripsi mahasiswa di Jurusan Perikanan dan Kelautan Universitas Lampung masih dilakukan secara manual sehingga proses rekapitulasi data membutuhkan waktu yang lama. Penelitian ini bertujuan untuk merancang dan mengembangkan *backend* sistem *monitoring* skripsi berbasis *website* yang mendukung pencatatan dan pemantauan progres skripsi mahasiswa secara lebih efisien. Pengembangan *backend* dilakukan menggunakan teknologi *Node.js* dan *Express.js* dengan arsitektur RESTful API serta *PostgreSQL* sebagai basis data, dan menerapkan metodologi *Scrum*. Sistem yang dikembangkan menyediakan fitur pengelolaan data skripsi, bimbingan, serta pemantauan progres mahasiswa. Hasil penelitian menunjukkan bahwa sistem *backend* mampu meningkatkan efisiensi proses rekapitulasi data progres skripsi, di mana waktu yang dibutuhkan untuk memperoleh data progres skripsi adalah sekitar  $\pm 5$  detik dibandingkan dengan proses manual yang memerlukan waktu sekitar 1–2 hari kerja.

Kata Kunci: Sistem *Monitoring* Skripsi, *Backend*, RESTful API, *Node.js*, *Express.js*, *PostgreSQL*.

## **ABSTRACT**

### **BACKEND DEVELOPMENT OF A THESIS MONITORING SYSTEM IN THE DEPARTMENT OF FISHERIES AND MARINE SCIENCE UNIVERSITAS LAMPUNG**

**By**

**DIMAS HIDAYANTO NAIM**

*The monitoring of undergraduate thesis progress in the Department of Fisheries and Marine Science at Universitas Lampung is still conducted manually, resulting in a time-consuming data recapitulation process. This study aims to design and develop the backend of a web-based thesis monitoring system that supports more efficient recording and monitoring of students' thesis progress. The backend was developed using Node.js and Express.js with a RESTful API architecture and PostgreSQL as the database, and the Scrum methodology was applied in the development process. The developed system provides features for thesis data management, supervision activities, and monitoring of students' progress. The results show that the backend system improves the efficiency of thesis progress data recapitulation, where the time required to retrieve thesis progress data is approximately  $\pm 5$  seconds compared to the manual process, which requires about 1–2 working days.*

*Keywords: Thesis Monitoring System, Backend, RESTful API, Node.js, Express.js, PostgreSQL.*

**Judul Skripsi** : **PENGEMBANGAN BACKEND PADA  
SISTEM MONITORING SKRIPSI DI  
JURUSAN PERIKANAN DAN KELAUTAN  
UNIVERSITAS LAMPUNG**

**Nama Mahasiswa** : **Dimas Hidayanto Naim**

**Nomor Pokok Mahasiswa** : **2115061035**

**Program Studi** : **Teknik Informatika**

**Jurusan** : **Teknik Elektro**

**Fakultas** : **Teknik**



**MENYETUJUI**

**1. Komisi Pembimbing**

**Pembimbing Utama**

**Wahyu Eko Sulistiono, S.T., M.Sc.**  
**NIP 197412012001121001**

**Pembimbing Pendamping**

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

**2. Mengetahui**

**Ketua Jurusan  
Teknik Elektro**

**Herlinawati, S.T., M.T.**  
**NIP 19710314199032001**

**Ketua Program Studi  
Teknik Informatika**

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

**MENGESAHKAN**

**1. Tim Penguji**

**Ketua : Wahyu Eko Sulistiono, S.T., M.Sc.**

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

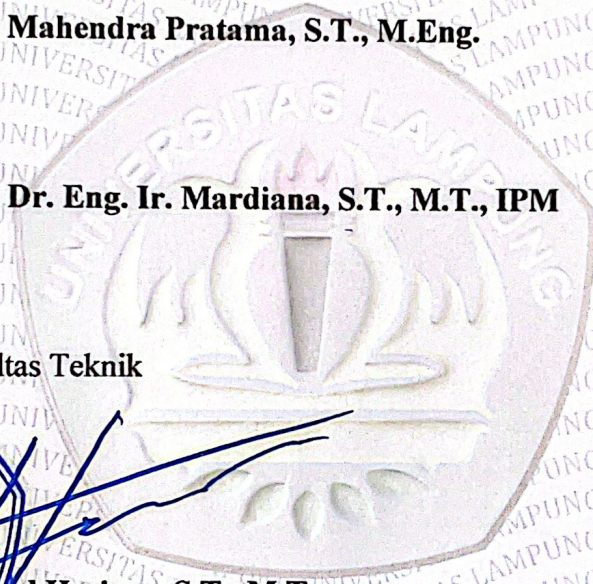
**Penguji : Dr. Eng. Ir. Mardiana, S.T., M.T., IPM**

**2. Dekan Fakultas Teknik**

**Dr. H. Alfiad Herison, S.T., M.T.**

196910302000031001

**Tanggal Lulus Ujian Skripsi : 3 Maret 2026**



*[Handwritten signature]*  
.....

*[Handwritten signature]*  
.....

*[Handwritten signature]*  
.....

*[Handwritten signature]*

## SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa skripsi berjudul "Pengembangan *Backend* pada Sistem *Monitoring* Skripsi di Jurusan Perikanan Dan Kelautan Universitas Lampung" sepenuhnya merupakan hasil karya saya sendiri. Apabila di kemudian hari terbukti pernyataan ini tidak benar, saya siap menerima sanksi sesuai dengan ketentuan hukum yang berlaku.

Bandar Lampung, 3 Maret 2026

Penulis,



Dimas Hidayanto Naim

NPM. 2115061035

## RIWAYAT HIDUP



Penulis dilahirkan di Lampung Selatan pada tanggal 24 Juli 2003 sebagai anak ketiga dari tiga bersaudara, dari pasangan Bapak Juwahir dan Ibu Faizah. Penulis menyelesaikan jenjang pendidikan formal di SDN 1 Sukapura pada tahun 2015, kemudian melanjutkan ke MTS GUPPI 03 Blangah dan lulus pada tahun 2018, serta menamatkan pendidikan menengah atas di SMAN 1 Sragi pada tahun 2021. Pada tahun 2021, penulis diterima sebagai mahasiswa Program Studi Teknik Informatika Universitas Lampung melalui jalur seleksi SBMPTN. Selama masa perkuliahan, penulis aktif berpartisipasi dalam berbagai kegiatan, antara lain:

1. Mengikuti kegiatan Magang Bersertifikat dari Kementerian Pendidikan dan kebudayaan di mitra Bakrie *Center Foundation* sebagai Divisi IT dan *Database* pada tahun 2024.
2. Mengikuti kegiatan Studi Independen Bersertifikat dari Kementerian Pendidikan dan Kebudayaan di mitra Dicoding Indonesia pada tahun 2023.
3. Melaksanakan Kuliah Kerja Nyata (KKN) di Desa Karang Agung, Kecamatan Pakuan Ratu, Kabupaten Way Kanan, Provinsi Lampung.

## **MOTTO**

*“The only impossible journey is the one you never begin”*

**(Tony Robbins)**

“Jadilah seperti kopi, meski pahit, tetap dinikmati dan memberi inspirasi”

**(Penulis)**

## **PERSEMBAHAN**

Puji dan syukur penulis panjatkan ke hadirat Allah SWT atas limpahan rahmat, taufik, dan hidayah-Nya, sehingga penulis dapat menyelesaikan penelitian dan penyusunan skripsi ini dengan baik.

Skripsi ini penulis persembahkan kepada:

“Ibu dan Almarhum Ayah, yang senantiasa memberikan doa, kasih sayang, serta teladan hidup kepada penulis, di mana doa dan pengorbanan beliau menjadi sumber kekuatan dan motivasi dalam menyelesaikan skripsi ini, serta kakak yang selalu memberikan dukungan, perhatian, dan semangat kepada penulis selama proses perkuliahan hingga penyusunan skripsi.”

## SANWACANA

Puji dan syukur penulis panjatkan ke hadirat Allah SWT atas limpahan rahmat, taufik, dan hidayah-Nya, sehingga penulis dapat menyelesaikan penyusunan skripsi yang berjudul “Pengembangan *Backend* pada Sistem *Monitoring* Skripsi di Jurusan Perikanan Dan Kelautan Universitas Lampung”. Selama proses penyusunan skripsi ini, penulis memperoleh banyak dukungan, bimbingan, serta bantuan, baik secara moril maupun materil, dari berbagai pihak. Oleh karena itu, dengan penuh rasa hormat dan terima kasih, penulis menyampaikan penghargaan yang sebesar-besarnya kepada:

1. Kedua orang tua dan keluarga, yang senantiasa memberikan doa, dukungan, dan nasihat kepada penulis selama menempuh pendidikan dan proses penyusunan skripsi ini;
2. Bapak Dr. H. Ahmad Herison, S.T., M.T., selaku Dekan Fakultas Teknik Universitas Lampung;
3. Ibu Herlinawati, S.T., M.T., selaku Ketua Jurusan Teknik Elektro Universitas Lampung;
4. Ibu Yessi Mulyani, S.T., M.T., selaku Ketua Program Studi Teknik Informatika Universitas Lampung;
5. Bapak Wahyu Eko Sulistiono, S.T., M.Sc., selaku Pembimbing Utama yang telah meluangkan waktu, memberikan bimbingan, arahan, serta saran kepada penulis selama pelaksanaan penelitian dan penyusunan skripsi;
6. Bapak Mahendra Pratama, S.T., M. Eng., selaku Pembimbing Pendamping yang telah memberikan bimbingan, masukan, dan saran kepada penulis dalam menyelesaikan penelitian dan penulisan skripsi;
7. Ibu Dr. Eng. Ir. Mardiana, S.T., M.T., IPM., selaku Penguji yang telah memberikan berbagai saran dan masukan yang membangun terhadap penelitian ini;

8. Ibu Dr. Munti Sarida, S.Pi., M.Sc., selaku Ketua Jurusan Perikanan dan Kelautan Universitas Lampung yang telah meluangkan waktu dan memberikan kesempatan kepada penulis dalam pelaksanaan penelitian ini.
9. Seluruh Dosen Program Studi Teknik Informatika Universitas Lampung yang telah memberikan ilmu dan bimbingan selama masa perkuliahan;
10. Nia Nurma Yunita, terima kasih telah menjadi bagian dalam proses perkuliahan hingga pada tahap penyusunan skripsi yang telah berkontribusi baik pikiran, tenaga, waktu, memberikan dukungan, semangat, serta selalu menemani dalam keadaan apapun;
11. Gibran Alfarabi, sebagai rekan satu tim penelitian yang telah berkerja sama dan saling mendukung dalam proses pengembangan sistem;
12. Rekan-rekan dalam grup “Teh Kotak” yang tidak dapat disebutkan satu per satu, atas dukungan dan kebersamaan yang diberikan kepada penulis selama masa perkuliahan;
13. Teman-teman KKN Universitas Lampung Periode I 2024 Desa Karang Agung yang telah kebersamai penulis selama pelaksanaan KKN;
14. Semua pihak yang terlibat dalam menyelesaikan skripsi ini.

Semoga Allah SWT selalu melimpahkan rahmat dan hidayah-Nya kepada kita semua, dan semoga skripsi ini dapat memberikan manfaat dan menjadi referensi bagi penelitian selanjutnya.

Bandar Lampung, 3 Maret 2026  
Penulis,

Dimas Hidayanto Naim  
NPM 2115061035

## DAFTAR ISI

Halaman

<b>DAFTAR ISI</b> .....	<b>xii</b>
<b>DAFTAR TABEL</b> .....	<b>xv</b>
<b>DAFTAR GAMBAR</b> .....	<b>xvi</b>
<b>I. PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Tujuan Penelitian .....	3
1.4 Manfaat Penelitian .....	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Penulisan .....	4
<b>II. TINJAUAN PUSTAKA</b> .....	<b>5</b>
2.1 Sistem Monitoring.....	5
2.2 Skripsi .....	5
2.3 <i>Unified Modelling Language (UML)</i> .....	6
2.4 <i>Entity Relationship Diagram (ERD)</i> .....	7
2.5 <i>Node JS</i> .....	9
2.6 <i>Express JS</i> .....	9
2.7 <i>Visual Studio Code</i> .....	10
2.8 <i>Scrum</i> .....	10
2.9 <i>Trello</i> .....	11
2.10 <i>Docker</i> .....	11
2.11 <i>RESTful API</i> .....	12

2.12 Redis.....	12
2.13 K6.....	13
2.14 Nginx.....	13
2.15 PostgreSQL .....	13
2.16 Penelitian Terkait .....	14
2.16.1 Sistem Manajemen dan Monitoring Bimbingan Tugas Akhir Berbasis Web .....	14
2.16.2 Sistem <i>Monitoring</i> Skripsi Berbasis <i>Progressive Web Application</i> (PWA) dengan <i>Push Notification</i> .....	15
2.16.3 Sistem Informasi <i>Monitoring</i> Skripsi Mahasiswa Berbasis <i>Website</i> .....	15
2.16.4 <i>Design and Development of a System for Managing Student Research: A Case Study for the School of Information Technology, Mapua University</i> 16	
2.16.5 <i>Development of a Framework to Implement a Web-Based Thesis Management System in Colleges of Education in Nigeria</i> .....	17
2.16.6 <i>THESISIT: Web-Based University Thesis Management Portal with a Defense Scheduling System</i> .....	18
<b>III. METODOLOGI PENELITIAN .....</b>	<b>20</b>
3.1 Waktu dan Tempat Penelitian .....	20
3.2 Alat Penelitian.....	21
3.3 Metode Penelitian.....	21
3.4 Tahapan Penelitian .....	22
3.4.1 Analisis Kebutuhan .....	24
3.4.2 Tahapan Pengembangan Sistem.....	41
<b>IV. HASIL DAN PEMBAHASAN .....</b>	<b>44</b>
4.1 Perancangan Sistem .....	44
4.1.1 <i>Entity Relationship Diagram</i> .....	44
4.1.2 Penerapan Kontainerisasi Menggunakan <i>Docker</i> .....	47
4.2 <i>Product Backlog</i> .....	50
4.2.1 <i>Product Backlog</i> Awal .....	50
4.2.2 Perkembangan <i>Product Backlog</i> .....	61
4.3 <i>Sprint</i> .....	64

4.3.1 <i>Sprint 1</i> .....	64
4.3.2 <i>Sprint 2</i> .....	77
4.3.3 <i>Sprint 3</i> .....	90
4.3.4 <i>Sprint 4</i> .....	110
4.3.5 <i>Sprint 5</i> .....	126
4.3.6 <i>Sprint 6</i> .....	142
4.3.7 <i>Sprint 7</i> .....	155
4.3.8 <i>Sprint 8</i> .....	170
4.3.9 <i>Sprint 9</i> .....	184
4.4 <i>End Product Sistem Backend</i> .....	202
4.5 <i>Deployment</i> .....	202
4.5.1 <i>Proses Deployment</i> .....	203
4.5.2 <i>Arsitektur Deployment Sistem</i> .....	205
4.6 <i>Pengujian Performa API</i> .....	206
4.6.1 <i>Tujuan Pengujian</i> .....	207
4.6.2 <i>Konfigurasi Skenario Pengujian</i> .....	207
4.6.3 <i>Hasil Pengujian Beban Normal</i> .....	209
4.6.4 <i>Hasil Pengujian Beban Ekstrem</i> .....	210
4.6.5 <i>Kesimpulan Pengujian Performa API</i> .....	211
4.7 <i>Pembahasan</i> .....	211
4.7.1 <i>Perancangan dan Pengembangan Sistem Monitoring Skripsi</i> .....	211
4.7.2 <i>Implementasi Backend Sistem Monitoring Skripsi</i> .....	213
4.7.3 <i>Penerapan Arsitektur RESTful API pada Backend</i> .....	214
4.7.4 <i>Efisiensi Sistem Monitoring Skripsi</i> .....	214
<b>V. KESIMPULAN DAN SARAN</b> .....	<b>216</b>
5.1 <i>Kesimpulan</i> .....	216
5.2 <i>Saran</i> .....	217
<b>DAFTAR PUSTAKA</b> .....	<b>218</b>
<b>LAMPIRAN</b> .....	<b>222</b>

## DAFTAR TABEL

Tabel	Halaman
3.1 Jadwal Penelitian.....	20
3.2 Alat ( <i>Hardware dan Software</i> ) pengembangan sistem .....	21
3.3 Tim <i>Scrum</i> .....	22
3.4 Kebutuhan Fungsional .....	25
3.5 Kebutuhan <i>Non-Fungsional</i> .....	26
3.6 Definisi Aktor <i>Use Case</i> .....	28
3.7 Definisi <i>Use Case</i> .....	28
3.8 Deskripsi Arsitektur Sistem .....	40
4.1 Entitas pada ERD .....	45
4.2 <i>Product Backlog</i> Awal .....	50
4.3 Perkembangan <i>Product Backlog</i> pada <i>Sprint 3</i> .....	61
4.4 Perkembangan <i>Product Backlog</i> pada <i>Sprint 5</i> .....	63
4.5 <i>Sprint Planning 1</i> .....	65
4.6 <i>Sprint Planning 2</i> .....	79
4.7 <i>Sprint Planning 3</i> .....	91
4.8 <i>Sprint Planning 4</i> .....	112
4.9 <i>Sprint Planning 5</i> .....	128
4.10 <i>Sprint Planning 6</i> .....	143
4.11 <i>Sprint Planning 7</i> .....	157
4.12 <i>Sprint Planning 8</i> .....	171
4.13 <i>Sprint Planning 9</i> .....	186
4.14 Konfigurasi Skenario Pengujian Beban Normal .....	208
4.15 Konfigurasi Skenario Pengujian Beban Ekstrem.....	208

## DAFTAR GAMBAR

Gambar	Halaman
2.1 Komponen <i>Use Case Diagram</i> .....	6
2.2 Komponen <i>Activity Diagram</i> .....	7
2.3 Simbol <i>Entity</i> ERD.....	8
2.4 Simbol <i>Relationship</i> ERD .....	8
2.5 Alur Kerangka Kerja <i>Scrum</i> .....	11
3.1 Tahapan Penelitian .....	23
3.2 <i>Use Case Diagram</i> .....	27
3.3 <i>Activity Diagram</i> Registrasi Akun .....	31
3.4 <i>Activity Diagram</i> Login.....	31
3.5 <i>Activity Diagram</i> Mengajukan Usul Judul Penelitian.....	32
3.6 <i>Activity Diagram</i> Mengajukan Seminar Usul Penelitian .....	32
3.7 <i>Activity Diagram</i> Mengajukan Seminar Hasil Penelitian .....	33
3.8 <i>Activity Diagram</i> Mengajukan Ujian Skripsi.....	33
3.9 <i>Activity Diagram</i> Mengirimkan Draf Laporan Skripsi .....	34
3.10 <i>Activity Diagram</i> Melihat Daftar Mahasiswa Bimbingan.....	34
3.11 <i>Activity Diagram</i> Melihat Riwayat Bimbingan.....	35
3.12 <i>Activity Diagram</i> Memverifikasi Pengajuan Seminar Usul .....	35
3.13 <i>Activity Diagram</i> Memverifikasi Pengajuan Seminar Hasil .....	36
3.14 <i>Activity Diagram</i> Memverifikasi Pengajuan Ujian Skripsi.....	36
3.15 <i>Activity Diagram</i> Verifikasi Pengajuan Usul Judul .....	37
3.16 <i>Activity Diagram</i> Melihat Progres Skripsi .....	37
3.17 <i>Activity Diagram</i> Mengelola Jadwal Seminar.....	38
3.18 <i>Activity Diagram</i> Mengajukan Layak Cetak Skripsi .....	38
3.19 <i>Activity Diagram</i> Memverifikasi Pengajuan Layak Cetak Skripsi .....	39

3.20	Gambaran Umum Sistem .....	39
3.21	Arsitektur Sistem <i>Backend</i> .....	40
4.1	<i>Entity Relationship Diagram</i> .....	44
4.2	Konfigurasi file <i>docker-compose.yml</i> .....	48
4.3	<i>Request</i> dan <i>response</i> API registrasi mahasiswa .....	67
4.4	<i>Request</i> dan <i>response</i> API verifikasi <i>email</i> .....	68
4.5	<i>Request</i> dan <i>response</i> API <i>login</i> .....	69
4.6	<i>Request</i> dan <i>response</i> API <i>refresh</i> token .....	70
4.7	<i>Request</i> dan <i>response</i> API <i>forgot password</i> .....	71
4.8	<i>Request</i> dan <i>response</i> API <i>reset password</i> .....	71
4.9	<i>Request</i> dan <i>response</i> API <i>logout</i> .....	72
4.10	<i>Request</i> dan <i>response</i> API <i>get all users</i> (admin sistem) .....	73
4.11	<i>Request</i> dan <i>response</i> API <i>create user</i> (admin sistem).....	74
4.12	<i>Request</i> dan <i>response</i> API <i>resend reset password</i> .....	75
4.13	<i>Request</i> dan <i>response</i> API <i>upload</i> foto mahasiswa.....	80
4.14	<i>Request</i> dan <i>response</i> API <i>update</i> profile mahasiswa.....	81
4.15	<i>Request</i> dan <i>response</i> API melihat profil mahasiswa .....	82
4.16	<i>Request</i> dan <i>response</i> <i>update</i> profile dosen .....	83
4.17	<i>Route</i> membuat pengajuan judul .....	84
4.18	Kode program konfigurasi <i>middleware upload</i> file .....	85
4.19	Kode program membuat pengajuan judul oleh mahasiswa.....	86
4.20	<i>Request Body</i> API membuat pengajuan judul .....	88
4.21	<i>Response</i> API membuat pengajuan judul.....	88
4.22	Kode program <i>update</i> pengajuan judul .....	93
4.23	<i>Request</i> dan <i>response</i> API <i>update</i> pengajuan judul .....	95
4.24	Kode program hapus pengajuan judul.....	97
4.25	<i>Request</i> dan <i>response</i> API hapus pengajuan judul.....	98
4.26	Kode program lihat pengajuan judul mahasiswa .....	99
4.27	<i>Request</i> dan <i>response</i> API lihat pengajuan judul mahasiswa .....	100
4.28	Kode program detail pengajuan judul .....	101
4.29	<i>Request</i> dan <i>response</i> API detail pengajuan judul .....	103
4.30	Kode program daftar pengajuan judul mahasiswa prodi.....	104

4.31 <i>Request</i> dan <i>response</i> API daftar pengajuan mahasiswa prodi .....	106
4.32 Kode program verifikasi pengajuan judul .....	107
4.33 <i>Request</i> dan <i>response</i> API verifikasi pengajuan judul .....	109
4.34 <i>Request</i> dan <i>response</i> API manajemen kuota dosen .....	114
4.35 <i>Request</i> dan <i>response</i> API melihat kuota dosen .....	115
4.36 <i>Request</i> dan <i>response</i> API penetapan dosen pembimbing/penguji .....	116
4.37 <i>Request</i> dan <i>response</i> API buat bimbingan mahasiswa .....	118
4.38 <i>Request</i> dan <i>response</i> API <i>update</i> bimbingan oleh dosen .....	119
4.39 <i>Request</i> dan <i>response</i> API <i>update</i> bimbingan oleh mahasiswa .....	120
4.40 <i>Request</i> dan <i>response</i> API daftar bimbingan .....	121
4.41 <i>Request</i> dan <i>response</i> API detail bimbingan .....	123
4.42 <i>Request</i> dan <i>response</i> API hapus file bimbingan .....	124
4.43 <i>Request</i> dan <i>response</i> API daftar file bimbingan .....	125
4.44 <i>Request</i> dan <i>response</i> API membuat pengajuan seminar usul .....	131
4.45 <i>Request</i> dan <i>response</i> API <i>update</i> pengajuan seminar usul .....	132
4.46 <i>Request</i> dan <i>response</i> API hapus pengajuan seminar usul .....	133
4.47 <i>Request</i> dan <i>response</i> API lihat pengajuan seminar usul mahasiswa .....	134
4.48 <i>Request</i> dan <i>response</i> API detail pengajuan seminar usul .....	136
4.49 <i>Request</i> dan <i>response</i> API daftar pengajuan seminar usul .....	137
4.50 <i>Request</i> dan <i>response</i> API verifikasi pengajuan seminar usul .....	138
4.51 <i>Request</i> dan <i>response</i> API membuat surat pengajuan seminar usul .....	139
4.52 <i>Request</i> dan <i>response</i> API unduh surat pengajuan seminar usul .....	140
4.53 <i>Request</i> dan <i>response</i> API membuat jadwal seminar .....	146
4.54 <i>Request</i> dan <i>response</i> API melihat daftar jadwal seminar .....	147
4.55 <i>Request</i> dan <i>response</i> API <i>update</i> jadwal seminar .....	148
4.56 <i>Request</i> dan <i>response</i> API hapus jadwal seminar .....	149
4.57 <i>Request</i> dan <i>response</i> API <i>upload</i> dokumen seminar .....	150
4.58 <i>Request</i> dan <i>response</i> API <i>update</i> dokumen seminar .....	151
4.59 <i>Request</i> dan <i>response</i> API hapus jadwal seminar .....	152
4.60 <i>Request</i> dan <i>response</i> API detail dokumen seminar .....	153
4.61 <i>Request</i> dan <i>response</i> API verifikasi dokumen seminar .....	154
4.62 <i>Request</i> dan <i>response</i> API membuat pengajuan seminar hasil .....	160

4.63 <i>Request</i> dan <i>response</i> API <i>update</i> pengajuan seminar hasil .....	161
4.64 <i>Request</i> dan <i>response</i> API menghapus pengajuan seminar hasil.....	162
4.65 <i>Request</i> dan <i>response</i> API melihat pengajuan seminar hasil mahasiswa ...	163
4.66 <i>Request</i> dan <i>response</i> API melihat daftar pengajuan seminar hasil oleh admin prodi .....	164
4.67 <i>Request</i> dan <i>response</i> API detail pengajuan seminar hasil .....	165
4.68 <i>Request</i> dan <i>response</i> API verifikasi pengajuan seminar hasil.....	166
4.69 <i>Request</i> dan <i>response</i> API membuat surat pengajuan seminar hasil .....	167
4.70 <i>Request</i> dan <i>response</i> API unduh surat undangan seminar hasil .....	168
4.71 <i>Request</i> dan <i>response</i> API membuat pengajuan ujian skripsi.....	174
4.72 <i>Request</i> dan <i>response</i> API <i>update</i> pengajuan ujian skripsi.....	175
4.73 <i>Request</i> dan <i>response</i> API menghapus pengajuan ujian skripsi .....	176
4.74 <i>Request</i> dan <i>response</i> API melihat pengajuan ujian skripsi mahasiswa .....	177
4.75 <i>Request</i> dan <i>response</i> API melihat daftar pengajuan ujian skripsi oleh admin prodi .....	178
4.76 <i>Request</i> dan <i>response</i> API melihat detail pengajuan ujian skripsi.....	180
4.77 <i>Request</i> dan <i>response</i> API verifikasi pengajuan ujian skripsi .....	181
4.78 <i>Request</i> dan <i>response</i> API membuat surat pengajuan ujian skripsi.....	182
4.79 <i>Request</i> dan <i>response</i> API unduh surat undangan ujian skripsi.....	183
4.80 <i>Request</i> dan <i>response</i> API membuat pengajuan layak cetak skripsi.....	189
4.81 <i>Request</i> dan <i>response</i> API <i>update</i> pengajuan layak cetak skripsi.....	190
4.82 <i>Request</i> dan <i>response</i> API menghapus pengajuan layak cetak skripsi .....	191
4.83 <i>Request</i> dan <i>response</i> API melihat pengajuan layak cetak skripsi mahasiswa .....	192
4.84 <i>Request</i> dan <i>response</i> API melihat daftar pengajuan layak cetak skripsi oleh ketua jurusan .....	194
4.85 <i>Request</i> dan <i>response</i> API melihat detail pengajuan layak cetak skripsi....	195
4.86 <i>Request</i> dan <i>response</i> verifikasi pengajuan layak cetak skripsi.....	196
4.87 <i>Request</i> dan <i>response</i> membuaat surat pengajuan layak cetak skripsi .....	197
4.88 <i>Request</i> dan <i>response</i> unduh surat keterangan layak cetak skripsi .....	198
4.89 <i>Request</i> dan <i>response</i> API <i>monitoring</i> kelulusan.....	200
4.90 Diagram Arsitektur <i>Deployment</i> .....	205

4.91 Hasil pengujian beban normal.....	209
4.92 Hasil pengujian beban ekstrem .....	210
4.93 Diagram komponen arsitektur <i>backend</i> sistem <i>monitoring</i> skripsi.....	212

## **I. PENDAHULUAN**

### **1.1 Latar Belakang**

Perkembangan Ilmu Pengetahuan dan Teknologi (IPTEK), khususnya teknologi informasi, memberikan pengaruh yang signifikan dalam berbagai aspek kehidupan dan mendorong kemajuan di bidang pendidikan [1]. Selain meningkatkan kualitas pembelajaran, pemanfaatan teknologi informasi dapat dimanfaatkan untuk mendukung pengelolaan proses akademik di perguruan tinggi, termasuk dalam hal monitoring penyusunan skripsi mahasiswa.

Skripsi merupakan karya tulis ilmiah yang wajib diselesaikan oleh mahasiswa program sarjana sebagai salah satu syarat untuk memperoleh gelar akademik. Ketentuan mengenai pelaksanaan dan penyusunan skripsi tersebut telah diatur dalam Peraturan Akademik Universitas Lampung. Dalam proses penyusunan skripsi, pemantauan progres mahasiswa menjadi aspek penting untuk memastikan pelaksanaan tugas akhir berjalan sesuai dengan standar dan ketentuan akademik yang berlaku.

Berdasarkan studi pendahuluan berupa wawancara yang dilakukan dengan Ketua Jurusan Perikanan dan Kelautan Universitas Lampung pada tanggal 5 Mei 2025, diketahui bahwa proses pemantauan progres skripsi mahasiswa hingga saat ini masih dilakukan secara manual dan belum didukung oleh sistem pencatatan terpusat. Mahasiswa dan dosen pembimbing mencatat perkembangan skripsi secara terpisah, sehingga informasi penting seperti progres penyusunan laporan dan tanggal pengajuan judul tidak terdokumentasi secara sistematis. Akibatnya, ketika pihak jurusan membutuhkan data perkembangan skripsi mahasiswa, dosen pembimbing harus melakukan konfirmasi ulang kepada masing-masing mahasiswa,

yang berdampak pada keterlambatan pengumpulan data dan berpotensi menimbulkan kehilangan informasi. Kondisi tersebut menyulitkan pihak jurusan dalam memantau perkembangan skripsi mahasiswa secara menyeluruh, serta menghambat deteksi dini terhadap mahasiswa yang mengalami stagnasi dalam penyusunan skripsi, sehingga pendampingan akademik tidak dapat dilakukan secara optimal.

Berdasarkan permasalahan yang telah diidentifikasi, diperlukan sebuah sistem *monitoring* skripsi berbasis *web* yang bertujuan untuk memfasilitasi pencatatan, dan pemantauan progres skripsi mahasiswa. Sistem ini memungkinkan dosen pembimbing dan mahasiswa untuk mencatat serta memperbarui progres skripsi, sementara ketua jurusan dapat memantau perkembangan tersebut secara langsung. Dengan adanya sistem ini, diharapkan proses pemantauan perkembangan skripsi mahasiswa dapat dilakukan secara lebih efisien.

Penelitian ini difokuskan pada pengembangan sisi *backend* dari sistem *monitoring* skripsi menggunakan teknologi *Node.js*, *Express.js*, dan *PostgreSQL*. *Node.js* dan *Express.js* dipilih karena memiliki komunitas pengembang yang kuat, performa yang baik, serta mendukung pengembangan aplikasi *web backend* secara efektif [2]. *PostgreSQL* digunakan sebagai sistem manajemen basis data karena bersifat bebas lisensi, mendukung pengelolaan basis data relasional, serta memiliki fitur yang memadai untuk penyimpanan dan pengelolaan data pada sistem informasi [3]. Dalam proses pengembangannya, penelitian ini menggunakan metodologi pengembangan perangkat lunak *Scrum*. *Metodologi Scrum* dipilih karena bersifat adaptif, fleksibel, dan mendukung proses pengembangan sistem secara cepat melalui tahapan iteratif dan inkremental [4].

## 1.2 Rumusan Masalah

Adapun rumusan masalah dari penelitian ini adalah sebagai berikut:

1. Bagaimana perancangan *arsitektur backend* pada sistem *monitoring* skripsi yang mampu mendukung efisiensi proses rekapitulasi data progres skripsi mahasiswa di Jurusan Perikanan dan Kelautan Universitas Lampung?

2. Bagaimana implementasi *backend* sistem *monitoring* skripsi dalam mendukung pengelolaan dan akses data progres skripsi mahasiswa secara terpusat?
3. Bagaimana penerapan arsitektur *RESTful* API pada *backend* sistem *monitoring* skripsi untuk mendukung integrasi dan pertukaran data dengan antarmuka *frontend*?

### **1.3 Tujuan Penelitian**

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Merancang arsitektur *backend* pada sistem *monitoring* skripsi yang mampu mendukung efisiensi proses rekapitulasi data progres skripsi mahasiswa di Jurusan Perikanan dan Kelautan Universitas Lampung.
2. Mengimplementasikan *backend* sistem *monitoring* skripsi untuk mendukung pengelolaan dan akses data progres skripsi mahasiswa secara terpusat.
3. Menerapkan arsitektur *RESTful* API pada *backend* sistem *monitoring* skripsi untuk mendukung integrasi dan pertukaran data dengan antarmuka *frontend*.

### **1.4 Manfaat Penelitian**

Adapun manfaat penelitian dari penelitian ini adalah sebagai berikut:

1. Mempermudah mahasiswa dalam mencatat progres skripsi mereka, serta memastikan bahwa setiap tahapan skripsi terdokumentasi dengan baik.
2. Memudahkan dosen dalam memantau perkembangan skripsi mahasiswa melalui sistem terpusat, tanpa harus bergantung pada pencatatan manual.
3. Memberikan data progres skripsi mahasiswa yang terdokumentasi dalam sistem sebagai bahan pendukung pengambilan keputusan akademik.

### **1.5 Batasan Masalah**

Adapun batasan masalah dari penelitian ini adalah sebagai berikut:

1. Sistem yang dikembangkan hanya digunakan untuk pemantauan progres skripsi mahasiswa di Jurusan Perikanan dan Kelautan Universitas Lampung.

2. *Backend* sistem dikembangkan menggunakan teknologi *Node.js*, *Express.js*, dan *PostgreSQL*, tanpa membahas implementasi teknologi *frontend* secara mendetail.
3. Penelitian ini tidak membahas aspek performa *backend* secara mendalam, seperti optimasi *query database*, skalabilitas sistem, atau beban *server*.

## 1.6 Sistematika Penulisan

Sistematika penulisan yang digunakan pada laporan penelitian ini adalah sebagai berikut:

### **BAB I : PENDAHULUAN**

Pada bab ini secara umum meliputi latar belakang penelitian, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, serta sistematika penulisan.

### **BAB II : TINJAUAN PUSTAKA**

Pada bab ini berisi tentang prinsip, dasar teori dan teori penunjang yang digunakan sebagai acuan dalam melakukan penelitian.

### **BAB III : METODOLOGI PENELITIAN**

Pada bab ini membahas mengenai metodologi penelitian yang digunakan dalam pengembangan *backend* pada sistem *monitoring* skripsi di Jurusan Perikanan dan Kelautan Universitas Lampung.

### **BAB IV : HASIL DAN PEMBAHASAN**

Pada bab ini membahas mengenai hasil serta pembahasan yang diperoleh dalam penelitian Pengembangan *backend* pada sistem *monitoring* skripsi di Jurusan Perikanan dan Kelautan Universitas Lampung, termasuk proses implementasi dan pengujian sistem.

### **BAB V : KESIMPULAN DAN SARAN**

Pada bab ini berisi tentang kesimpulan dan saran hasil penelitian.

### **DAFTAR PUSTAKA**

### **LAMPIRAN**

## II. TINJAUAN PUSTAKA

### 2.1 Sistem Monitoring

*Monitoring* merupakan proses yang dilakukan secara rutin melalui pengumpulan dan pencatatan data untuk mengukur kemajuan pencapaian tujuan program serta memantau perubahan yang terjadi pada proses dan hasil kegiatan [5]. Tujuan utama dari *monitoring* adalah untuk memastikan kepatuhan terhadap standar yang telah ditetapkan, mendeteksi masalah sejak dini, meningkatkan efisiensi dan efektivitas program serta menyediakan data untuk evaluasi.

Berdasarkan penjelasan tentang *monitoring* diatas, dapat disimpulkan bahwa sistem *monitoring* berarti suatu *software* yang digunakan untuk menganalisis dan mengumpulkan sebuah informasi yang telah ditentukan. Sistem *monitoring* yang dibuat dalam penelitian ini adalah sistem *monitoring* yang berguna untuk memantau proses skripsi mahasiswa. Informasi yang dimaksud mencakup pengajuan judul skripsi, bimbingan dengan dosen yang telah ditetapkan, serta mahasiswa dapat mengakses revisi yang diberikan oleh dosen pembimbing melalui sistem *monitoring* skripsi berbasis *website*.

### 2.2 Skripsi





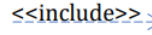
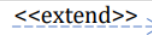
Skripsi atau thesis merupakan karya tulis ilmiah yang harus disusun oleh mahasiswa sebagai bagian dari persyaratan kelulusan untuk memperoleh gelar akademik pada pendidikan tinggi [6]. Menurut Kamus Besar Bahasa Indonesia (KBBI) versi Departemen Pendidikan Nasional (2021), skripsi merupakan karangan ilmiah yang wajib ditulis oleh mahasiswa sebagai bagian dari persyaratan akhir pendidikannya.

### 2.3 Unified Modelling Language (UML)

*Unified Modelling Language* (UML) adalah bahasa pemodelan visual untuk memodelkan persyaratan sistem, mendeskripsikan desain, dan menggambarkan detail implementasi. UML bukan hanya digunakan sebagai bahasa pemodelan untuk memvisualisasikan dan mendokumentasikan sistem perangkat lunak, tetapi juga telah diterima sebagai standar dalam pemodelan desain sistem berorientasi objek pada penelitian rekayasa perangkat lunak [7]. Bahasa ini direpresentasikan dalam bentuk diagram. Beberapa jenis diagram dalam UML meliputi:

#### a. Use Case Diagram

Dalam UML, *use case* digunakan untuk menangkap persyaratan fungsional aplikasi. Diagram *use case* berperan dalam memodelkan interaksi fungsional antara pengguna dan sistem, yang memungkinkan pengembang untuk memahami kebutuhan fungsional utama berdasarkan sudut pandang pengguna akhir, sehingga mendukung proses analisis kebutuhan sistem [8]. Gambar 2.1 di bawah ini menjelaskan komponen – komponen yang ada pada *use case diagram*.






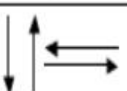
Simbol	Keterangan
	<b>Aktor:</b> mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<b>Use Case:</b> abstraksi atau interaksi antara system dengan aktor
	<b>Association:</b> abstraksi dari penghubung antara <i>actor</i> dengan <i>use case</i>
	<b>Generalisasi:</b> menunjukkan spesialisasi <i>actor</i> untuk dapat berpartisipasi dengan <i>use case</i>
	<b>Includes:</b> menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	<b>Extends:</b> menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

Gambar 2.1 Komponen *Use Case Diagram*

Sumber : *E-book* Dasar – Dasar Rekayasa Perangkat Lunak

### b. Activity Diagram

*Activity Diagram* merupakan salah satu komponen penting dalam rangkaian diagram yang digunakan dalam *Unified Modeling Language* (UML). Activity diagram berfungsi untuk memodelkan urutan tindakan dalam suatu alur proses dengan tujuan menangkap aktivitas serta hasil yang dihasilkan dari proses tersebut. Diagram ini digunakan untuk menggambarkan aktivitas dan alur kerja dalam suatu sistem, baik pada implementasi suatu operasi maupun aktivitas yang terjadi dalam sebuah use case [9]. Gambar 2.2 di bawah ini menjelaskan komponen – komponen yang ada pada *activity diagram*.

NO	GAMBAR	NAMA	KETERANGAN
1		Activity	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		Action	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		Initial Node	Bagaimana objek dibentuk atau diawali.
4		Activity Final Node	Bagaimana objek dibentuk dan diakhiri
5		Decision	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu
6		Line Connector	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya

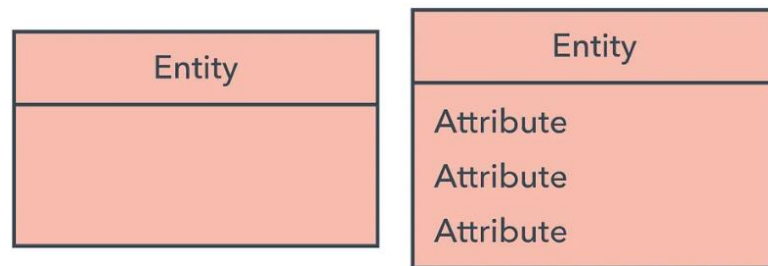
Gambar 2.2 Komponen *Activity Diagram*

Sumber : Rasiban dkk., 2024

### 2.4 Entity Relationship Diagram (ERD)

*Entity Relationship Diagram* (ERD) adalah salah satu teknik pemodelan data pertama yang dikembangkan dan masih banyak digunakan untuk merepresentasikan entitas, atribut, dan relasi antar entitas dalam sistem basis data [10]. Dengan kata lain, ERD adalah model konseptual yang digunakan untuk menggambarkan hubungan antar data melalui entitas dan relasi yang saling

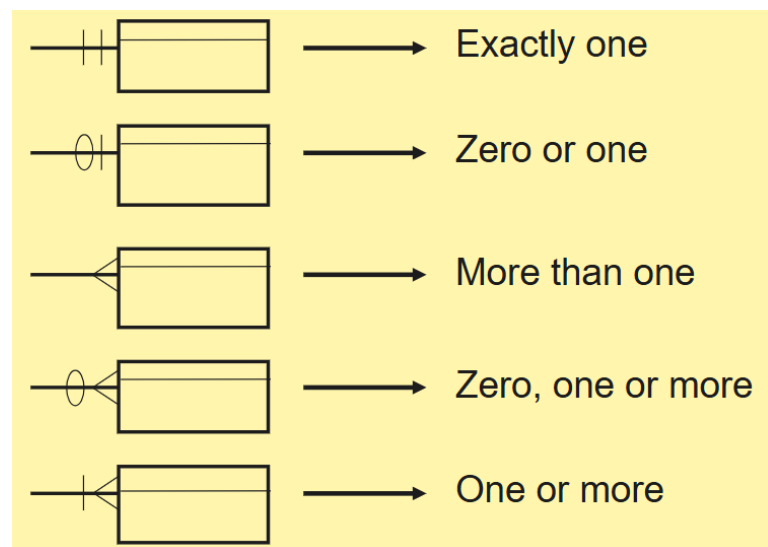
terhubung. Pada penelitian ini ERD digambarkan menggunakan jenis relationship ERD versi Martin yang digambarkan dengan garis dan menambahkan *connectivity*-nya. Simbol *entity set* pada ERD versi Martin ditunjukkan Gambar 2.3 dimana terbuat dari segi empat yang dilengkapi dengan nama *entity set* dan nama-nama atribut yang tertulis di dalam segi empat tersebut.



Gambar 2.3 Simbol *Entity* ERD

Sumber : *E-book Desain Database dengan ERD dan LRS*

Sedangkan, gambar simbol-simbol *relationship* pada ERD dapat dilihat pada Gambar 2.4.



Gambar 2.4 Simbol *Relationship* ERD

Sumber : *E-book Desain Database dengan ERD dan LRS*

## 2.5 Node JS

*Node JS* adalah *runtime JavaScript* yang dibangun di mesin *JavaScript V8 Chrome* [2]. *Node.js* merupakan platform yang banyak digunakan dalam pengembangan aplikasi sisi server pada era *JavaScript modern*. Penting untuk dipahami bahwa *Node.js* berbeda dengan *JavaScript* itu sendiri. *Node.js* merupakan sebuah runtime environment yang dibangun di atas mesin *JavaScript Google Chrome V8* dan memungkinkan eksekusi kode *JavaScript* di sisi server. Dengan demikian, *Node.js* tidak hanya menggunakan bahasa *JavaScript*, tetapi juga menyediakan lingkungan dan fungsionalitas tambahan untuk pengembangan aplikasi server-side. Arsitektur *Node.js* menerapkan model event-driven dan non-blocking I/O, sehingga mendukung pengembangan aplikasi yang bersifat real-time dan skalabel [11].

## 2.6 Express JS

*Express JS* adalah *framework web* minimalis dan fleksibel yang berjalan di atas *Node.js* [12]. *Express JS* menyediakan berbagai fitur untuk membangun aplikasi *web* dan API dengan lebih cepat dan efisien, tanpa menghilangkan fleksibilitas yang dimiliki oleh *Node.js*. *Express JS* memiliki beberapa keunggulan yang membuatnya populer di kalangan *developer web*, sebagai berikut [12].

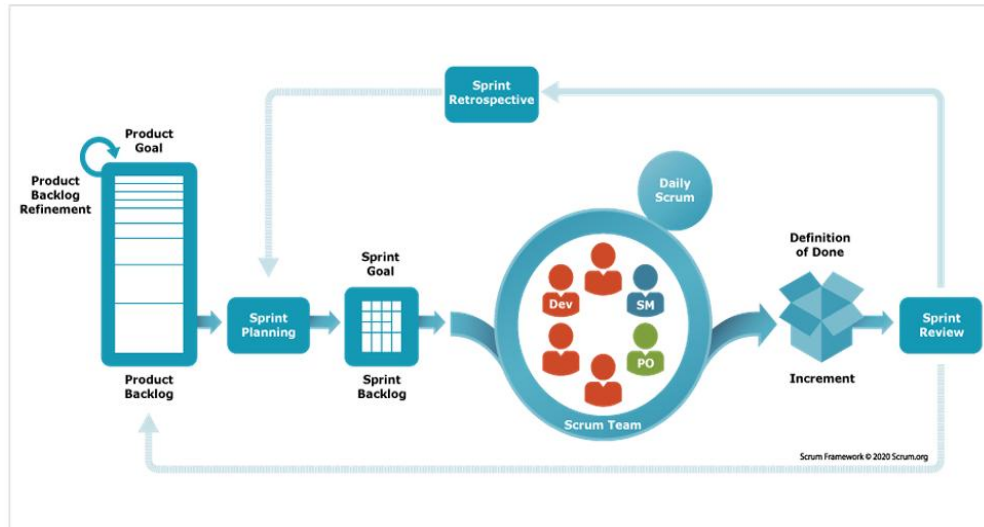
- a. Mendukung pembuatan *middleware* yang berperan menyediakan mekanisme penyaringan *HTTP Request* yang masuk ke aplikasi dengan kata lain setiap kali ada *Request* masuk, maka akan difilter oleh *middleware*.
- b. Mendukung berbagai *HTTP verb* seperti *POST*, *GET*, *PUT*, *DELETE*, *OPTION*, *HEAD*, dan lainnya.
- c. *Express JS* memberikan layanan untuk dapat memilih *template engine* yang diperlukan.
- d. *Express JS* dapat memanajemen file statik seperti *CSS* dan *Javascript*.
- e. *Express JS* mempunyai sifat fleksibel sehingga sangat bebas untuk dikostumisasi.

## 2.7 Visual Studio Code

*Visual Studio Code* adalah editor kode lintas *platform* yang tersedia untuk sistem operasi populer seperti *Windows*, *Linux*, dan *macOS*. Editor ini memiliki berbagai fitur bawaan, termasuk pelengkapan kode *IntelliSense* yang selalu aktif, pemahaman semantik kode yang mendalam, navigasi yang intuitif, dan pemfaktoran ulang kode yang efisien. *Visual Studio Code* mendukung pengembangan berbagai bahasa pemrograman seperti *JavaScript*, *TypeScript*, *C#*, *C++*, *Python*, *PHP*, serta *runtime* seperti *Unity* dan *.NET*. Editor ini juga mendukung teknologi *web* seperti *HTML*, *CSS*, *Sass*, *Less*, dan *JSON*. *VS Code* terintegrasi dengan berbagai paket dan repositori, serta memiliki dukungan *Git* bawaan yang menyediakan alur kerja pengembangan yang efisien. Fitur *debugging* adalah salah satu yang paling populer di *Visual Studio Code*. Selain itu, *editor* ini sangat fleksibel dan dapat dikustomisasi melalui berbagai ekstensi yang tersedia. *Visual Studio Code* juga terus diperbarui secara berkala untuk menjaga performa dan kompatibilitasnya [13].

## 2.8 Scrum

*Scrum* adalah kerangka kerja yang banyak digunakan dalam pengembangan sistem informasi karena mengadopsi pendekatan iteratif dan inkremental. Prinsip utama *scrum* adalah volatilitas persyaratan, yaitu mengakui fakta bahwa selama proses produksi, pelanggan dapat berubah pikiran tentang apa yang mereka inginkan dan butuhkan. Proses pengembangan dilakukan melalui siklus *sprint* yang terstruktur, di mana kebutuhan sistem dikelola dalam *Product Backlog* dan direalisasikan secara bertahap. Selain itu, aktivitas seperti *Daily Scrum* digunakan untuk memantau progres dan mengoordinasikan pekerjaan tim secara rutin [14]. *Scrum* memiliki nilai *Agile Manifesto* yang dilakukan secara teratur dan bertahap dalam jangka waktu yang ditentukan sehingga kualitas produk bernilai tinggi, produktif, dan inovatif [15].



Gambar 2.5 Alur Kerangka Kerja *Scrum*

Sumber : <https://www.scrum.org/resources/scrum-framework-poster>

## 2.9 Trello

*Trello* merupakan sebuah aplikasi yang berfungsi sebagai alat kolaborasi antar tim dalam pengembangan proyek. Salah satu fungsi *trello* adalah bisa melakukan pencatatan tentang list proses yang ingin dikerjakan dalam pengembangan proyek. Maka dari itu *trello* bisa berfungsi sebagai alat manajemen proyek di dalam metode *scrum* ini seperti pencatatan *product Backlog*, *Sprint Backlog*, dll. Selain itu *trello* juga bisa diakses via *website* dan aplikasi *mobile* [16].

## 2.10 Docker

*Docker* menyediakan mekanisme kontainerisasi dengan mengemas kode program, *runtime*, serta seluruh dependensi aplikasi ke dalam unit kontainer yang bersifat ringan dan terisolasi. Dengan pendekatan tersebut, aplikasi dapat dijalankan secara konsisten pada berbagai lingkungan, baik pada tahap pengembangan, pengujian, maupun implementasi. Kontainer *Docker* dijalankan di atas sistem operasi *host* dengan berbagi kernel yang sama, sehingga tidak memerlukan sistem operasi terpisah sebagaimana pada mesin virtual. Karakteristik ini menjadikan *Docker* banyak digunakan dalam lingkungan enterprise untuk mendukung proses

*deployment* aplikasi serta pemanfaatan sumber daya sistem secara lebih optimal, khususnya dalam pengelolaan dan penyediaan layanan aplikasi berbasis server. [17].

### **2.11 RESTful API**

RESTful API merupakan bentuk arsitektur dari *Application Programming Interface* (API) yang sering disebut juga sebagai REST API atau RESTful *Web Service*. REST, atau *Representational State Transfer*, adalah gaya arsitektur dan metode komunikasi yang banyak digunakan dalam pengembangan *web service* modern. Melalui RESTful API, berbagai sistem dapat saling berinteraksi dan bertukar data melalui endpoint dan format data yang telah ditentukan. Setiap *resource* yang tersimpan di dalam *database* dipetakan menjadi *endpoint* tertentu pada API, sehingga memudahkan proses akses dan pengelolaan data. RESTful API menggunakan perintah HTTP seperti *GET*, *POST*, *PUT*, dan *DELETE* untuk mengakses atau memodifikasi *resource*, sementara format JSON umumnya digunakan sebagai representasi data yang dikirim maupun diterima [18].

### **2.12 Redis**

Redis merupakan database *open-source* yang menyimpan dataset sepenuhnya dalam memori, sehingga memberikan kecepatan akses yang sangat tinggi. Mekanisme penyimpanan berbasis memori ini menjadikan Redis ideal untuk aplikasi yang memerlukan waktu respons rendah, seperti sistem *caching* dengan volume data besar. Redis dikenal karena performanya yang cepat, responsif, serta kemampuannya dalam menangani beragam tipe struktur data. Selain itu, Redis menawarkan manajemen *cache* yang fleksibel dan efisien, sehingga mampu meningkatkan kinerja aplikasi secara signifikan [19].

### 2.13 K6

K6 merupakan salah satu alat yang digunakan untuk melakukan pengujian beban (*load testing*) maupun pengujian stres (*stress testing*) pada aplikasi web dengan mensimulasikan jumlah akses atau interaksi pengguna terhadap sistem. Berdasarkan dokumentasi resmi k6, *platform* ini merupakan *tool* pengujian performa yang dirancang untuk kebutuhan rekayasa perangkat lunak, berjalan pada lingkungan *Node.js*, bersifat gratis, serta mudah digunakan. K6 juga bersifat *multiplatform* dan dapat dijalankan pada sistem operasi *Windows*, *Linux*, maupun *macOS* [20].

### 2.14 Nginx

Nginx merupakan perangkat lunak *web server* berbasis *open-source* yang pada awalnya dikembangkan untuk menangani layanan HTTP. Seiring dengan perkembangan kebutuhan dan fitur yang disediakan, Nginx tidak hanya berfungsi sebagai server web, tetapi juga dapat difungsikan sebagai *reverse proxy* dan *load balancer*. Dalam peran tersebut, Nginx digunakan untuk meneruskan permintaan dari klien ke server backend serta mendistribusikan lalu lintas permintaan pada sistem berbasis web, sehingga mendukung pengelolaan akses layanan dalam arsitektur aplikasi modern [21].

### 2.15 PostgreSQL

PostgreSQL merupakan sistem manajemen basis data relasional (*RDBMS*) yang bersifat *open source* dan mendukung pengelolaan data secara aman serta kompleks. PostgreSQL menggunakan dan memperluas bahasa SQL serta mendukung data relasional dan non-relasional seperti JSON. Sistem basis data ini dikembangkan sejak tahun 1982 dan terus dikembangkan oleh komunitas global sehingga memiliki tingkat stabilitas, keandalan, dan integritas data yang tinggi. Selain itu, PostgreSQL menyediakan fitur keamanan, skalabilitas, serta dapat digunakan tanpa biaya

lisensi, sehingga sesuai untuk pengembangan aplikasi berbasis *web* yang membutuhkan pengelolaan data yang andal dan fleksibel [3].

## **2.16 Penelitian Terkait**

Studi-studi yang relevan dengan pengembangan *backend* untuk sistem *monitoring* skripsi merupakan referensi penting dalam penelitian ini. Penelitian-penelitian tersebut berfungsi sebagai acuan dalam perancangan, implementasi, dan pengujian sistem *backend* yang dikembangkan. Berikut adalah beberapa jurnal atau penelitian yang berhubungan langsung dengan topik ini.

### **2.16.1 Sistem Manajemen dan *Monitoring* Bimbingan Tugas Akhir Berbasis *Web***

Penelitian yang dilakukan oleh Salmi dan Darmatasia [22] dari Universitas Islam Negeri Alauddin Makassar bertujuan untuk merancang dan membangun sistem manajemen serta *monitoring* bimbingan tugas akhir berbasis *web* guna mengatasi kendala dalam proses pembimbingan skripsi antara dosen dan mahasiswa. Permasalahan yang diangkat dalam penelitian ini meliputi perbedaan aktivitas dan kesibukan antara dosen dan mahasiswa yang sering menghambat kelancaran proses bimbingan tugas akhir. Metode pengembangan sistem yang digunakan pada penelitian ini adalah *Waterfall*, yang mencakup tahapan analisis kebutuhan, perancangan sistem, implementasi, pengujian, hingga pemeliharaan sistem. Sistem yang dikembangkan menyediakan fitur pengajuan judul, penetapan dosen pembimbing, pencatatan aktivitas bimbingan, pengunggahan dokumen tugas akhir, serta fasilitas monitoring progres bimbingan oleh admin program studi. Hasil penelitian menunjukkan bahwa sistem manajemen dan monitoring bimbingan tugas akhir yang dibangun dapat membantu mahasiswa dan dosen dalam menjalankan proses bimbingan secara daring serta memberikan kemudahan bagi pihak administrasi dalam memantau perkembangan tugas akhir mahasiswa. Berdasarkan hasil pengujian kelayakan menggunakan kuesioner, sistem memperoleh tingkat penerimaan sebesar 80%, yang menunjukkan bahwa sistem telah berjalan sesuai

dengan tujuan penelitian dan dapat digunakan untuk mendukung proses bimbingan tugas akhir mahasiswa.

### **2.16.2 Sistem *Monitoring Skripsi* Berbasis *Progressive Web Application* (PWA) dengan *Push Notification***

Penelitian yang dilakukan oleh Herdiesel Santoso, Wahyu Widodo, dan Muhammad Alvian Rizky [23] dari STMIK El Rahma bertujuan untuk mengembangkan sistem monitoring skripsi berbasis *Progressive Web Application* (PWA) guna mendukung pelaksanaan bimbingan skripsi secara *online*, terutama selama masa pandemi *Covid-19*. Penelitian ini menggunakan metode *Mobile-D* dalam pengembangan perangkat lunak, yang mencakup tahapan perancangan arsitektur sistem, desain antarmuka, implementasi, integrasi modul, dan pengujian sistem. Hasil pengujian menunjukkan bahwa sistem yang dikembangkan berjalan dengan baik tanpa adanya *bug* atau *error*, baik pada perangkat laptop maupun *smartphone*. Sistem ini membantu mahasiswa dan dosen pembimbing dalam melakukan *monitoring* skripsi secara efisien dan *real-time*, serta meningkatkan efektivitas komunikasi melalui fitur *push notification*. Saran untuk pengembangan lebih lanjut mencakup peningkatan keamanan sistem, optimalisasi performa aplikasi, dan integrasi dengan sistem akademik universitas.

### **2.16.3 Sistem Informasi *Monitoring Skripsi Mahasiswa* Berbasis *Website***

Penelitian yang dilakukan oleh Sri Ayu Ningsi dan Baharuddin Rahman [24] dari STMIK Catur Sakti Kendari bertujuan untuk mengembangkan sistem informasi *monitoring* skripsi berbasis *website* guna memantau dan menyajikan informasi terkait pelaksanaan skripsi mahasiswa. Sistem ini mencakup proses seminar proposal, seminar hasil, dan ujian tutup, serta membantu admin dalam mengelola dan menyampaikan informasi secara lebih efisien. Dengan adanya sistem ini, STMIK Catur Sakti Kendari memiliki basis data skripsi mahasiswa yang terstruktur dan dapat memberikan informasi mengenai rekap kehadiran mahasiswa dalam bimbingan, progres pengerjaan tugas akhir, serta riwayat hasil bimbingan. Sistem ini memastikan bahwa seluruh aktivitas bimbingan tercatat secara detail, akurat,

dan dapat dipantau dengan mudah. Hasil penelitian menunjukkan bahwa sistem yang dikembangkan dapat meningkatkan efisiensi *monitoring* skripsi, serta mempermudah mahasiswa dan dosen dalam melakukan proses bimbingan secara lebih sistematis. Saran untuk pengembangan lebih lanjut meliputi peningkatan fitur otomatisasi proses *monitoring*, integrasi dengan sistem akademik, serta pengembangan aplikasi *mobile* untuk akses yang lebih fleksibel.

#### **2.16.4 *Design and Development of a System for Managing Student Research: A Case Study for the School of Information Technology, Mapua University***

Penelitian yang dilakukan oleh Julio Jerison E. Macrohon, Aevin S. Bulaong, Joshua Claude A. Bonsol, John Andrei C. Gayeta, dan Jyh-Horng Jeng [25] bertujuan untuk mengembangkan sistem manajemen penelitian mahasiswa yang komprehensif untuk Sekolah Teknologi Informasi di Mapua University, Filipina. Sistem ini dirancang untuk mengotomatisasi proses manajemen skripsi/tesis mulai dari pemilihan topik, persetujuan dosen pembimbing, penjadwalan sidang, hingga pengarsipan dan pengindeksan dokumen penelitian. Pengembangan sistem ini menggunakan pendekatan *Agile Methodology*, dengan melibatkan para pemangku kepentingan seperti mahasiswa, dosen pembimbing, koordinator penelitian, panel penguji, hingga dekan dalam tahap perencanaan dan pengumpulan kebutuhan sistem. Sistem yang dikembangkan terdiri dari enam modul utama, yaitu: *Dashboard*, Penjadwalan Sidang, Revisi dan Persetujuan, Notifikasi *Email*, Sistem Manajemen Dokumen Elektronik (EDMS), dan Pembuatan Laporan. Fitur-fitur tersebut memungkinkan pengguna untuk memantau status dokumen secara *real-time*, mengelola komunikasi dan persetujuan, serta mengakses dokumen akhir secara digital. Evaluasi sistem dilakukan melalui pengujian unit, *black-box testing*, dan *beta testing* yang menunjukkan hasil positif terhadap kestabilan dan fungsionalitas sistem. Sistem ini juga di-*hosting* pada *server* untuk digunakan secara langsung oleh civitas akademika. Hasil penelitian menunjukkan bahwa sistem ini mampu mengurangi beban kerja manual dalam pengelolaan skripsi serta meningkatkan keteraturan dan efisiensi dalam proses akademik. Peneliti juga merekomendasikan pengembangan lebih lanjut, seperti integrasi tanda tangan

elektronik dalam dokumen PDF dan pengembangan aplikasi *mobile* untuk notifikasi jadwal sidang dan pengumuman lainnya.

#### **2.16.5 *Development of a Framework to Implement a Web-Based Thesis Management System in Colleges of Education in Nigeria***

Penelitian yang dilakukan oleh Aminat Obakhume Abdul-Salaam [26] dari *Federal College of Education (Special), Oyo State, Nigeria* bertujuan untuk mengembangkan kerangka kerja (*framework*) dalam penerapan sistem manajemen skripsi berbasis *web* guna mengatasi berbagai permasalahan yang muncul pada proses pengelolaan skripsi secara manual di perguruan tinggi, khususnya pada *colleges of education* di Nigeria. Penelitian ini dilatarbelakangi oleh proses manajemen skripsi yang masih bergantung pada pengumpulan dokumen fisik, pencatatan manual, serta komunikasi tatap muka yang sering menimbulkan keterlambatan, kesalahan pencatatan, dan kesulitan dalam pemantauan progres skripsi mahasiswa. Penelitian ini menggunakan metode *mixed-methods*, dengan mengombinasikan pengumpulan data kuantitatif melalui kuesioner dan data kualitatif melalui wawancara kepada mahasiswa, dosen, staf IT, dan administrator. Hasil analisis digunakan untuk menyusun sebuah *framework* yang mencakup tahapan penilaian kesiapan institusi, kebutuhan teknis, pelatihan dan dukungan pengguna, fitur sistem, serta proses pemantauan dan evaluasi berkelanjutan. Sistem manajemen skripsi berbasis *web* yang dikaji dalam penelitian ini dirancang untuk mendukung proses pengajuan dokumen, pemantauan progres skripsi, komunikasi antara mahasiswa dan dosen pembimbing, serta pengelolaan data skripsi secara terpusat. Hasil penelitian menunjukkan bahwa penerapan sistem manajemen skripsi berbasis *web* berpotensi membantu meningkatkan kualitas pengelolaan skripsi, memperbaiki komunikasi antara mahasiswa dan dosen pembimbing, serta mempermudah pihak administrasi dalam melakukan pemantauan proses skripsi. Namun demikian, penelitian ini juga menyoroti sejumlah tantangan dalam implementasi sistem, seperti keterbatasan infrastruktur teknologi, konektivitas internet, sumber daya manusia, serta resistensi terhadap perubahan. Oleh karena itu, penelitian ini merekomendasikan perlunya perencanaan yang matang, dukungan

pelatihan, serta kesiapan institusi agar sistem manajemen skripsi berbasis *web* dapat diterapkan secara berkelanjutan dan sesuai dengan kebutuhan akademik.

#### **2.16.6 THESISIT: Web-Based University Thesis Management Portal with a Defense Scheduling System**

Penelitian yang dilakukan oleh Ma. Esther B. Chio, Petal May M. Dal, Jocelyn L. Garrido, Arlene Baldelovar, dan JC Vanny Mill Saledaen [27] dari *University of Science and Technology of Southern Philippines* bertujuan untuk mengembangkan portal manajemen skripsi berbasis *web* yang dapat mendukung seluruh siklus penyusunan skripsi mahasiswa di lingkungan universitas. Penelitian ini dilatarbelakangi oleh permasalahan dalam proses penyusunan skripsi yang masih dilakukan secara manual, seperti kesulitan penjadwalan sidang, pengelolaan dokumen skripsi, serta koordinasi antara mahasiswa, dosen pembimbing, dan penguji, yang berpotensi menyebabkan keterlambatan kelulusan mahasiswa. Pengembangan sistem pada penelitian ini menggunakan pendekatan *waterfall*, yang meliputi tahapan pengumpulan dan analisis data, perancangan basis data dan aplikasi berbasis *web*, pengembangan sistem, serta implementasi dan pengujian. Sistem yang dikembangkan menyediakan fitur pengunggahan dan pengelolaan dokumen skripsi, pencatatan hasil sidang, penyimpanan dokumen dalam repositori terpusat, serta penjadwalan sidang skripsi secara otomatis menggunakan algoritma genetika. Sistem dirancang untuk digunakan oleh beberapa peran pengguna, antara lain mahasiswa, dosen pembimbing, penguji, sekretaris, serta pimpinan fakultas yang berperan dalam pemantauan proses skripsi. Hasil pengujian menunjukkan bahwa sistem mampu menjalankan fungsi utama dalam mendukung proses manajemen skripsi dan penjadwalan sidang. Berdasarkan hasil uji kegunaan, sekitar 94% responden menyatakan bahwa sistem mudah digunakan dan sesuai dengan kebutuhan pengguna, sedangkan hasil uji fungsionalitas menunjukkan bahwa 84% responden menilai sistem telah mampu mendukung fitur-fitur yang dibutuhkan dalam siklus penyusunan skripsi. Penelitian ini menyimpulkan bahwa portal manajemen skripsi berbasis *web* dengan penjadwalan sidang otomatis dapat membantu mengurangi permasalahan pada proses manual, khususnya dalam

pengelolaan dokumen dan penjadwalan sidang skripsi, serta memberikan dasar untuk pengembangan sistem yang lebih lanjut di masa mendatang.

### III. METODOLOGI PENELITIAN

#### 3.1 Waktu dan Tempat Penelitian

Waktu dan tempat pelaksanaan penelitian dilakukan pada :

1. Waktu Penelitian : Mei 2025 sampai dengan November 2025
2. Tempat Penelitian : Jurusan Perikanan dan Kelautan Universitas Lampung

Adapun jadwal kegiatan penelitian ini adalah sebagai berikut:

Tabel 3.1 Jadwal Penelitian

No	Kegiatan Penelitian	Mei	Juni	Juli	Agustus	September	Oktober	November
1	Analisis Kebutuhan							
2	<i>Sprint Planning</i>							
3	<i>Sprint</i>							
4	<i>Sprint Review</i>							
5	<i>Retrospective</i>							
6	<i>Functional Testing</i>							
7	Penulisan Laporan							

### 3.2 Alat Penelitian

Adapun alat yang dibutuhkan dalam penelitian ini adalah sebagai berikut:

Tabel 3.2 Alat (*Hardware dan Software*) pengembangan sistem

No	Perangkat	Spesifikasi	Kegunaan
1	Laptop	11 <sup>th</sup> Gen Intel(R) Core(TM) i5- 11400H	Perangkat utama untuk mengembangkan <i>backend</i> aplikasi.
2	Server	2 Vcpu, RAM 4GB, SSD 50GB	Digunakan untuk men- <i>deploy</i> <i>backend</i> .
3	Visual Studio Code	Versi 1.73.1	Perangkat Lunak yang digunakan sebagai <i>text editor</i> .
4	PostgreSQL	Versi 15.14	<i>Database</i> relasional utama untuk menyimpan data.
5	Node.js	Versi 22.14.0	<i>Runtime</i> untuk menjalankan aplikasi <i>backend</i> berbasis <i>JavaScript</i> .
6	Express.js	Versi 4.18.2	<i>Framework backend</i> untuk membangun API menggunakan <i>Node.js</i> .
7	Postman	Versi 11.37.3	Digunakan untuk menguji API.
8	Design Requirement	Draw.io	Perangkat lunak untuk mendesain UML.
9	Nginx	Versi 1.24.0	<i>Server Web</i> yang digunakan sebagai <i>reverse proxy</i> untuk mengelola lalu lintas jaringan.

### 3.3 Metode Penelitian

Penelitian ini menggunakan metode yang mengadopsi prinsip *Agile* dengan kerangka kerja *Scrum*. *Scrum* mengatur cara tim berkolaborasi secara efektif untuk

mencapai tujuan proyek melalui siklus iteratif yang disebut dengan *Sprint*. Dalam proyek ini, tim *Scrum* terdiri dari tiga peran utama, yaitu:

1. *Product owner* (Pemilik Produk), yaitu individu yang bertanggung jawab untuk memastikan bahwa sistem *monitoring* skripsi yang dikembangkan sesuai dengan kebutuhan dan tujuan pengguna akhir. Perannya mencakup menentukan fitur dan fungsionalitas utama, mengelola *Backlog* produk, serta menetapkan prioritas dalam pengembangan sistem.
2. *Scrum Master*, yaitu individu yang perannya adalah memastikan proses *scrum* diikuti dengan baik, memfasilitasi pertemuan seperti *Sprint Planning*, *Daily Scrum*, *Sprint Review*, dan *Sprint Retrospective*, serta membantu tim mengatasi hambatan dan memastikan kolaborasi yang efektif.
3. *Development Team*, yaitu tim pengembang yang bertanggung jawab untuk merancang, membangun, dan menguji produk yang sedang dikembangkan.

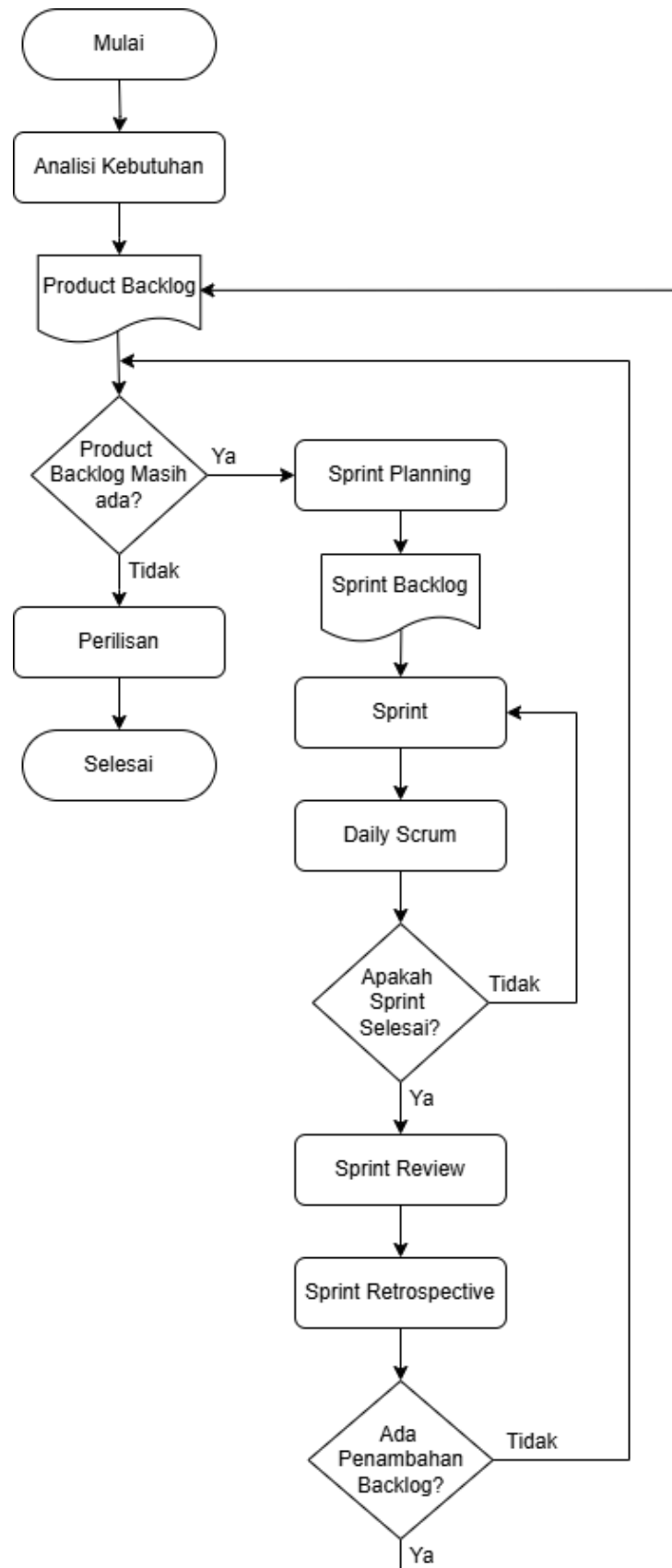
Daftar lengkap anggota tim *Scrum* yang terlibat dalam penelitian ini dapat dilihat pada Tabel 3.3 berikut.

Tabel 3.3 Tim *Scrum*

<b>Peran</b>	<b>Nama</b>
<i>Product Owner</i>	Dr. Munti Sarida, S.Pi., M.Sc.
<i>Scrum Master</i>	Dimas Hidayanto Naim
<i>Development Team</i>	Gibran Alfarabi
	Dimas Hidayanto Naim

### 3.4 Tahapan Penelitian

Adapun tahapan penelitian yang digunakan adalah sebagai berikut:



Gambar 3.1 Tahapan Penelitian

Gambar diatas menunjukkan bahwa penelitian dimulai dengan pengumpulan data dan analisis kebutuhan untuk membentuk *Product Backlog*, yaitu daftar semua pekerjaan yang harus diselesaikan. Jika *Product Backlog* masih ada item yang perlu dikerjakan, tim melakukan *Sprint Planning* untuk memilih item dan membuat *Sprint Backlog*. Selama *Sprint*, tim mengerjakan tugas dan mengadakan *Daily Scrum* untuk memantau progres. Setelah *Sprint* selesai, dilakukan *Sprint Review* untuk menilai hasil kerja, dan *Sprint Retrospective* untuk mengevaluasi proses kerja. Jika ada penambahan item dari hasil evaluasi atau review, item baru dimasukkan ke *Product Backlog* dan siklus *Sprint* diulang. Jika semua item *Product Backlog* sudah selesai, sistem akan dirilis, menandai selesainya penelitian.

### **3.4.1 Analisis Kebutuhan**

Dalam penelitian ini, sebelum memulai pengembangan sistem, dilakukan terlebih dahulu analisis kebutuhan untuk mengidentifikasi berbagai aspek yang diperlukan dalam proses tersebut. Analisis ini dilaksanakan melalui diskusi dengan *Product owner* serta tim pengembang lain guna memperoleh pemahaman yang jelas mengenai fitur dan fungsionalitas yang diharapkan. Hasil dari diskusi tersebut menjadi dasar dalam penyusunan *product Backlog*, yang memuat daftar fitur dan fungsi yang akan dikembangkan selama proses pembuatan sistem *monitoring skripsi*.

#### **3.4.1.1 Kebutuhan Fungsional, dan Kebutuhan Non-Fungsional**

Kebutuhan fungsional menggambarkan fungsionalitas apa saja yang akan dimiliki oleh sistem. Kebutuhan fungsional mencakup fitur-fitur yang akan diterapkan untuk memenuhi kebutuhan pengguna. Berikut adalah tabel kebutuhan fungsionalitas Sistem *Monitoring Skripsi*.

Tabel 3.4 Kebutuhan Fungsional

ID	Penjelasan
KF-01	Sistem harus memungkinkan mahasiswa untuk membuat akun baru (registrasi mandiri)
KF-02	Sistem harus memungkinkan semua pengguna untuk <i>login</i> menggunakan akun yang sudah terdaftar
KF-03	Sistem harus memungkinkan semua pengguna untuk <i>logout</i> dari sistem
KF-04	Mahasiswa dapat melakukan pengajuan usul judul penelitian
KF-05	Mahasiswa dapat mengajukan permohonan seminar usul penelitian, seminar hasil penelitian, dan ujian skripsi
KF-06	Mahasiswa dapat melihat jadwal seminar yang tersedia serta mengisi atau memilih jadwal seminar melalui sistem
KF-07	Mahasiswa dapat mengunggah file laporan bimbingan untuk ditinjau dan diberi <i>feedback</i> oleh dosen pembimbing melalui sistem
KF-08	Mahasiswa dapat mengakses riwayat bimbingan pribadinya melalui sistem
KF-09	Koordinator Prodi dapat melakukan verifikasi terhadap pengajuan usulan judul penelitian
KF-10	Admin dapat melakukan verifikasi terhadap pengajuan seminar usul penelitian, pengajuan seminar hasil penelitian, dan pengajuan ujian skripsi
KF-11	Dosen Pembimbing dapat melihat daftar mahasiswa yang mereka bimbing
KF-12	Dosen Pembimbing dapat memberikan komentar atau catatan terhadap laporan bimbingan mahasiswa

Tabel 3.5 Kebutuhan Fungsional (Lanjutan)

ID	Penjelasan
KF-13	Dosen Pembimbing dapat mengakses riwayat bimbingan mahasiswa yang dibimbingnya melalui sistem
KF-14	Ketua Jurusan dapat memantau progres skripsi setiap mahasiswa melalui sistem

### 3.4.1.2 Kebutuhan *Non-Fungsional*

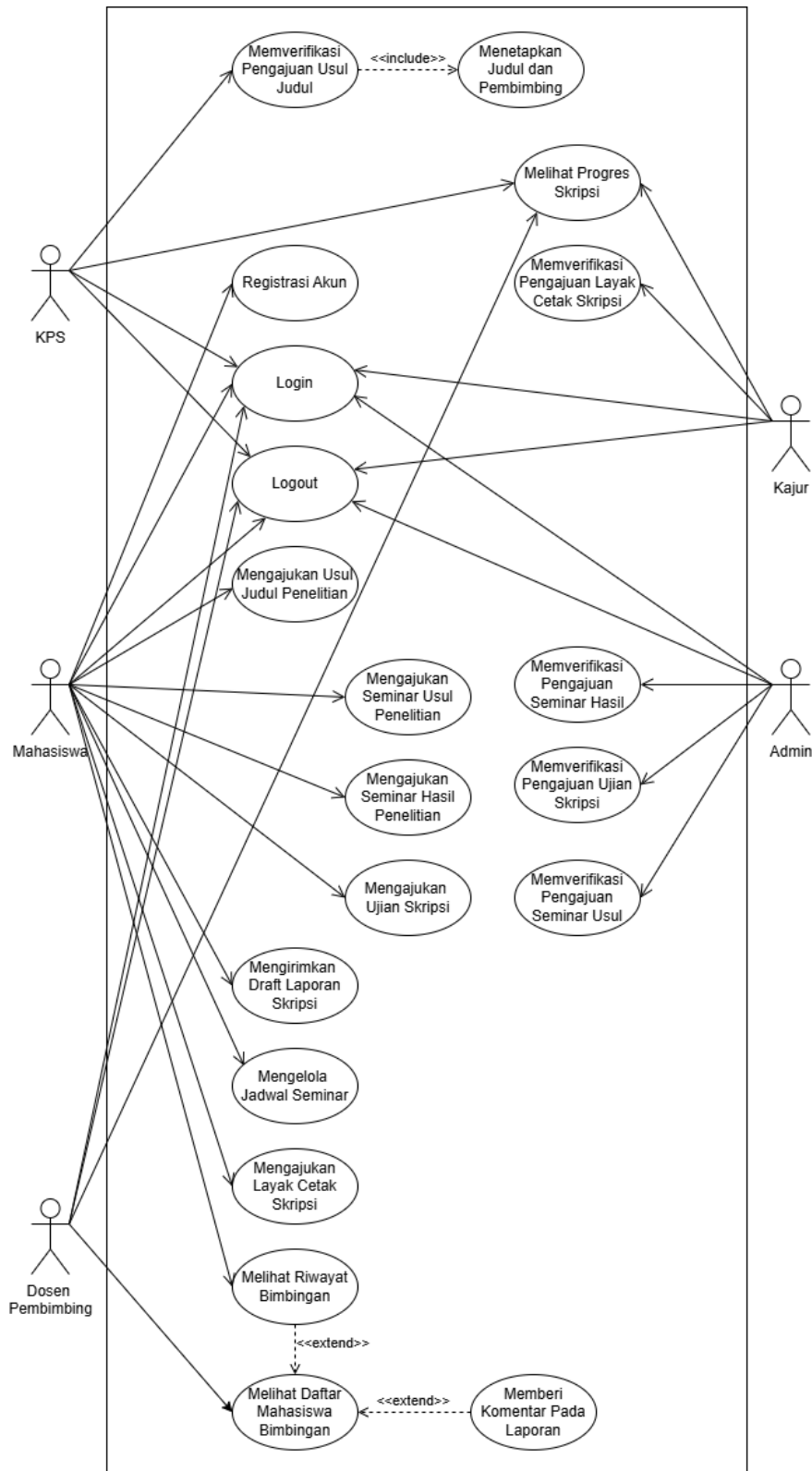
Kebutuhan *non-fungsional* merupakan kebutuhan yang berkaitan dengan kualitas dan karakteristik sistem, yang tidak secara langsung berhubungan dengan fungsi atau fitur, melainkan lebih menekankan pada aspek kinerja, keamanan, serta keandalan sistem. Berikut ini adalah daftar kebutuhan *non-fungsional*.

Tabel 3.5 Kebutuhan *Non-Fungsional*

ID	Parameter	Penjelasan
KnF-01	<i>Availability</i>	Sistem harus dapat diakses 24 jam
KnF-02	<i>Scalability</i>	Sistem harus mendukung <i>caching</i> untuk mengurangi beban pada <i>database</i>
KnF-03	<i>Security</i>	<i>Backend</i> harus menerapkan autentikasi ( <i>login</i> ) dan otorisasi (pengaturan akses)
KnF-04	<i>Performance</i>	<i>Backend</i> harus memproses permintaan API dengan waktu respon maksimal 10 detik untuk 95% <i>Request</i>
KnF-05	<i>Compatibility</i>	<i>Backend</i> harus menggunakan format data JSON

### 3.4.1.3 Use Case Diagram

Adapun *use case diagram* dari sistem *monitoring* skripsi adalah sebagai berikut.



Gambar 3.2 Use Case Diagram

Berikut merupakan tabel definisi aktor pada *use case*.

Tabel 3.6 Definisi Aktor *Use Case*

<b>Aktor</b>	<b>Deskripsi</b>
Mahasiswa	Aktor ini dapat melakukan registrasi akun, <i>login</i> , <i>logout</i> , mengajukan usulan judul penelitian, mengajukan seminar usul, mengajukan seminar hasil, mengajukan ujian skripsi, melakukan bimbingan skripsi, mengisi jadwal seminar, dan mengakses riwayat bimbingan.
Dosen Pembimbing	Aktor ini dapat <i>login</i> , <i>logout</i> , melihat daftar mahasiswa bimbingan, mengakses riwayat bimbingan mahasiswa, memberikan komentar pada laporan, serta melihat progres skripsi dari mahasiswa bimbingannya.
Koordinator Program Studi (KPS)	Aktor ini dapat <i>login</i> , <i>logout</i> , memverifikasi usulan judul, menetapkan judul dan dosen pembimbing, serta melihat progres skripsi dari program studinya.
Admin	Aktor ini dapat <i>login</i> , <i>logout</i> , memverifikasi pengajuan seminar usul, memverifikasi seminar hasil, serta memverifikasi ujian skripsi.
Ketua Jurusan (Kajur)	Aktor ini dapat <i>login</i> , <i>logout</i> , dan melihat progres skripsi dari seluruh mahasiswa di jurusannya.

Berikut adalah tabel definisi *use case*.

Tabel 3.7 Definisi *Use Case*

<b>Use Case</b>	<b>Deskripsi</b>
Registrasi Akun	Mahasiswa dapat mendaftar ke sistem untuk membuat akun baru
<i>Login</i>	Pengguna dapat masuk ke dalam sistem dengan akun yang sudah terdaftar

Tabel 3.8 Definisi *Use Case* (Lanjutan)

<i>Use Case</i>	<b>Deskripsi</b>
<i>Logout</i>	Pengguna dapat keluar dari sistem
Mengajukan Usul Judul Penelitian	Mahasiswa dapat mengajukan judul penelitian skripsi melalui sistem
Mengajukan Seminar Usul Penelitian	Mahasiswa dapat mengajukan permohonan untuk melaksanakan seminar usul penelitian
Mengajukan Seminar Hasil Penelitian	Mahasiswa dapat mengajukan permohonan untuk melaksanakan seminar hasil penelitian
Mengajukan Ujian Skripsi	Mahasiswa dapat mengajukan permohonan untuk melaksanakan ujian skripsi
Mengirimkan Draft Laporan Skripsi	Mahasiswa dapat mengunggah file draf skripsi melalui sistem
Mengelola Jadwal Seminar	Mahasiswa dapat melihat serta mengisi jadwal seminar
Mengajukan Layak Cetak Skripsi	Mahasiswa dapat mengajukan permohonan kepada ketua jurusan untuk memperoleh persetujuan kelayakan cetak skripsi
Memverifikasi Pengajuan Usul Judul	Koordinator Program Studi dapat melakukan verifikasi terhadap pengajuan judul
Menetapkan Judul dan Pembimbing	Koordinator Program Studi dapat menetapkan judul penelitian serta pembimbing untuk mahasiswa
Memverifikasi Pengajuan Seminar Usul	Admin dapat melakukan verifikasi terhadap pengajuan seminar usul yang diajukan oleh mahasiswa
Memverifikasi Pengajuan Seminar Hasil	Admin dapat melakukan verifikasi terhadap pengajuan seminar hasil yang diajukan oleh mahasiswa

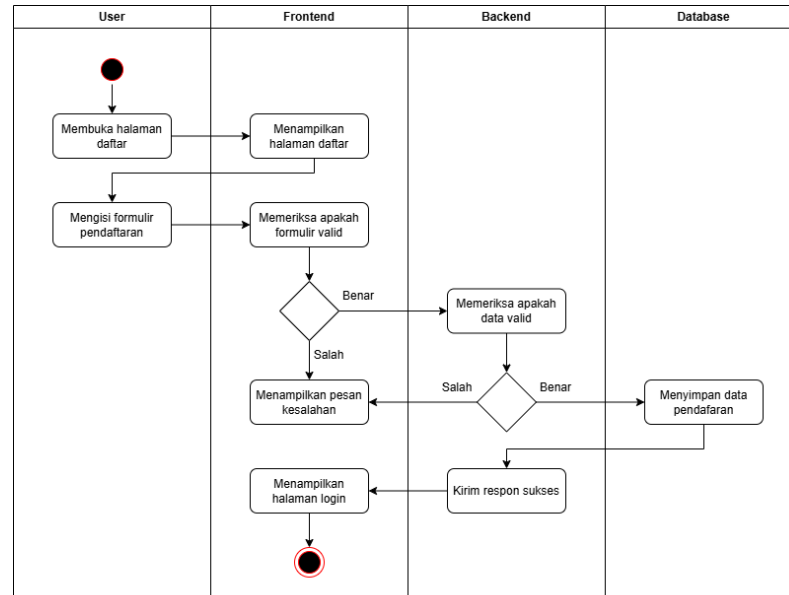
Tabel 3.8 Definisi *Use Case* (Lanjutan)

<i>Use Case</i>	Deskripsi
Memverifikasi Pengajuan Ujian Skripsi	Admin dapat melakukan verifikasi terhadap pengajuan ujian skripsi yang diajukan oleh mahasiswa
Melihat Daftar Mahasiswa Bimbingan	Dosen Pembimbing dapat melihat daftar mahasiswa yang mereka bimbing
Melihat Riwayat Bimbingan	Mahasiswa dan dosen pembimbing dapat mengakses riwayat aktivitas bimbingan skripsi
Memberi Komentar Pada Laporan	Dosen pembimbing dapat memberikan komentar atau <i>feedback</i> pada laporan bimbingan yang diunggah mahasiswa
Melihat Progres Skripsi	Ketua Jurusan, Koordinator Program Studi, dan Dosen Pembimbing dapat memantau perkembangan pengerjaan skripsi mahasiswa
Memverifikasi Pengajuan Layak Cetak Skripsi	Kajur dapat melakukan verifikasi terhadap pengajuan layak cetak skripsi yang diajukan oleh mahasiswa

#### 3.4.1.4 Activity Diagram

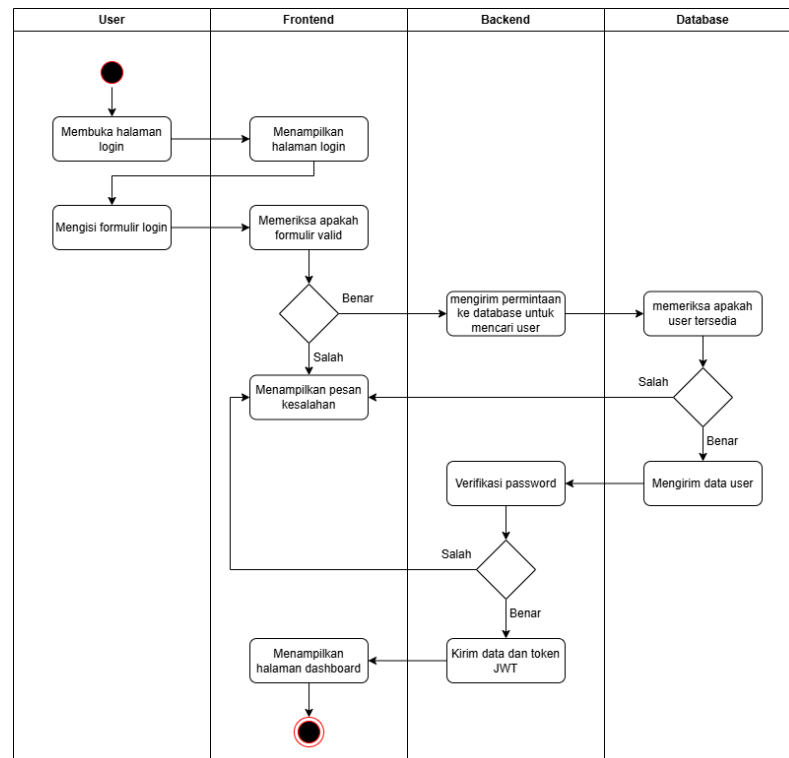
*Activity Diagram* merupakan diagram yang berfungsi untuk memodelkan urutan tindakan dalam suatu alur proses, dengan tujuan menangkap aktivitas serta hasil yang dihasilkan dari proses tersebut. Adapun *Activity Diagram* dari sistem *monitoring* skripsi adalah sebagai berikut.

a. *Activity Diagram Registrasi Akun*



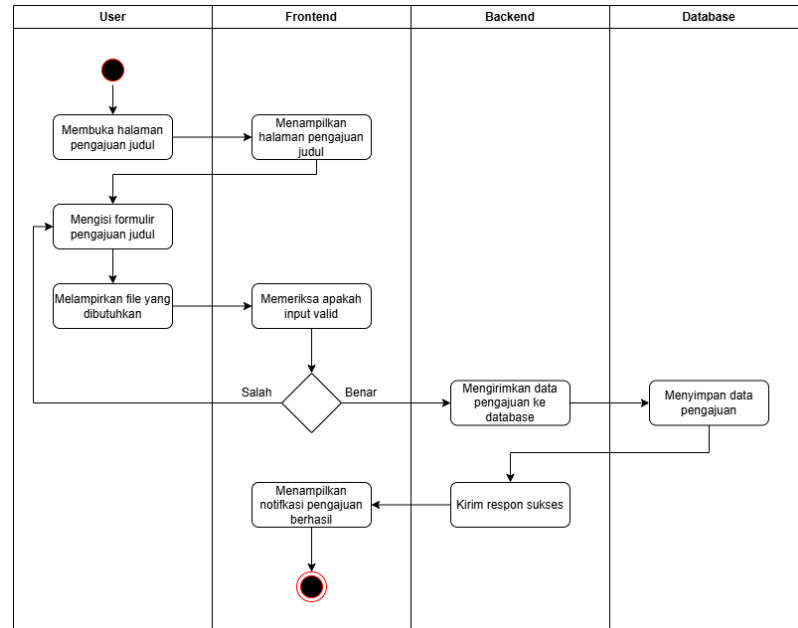
Gambar 3.3 *Activity Diagram Registrasi Akun*

b. *Activity Diagram Login*



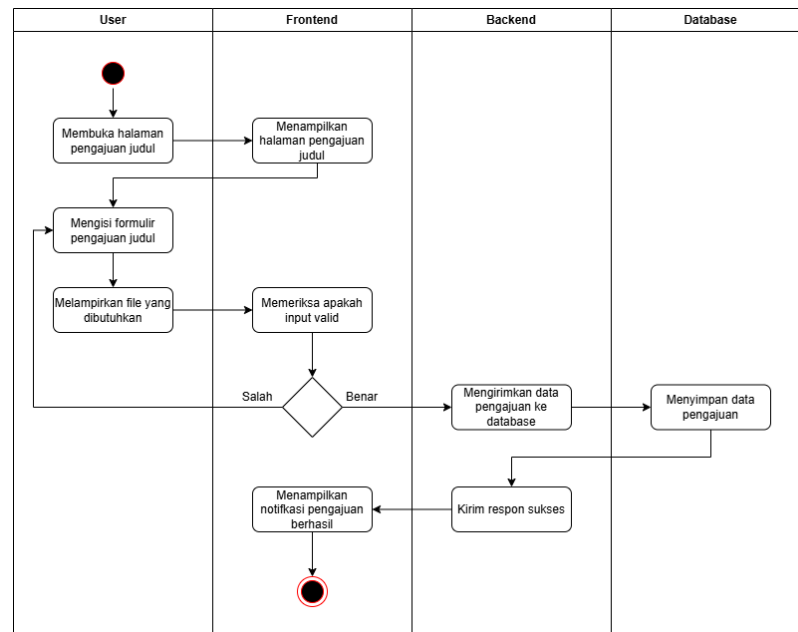
Gambar 3.4 *Activity Diagram Login*

c. *Activity Diagram Mengajukan Usul Judul Penelitian*



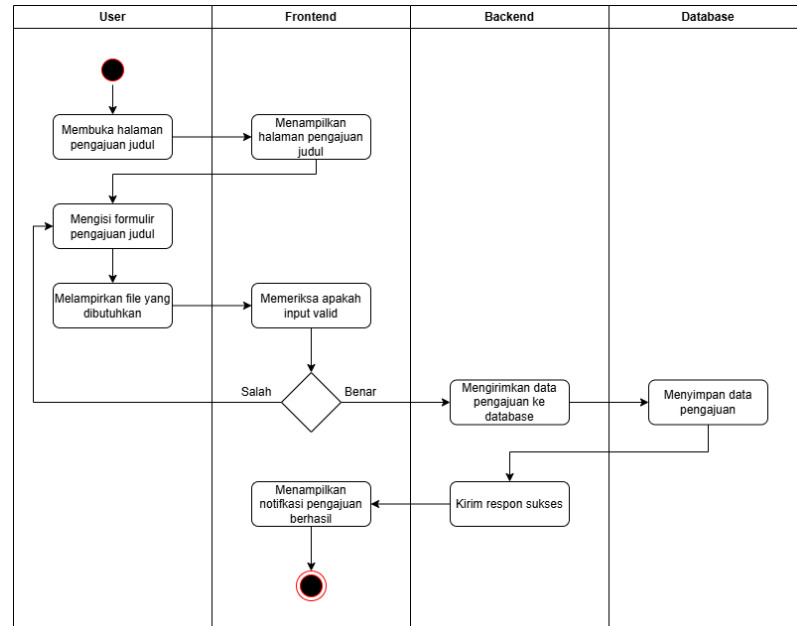
Gambar 3.5 *Activity Diagram Mengajukan Usul Judul Penelitian*

d. *Activity Diagram Mengajukan Seminar Usul Penelitian*



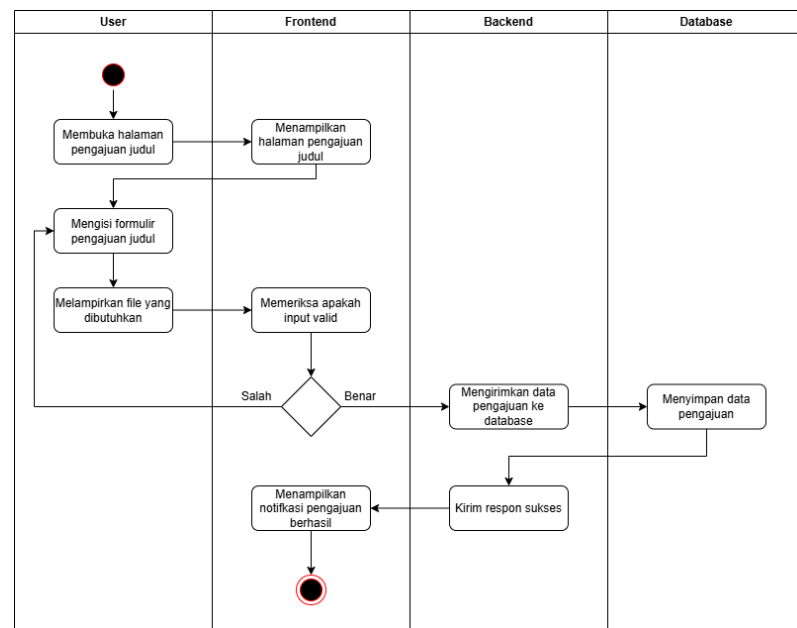
Gambar 3.6 *Activity Diagram Mengajukan Seminar Usul Penelitian*

e. *Activity Diagram Mengajukan Seminar Hasil Penelitian*



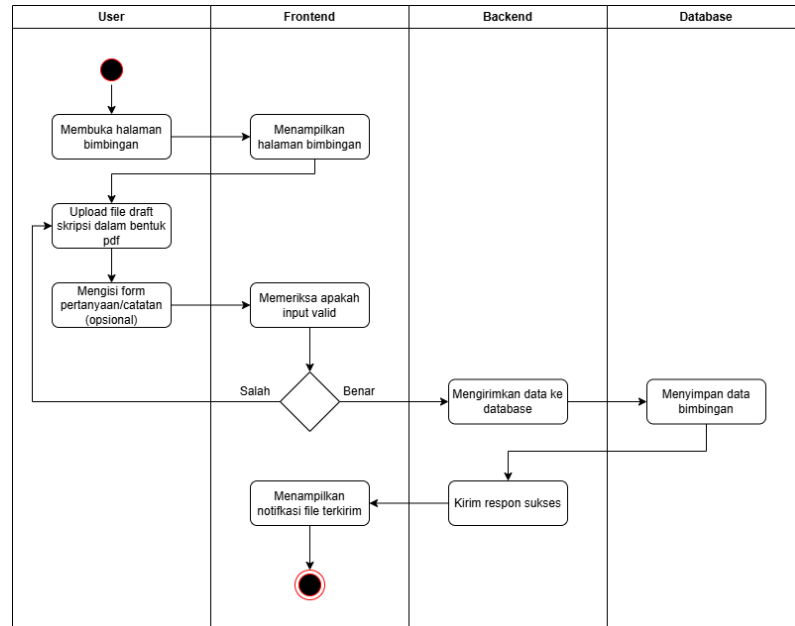
Gambar 3.7 *Activity Diagram Mengajukan Seminar Hasil Penelitian*

f. *Activity Diagram Mengajukan Ujian Skripsi*



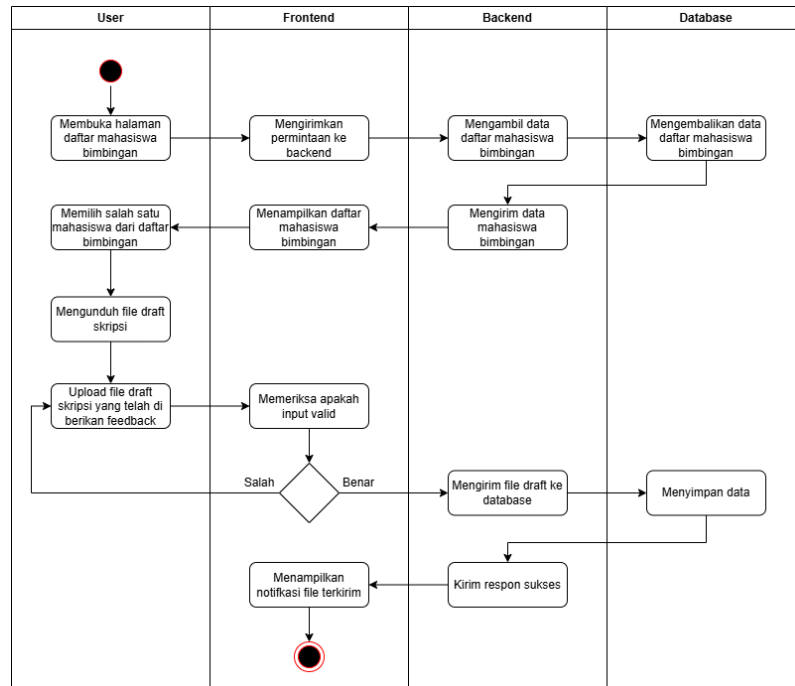
Gambar 3.8 *Activity Diagram Mengajukan Ujian Skripsi*

g. *Activity Diagram* Mengirimkan Draft Laporan Skripsi



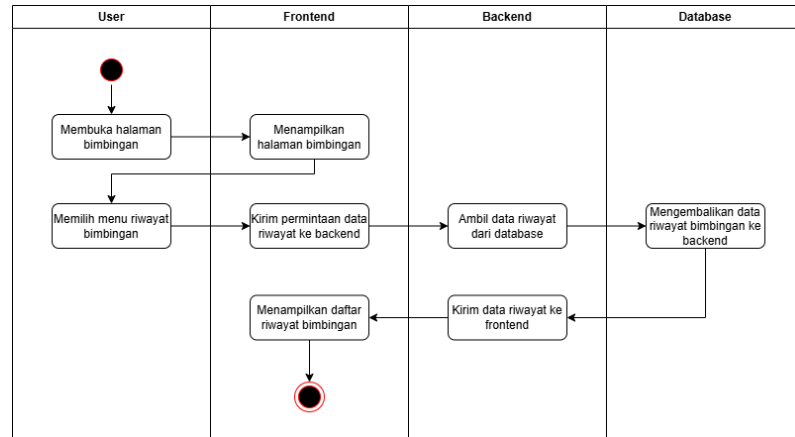
Gambar 3.9 *Activity Diagram* Mengirimkan Draft Laporan Skripsi

h. *Activity Diagram* Melihat Daftar Mahasiswa Bimbingan



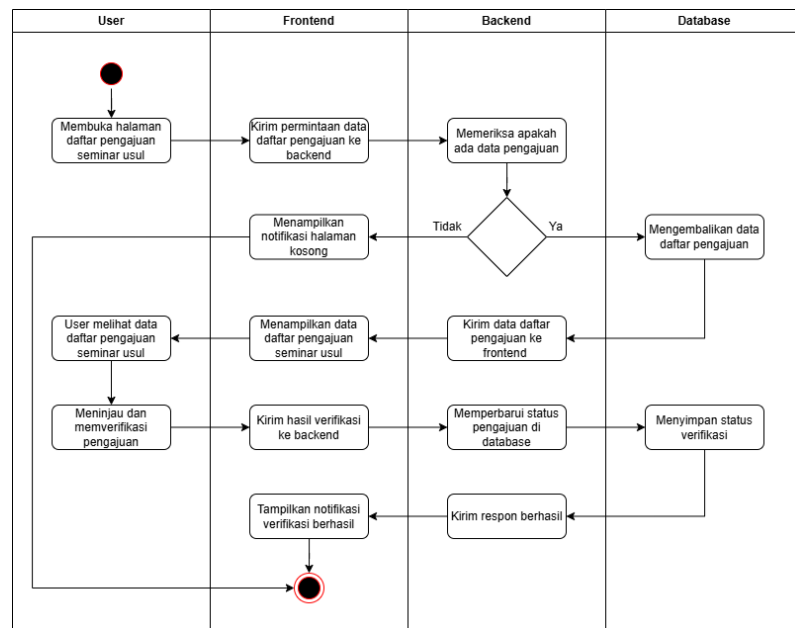
Gambar 3.10 *Activity Diagram* Melihat Daftar Mahasiswa Bimbingan

i. *Activity Diagram* Melihat Riwayat Bimbingan



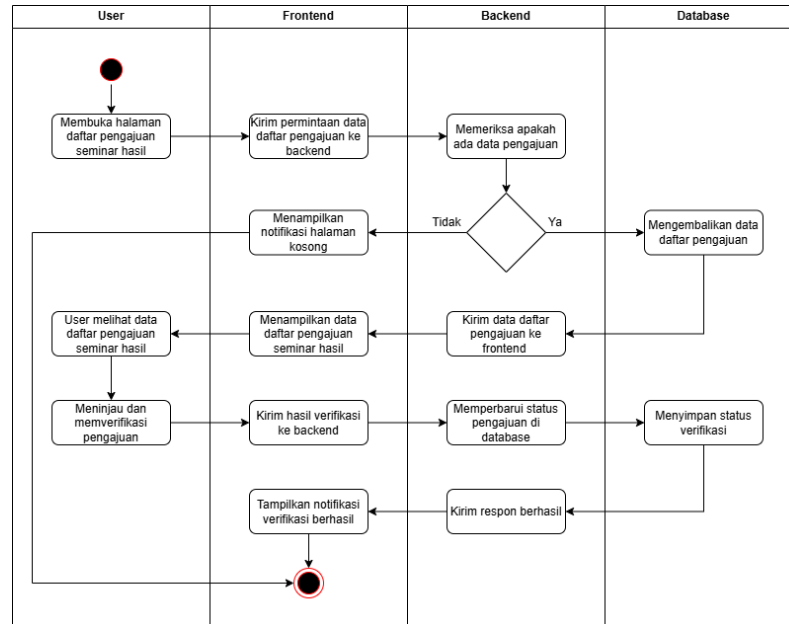
Gambar 3.11 *Activity Diagram* Melihat Riwayat Bimbingan

j. *Activity Diagram* Memverifikasi Pengajuan Seminar Usul



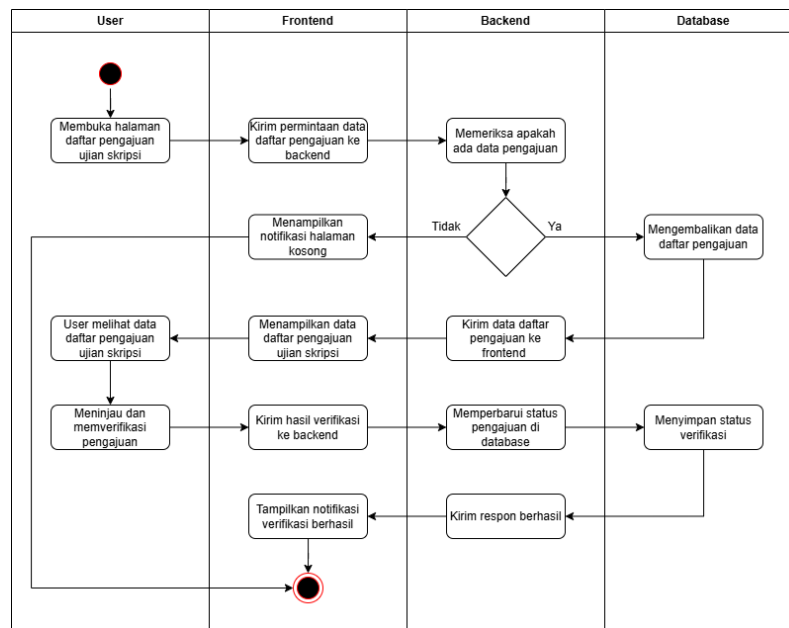
Gambar 3.12 *Activity Diagram* Memverifikasi Pengajuan Seminar Usul

k. *Activity Diagram Memverifikasi Pengajuan Seminar Hasil*



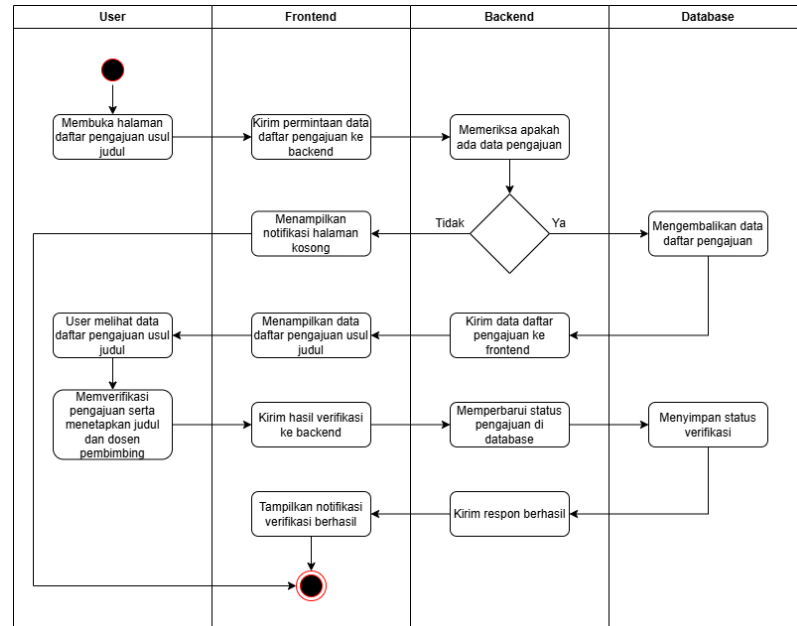
Gambar 3.13 *Activity Diagram Memverifikasi Pengajuan Seminar Hasil*

l. *Activity Diagram Memverifikasi Pengajuan Ujian Skripsi*



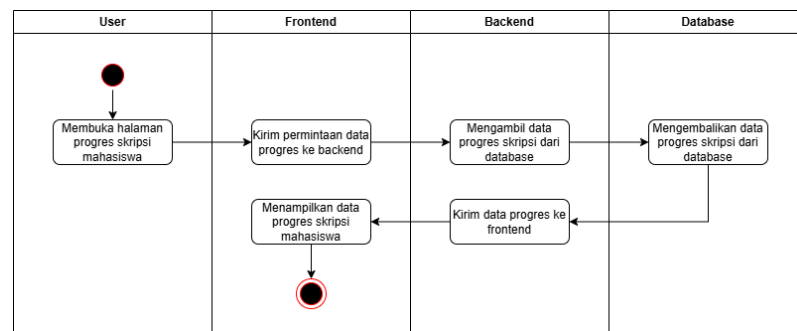
Gambar 3.14 *Activity Diagram Memverifikasi Pengajuan Ujian Skripsi*

m. *Activity Diagram* Memverifikasi Pengajuan Usul Judul



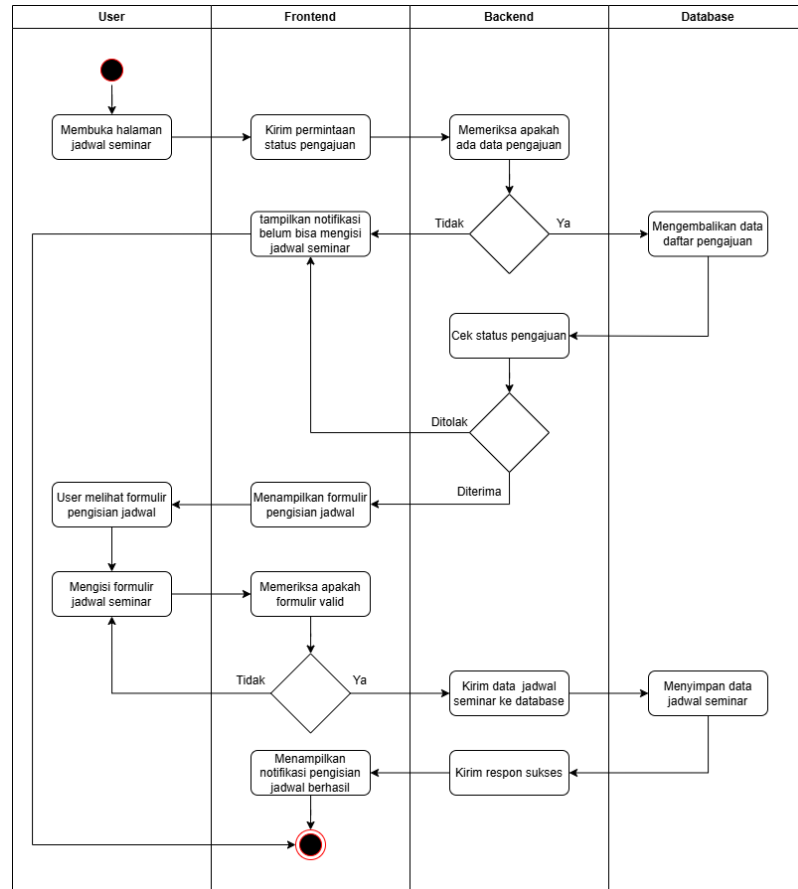
Gambar 3.15 *Activity Diagram* Verifikasi Pengajuan Usul Judul

n. *Activity Diagram* Melihat Progres Skripsi



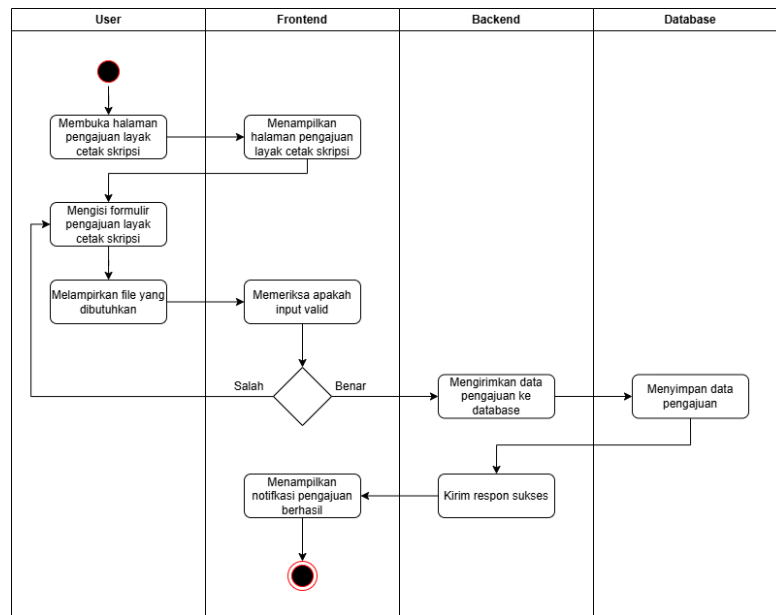
Gambar 3.16 *Activity Diagram* Melihat Progres Skripsi

o. Activity Diagram Mengelola Jadwal Seminar



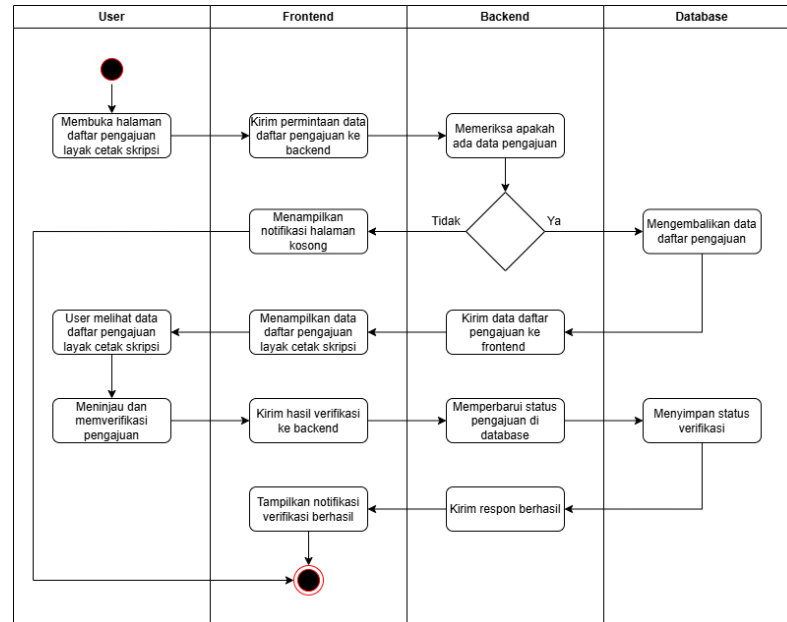
Gambar 3.17 Activity Diagram Mengelola Jadwal Seminar

p. Activity Diagram Mengajukan Layak Cetak Skripsi



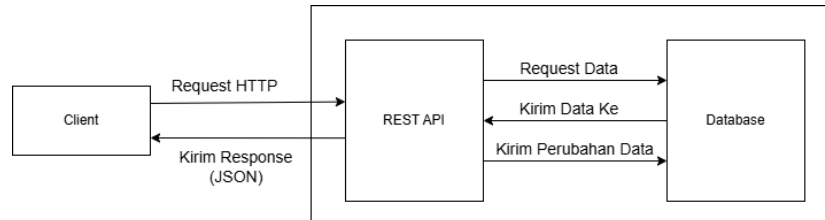
Gambar 3.18 Activity Diagram Mengajukan Layak Cetak Skripsi

q. *Activity Diagram* Memverifikasi Pengajuan Layak Cetak Skripsi



Gambar 3.19 *Activity Diagram* Memverifikasi Pengajuan Layak Cetak Skripsi

### 3.4.1.5 Gambaran Umum Sistem



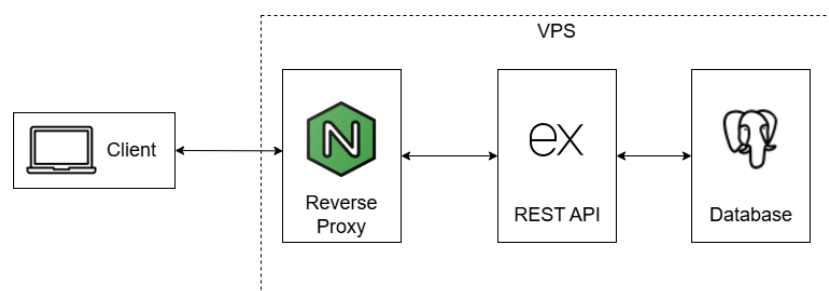
Gambar 3.20 Gambaran Umum Sistem

Berdasarkan gambar 3.20, dapat dilihat bahwa untuk memperoleh data dari *database*, *client* harus mengirimkan *Request* ke *backend server* menggunakan metode *HTTP Request*. *Request* ini kemudian diterima oleh *REST API* yang bertugas memproses permintaan sesuai dengan jenis *Request* yang dikirimkan oleh *client*. *REST API* kemudian melakukan pengambilan data dari *database PostgreSQL* untuk memenuhi permintaan tersebut. Data yang berhasil diambil kemudian diproses oleh subprogram atau fungsi tertentu sesuai dengan kebutuhan *Request*. Apabila permintaan *client* hanya berupa permintaan untuk melihat data yang ada, maka *REST API* langsung mengirimkan *response* kembali kepada *client* tanpa melakukan perubahan terhadap data dalam *database*. Namun, jika layanan

yang diminta *client* menghasilkan perubahan terhadap data yang ada, maka REST API akan mengirimkan data hasil perubahan tersebut kembali ke *database* untuk memperbarui data yang sebelumnya tersimpan. Setelah semua proses selesai, REST API mengembalikan *response* kepada *client* dalam format JSON.

### 3.4.1.6 Arsitektur Sistem *Backend*

Adapun arsitektur sistem *backend* dalam sistem *monitoring* skripsi adalah sebagai berikut:



Gambar 3.21 Arsitektur Sistem *Backend*

Berdasarkan dari gambar 3.21 Arsitektur Sistem, terdapat beberapa komponen yang digunakan sebagai berikut:

Tabel 3.8 Deskripsi Arsitektur Sistem

No	Komponen	Deskripsi
1	<i>Client</i>	Komponen <i>frontend</i> yang berjalan di <i>web browser</i> dan digunakan oleh mahasiswa, dosen pembimbing, admin, dan ketua jurusan untuk mengakses sistem <i>monitoring</i> skripsi. Semua permintaan dikirim ke <i>server</i> melalui <i>client</i> ini.
2	<i>Reverse Proxy</i> ( <i>Nginx</i> )	Bertindak sebagai gerbang utama yang menerima permintaan dari <i>client</i> . <i>Nginx</i> bertugas melakukan <i>routing</i> ke REST API, serta menangani SSL (HTTPS).

Tabel 3.8 Deskripsi Arsitektur Sistem (lanjutan)

No	Komponen	Deskripsi
3	RESTful API ( <i>Express.js</i> )	<i>Server</i> aplikasi utama yang menangani seluruh logika bisnis sistem. Dibangun menggunakan <i>framework Express.js</i> .
4	<i>Database</i> ( <i>PostgreSQL</i> )	Sistem basis data yang menyimpan seluruh data penting, seperti data pengguna (mahasiswa, dosen, admin), informasi judul skripsi, file bimbingan, jadwal seminar, serta riwayat bimbingan.

### 3.4.2 Tahapan Pengembangan Sistem

Pada tahap ini, pengembangan sistem dilakukan berdasarkan hasil analisis kebutuhan yang telah disusun sebelumnya. Metode *agile scrum* digunakan dalam proses pengembangan, sehingga memungkinkan pendekatan yang iteratif dan adaptif. Setiap fitur yang tercantum dalam *product backlog* akan diimplementasikan melalui siklus *sprint*, yang meliputi beberapa tahap, yaitu *product backlog*, *sprint planning*, *sprint backlog*, *sprint*, *daily scrum*, *sprint review*, dan *sprint retrospective*.

#### 3.4.2.1 Product Backlog

*Product backlog* adalah kumpulan fitur dan fungsi yang direncanakan untuk dikembangkan selama proses pembangunan sistem. Daftar ini bersifat fleksibel dan dapat berubah sesuai dengan penyesuaian kebutuhan dari *Product owner* atau tim pengembang. *Product owner* memiliki tanggung jawab dalam mengelola *product backlog* untuk memastikan bahwa fitur yang dikembangkan sejalan dengan tujuan sistem.

### **3.4.2.2 *Sprint Planning***

*Sprint planning* adalah tahap perencanaan *sprint*. Pada tahap ini, dipilih daftar *product backlog* yang akan diselesaikan dalam satu *sprint*, ditetapkan tujuan dari masing-masing *backlog*, serta ditentukan durasi *sprint* yang biasanya berlangsung selama dua minggu. Seluruh tim *scrum* berkumpul untuk berdiskusi dan menyepakati tugas-tugas yang akan dikerjakan selama *sprint*, sesuai waktu dan tempat yang telah disepakati.

### **3.4.2.3 *Sprint Backlog***

*Sprint backlog* adalah kumpulan item dari *product backlog* yang dipilih untuk diselesaikan dalam satu *sprint*. Setiap anggota tim pengembang bertanggung jawab atas satu atau lebih *sprint backlog* yang telah ditetapkan saat *sprint planning*.

### **3.4.2.4 *Sprint***

*Sprint* adalah tahap pelaksanaan di mana tim pengembang mulai mengimplementasikan fitur-fitur sesuai dengan *sprint backlog* yang telah ditetapkan. Tim pengembang wajib menyelesaikan tugas dalam waktu *sprint* yang telah disepakati, tanpa melakukan perubahan besar selama proses berlangsung.

### **3.4.2.5 *Daily Scrum***

*Daily scrum* merupakan pertemuan harian yang difasilitasi oleh *scrum master* dan diikuti oleh seluruh tim pengembang, baik secara *online* maupun langsung. Dalam pertemuan ini, setiap anggota tim menyampaikan perkembangan tugasnya, mengungkapkan hambatan yang ditemui, serta berkoordinasi untuk memastikan tercapainya *sprint backlog* yang telah direncanakan.

### **3.4.2.6 *Sprint Review***

*Sprint review* adalah pertemuan yang melibatkan seluruh tim *scrum* untuk meninjau hasil kerja selama satu *sprint*. Pertemuan ini dilaksanakan setelah *sprint* berakhir

dan dipimpin oleh *product owner*. Pada tahap ini, *product owner* menilai *backlog* yang telah diselesaikan, memberikan umpan balik, serta menentukan apakah diperlukan perubahan atau penambahan.

#### **3.4.2.7 *Sprint Retrospective***

*Sprint retrospective* adalah tahap evaluasi terhadap proses kerja yang dilakukan selama *sprint* sebelumnya. Dalam sesi ini, tim pengembang bersama *product owner* membahas kendala yang ditemui, peluang perbaikan, serta strategi untuk meningkatkan efisiensi pada *sprint* selanjutnya. Hasil diskusi ini akan digunakan sebagai dasar untuk iterasi berikutnya dan perencanaan *sprint* selanjutnya.

## V. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Berdasarkan hasil penelitian dan pembahasan mengenai pengembangan *backend* pada sistem *monitoring* skripsi di Jurusan Perikanan dan Kelautan Universitas Lampung, dapat disimpulkan beberapa hal berikut:

1. Arsitektur *backend* sistem *monitoring* skripsi telah dirancang menggunakan pendekatan *client-server* dengan komponen *reverse proxy* (*Nginx*), *RESTful* API berbasis *Express.js*, serta basis data *PostgreSQL* sebagai penyimpanan data terpusat. Perancangan arsitektur ini mendukung proses pengelolaan data progres skripsi secara terpusat serta memungkinkan akses data secara langsung melalui sistem. Berdasarkan hasil pengujian, waktu rekapitulasi data progres skripsi yang sebelumnya memerlukan  $\pm 1-2$  hari kerja dapat diperoleh melalui sistem dalam waktu sekitar  $\pm 5$  detik pada kondisi beban normal, sehingga menunjukkan peningkatan efisiensi pada proses rekapitulasi data.
2. *Backend* sistem *monitoring* skripsi telah diimplementasikan melalui pengembangan *endpoint* API berdasarkan kebutuhan fungsional yang telah ditetapkan dalam *product backlog*. Implementasi dilakukan secara iteratif menggunakan metodologi *Scrum* melalui 9 *sprint*, sehingga setiap fitur dapat dikelola melalui sistem secara terpusat sesuai dengan peran dan hak akses masing-masing pengguna. Sistem yang dikembangkan mencakup 20 fitur utama yang mendukung proses pengelolaan skripsi secara menyeluruh.
3. Arsitektur *RESTful* API telah diterapkan pada *backend* dengan penggunaan metode HTTP seperti *GET*, *POST*, *PUT*, dan *DELETE*, serta pertukaran data dalam format JSON. Setiap fitur direpresentasikan dalam bentuk *endpoint* berbasis *resource* yang dapat diakses oleh *frontend* melalui mekanisme *request-response*.

## 5.2 Saran

1. Meningkatkan aspek keamanan sistem dengan menambahkan mekanisme seperti pembatasan akses (*rate limiting*), autentikasi dua faktor (2FA), enkripsi pada penyimpanan berkas, serta pencatatan aktivitas pengguna. Penerapan fitur-fitur ini diharapkan dapat memperkuat perlindungan data dan meminimalkan potensi penyalahgunaan sistem. Teknologi yang digunakan dapat disesuaikan dengan kebutuhan dan *platform* yang digunakan oleh pengembang pada tahap pengembangan berikutnya.
2. Mengimplementasikan integrasi dengan SIAKAD untuk memungkinkan sinkronisasi data akademik secara otomatis, sehingga proses administratif dapat dilakukan dengan lebih efisien. Mekanisme integrasi ini dapat disesuaikan dengan fasilitas yang tersedia pada sistem SIAKAD, baik melalui layanan pertukaran data yang disediakan maupun metode integrasi lain yang direkomendasikan oleh pihak pengelola SIAKAD.

## DAFTAR PUSTAKA

- [1] F. Mulyani dan N. Haliza, “Analisis Perkembangan Ilmu Pengetahuan dan Teknologi (IPTEK) dalam Pendidikan,” *Jurnal Pendidikan dan Konseling (JPDK)*, vol. 3, no. 1, hlm. 101–109, Feb. 2021.
- [2] P. Pant, A. Singh, R. Kumar, dan S. Kumar, “Authentication and Authorization in Modern Web Apps for Data Security Using Node.js and Role of Dark Web,” *Procedia Computer Science*, vol. 215, hlm. 781–790, 2022.
- [3] M. F. A. Nasrullah, R. Yulius, dan L. Pratama, “Aplikasi Issue Tracking System Berbasis Web (Studi Kasus: PT Mitra Kuadran Teknologi),” *Teknik: Jurnal Ilmu Teknik dan Informatika*, vol. 2, no. 3, hlm. 47–50, Sep. 2022.
- [4] M. Hilmyansyah, M. Malabay, H. Simorangkir, dan Y. Yulhendri, “Implementasi Metode Scrum pada Pembangunan Sistem Informasi Monitoring Progress Proyek Berbasis Web (Studi Kasus: PT Quatra Engineering Mandiri),” *Ikraith-Informatika*, vol. 6, no. 3, hlm. 121–129, Nov. 2022.
- [5] E. Lase dan H. Lase, “Implementasi Sistem Monitoring dalam Meningkatkan Pelayanan Publik pada Dinas SPMDP2A Kabupaten Nias,” *Jurnal Manajemen Pendidikan dan Ilmu Sosial*, vol. 5, no. 4, hlm. 1765–1773, 2024.
- [6] R. D. P. Shinta dan A. R. Syarif, “Students’ Barriers in Writing Thesis Proposal: Psychological Factors,” *Journal of Educational Psychology Studies*, vol. 9, no. 1, hlm. 45–53, 2025.
- [7] H. Koç, A. M. Erdoğan, Y. Barjakly, dan S. Peker, “UML Diagrams in Software Engineering Research: A Systematic Literature Review,” dalam *Proceedings of the 7th International Management Information Systems Conference*, Mar. 2021, hlm. 1–13.

- [8] R. D. Setiyawan, D. Hermawan, O. Herdiyanto, dan R. Rahmawati, “Integrasi Gamifikasi dalam Perancangan Sistem: Tinjauan Metodologi, Kerangka Kerja, dan Pertimbangan Kritis,” *RIGGS: Journal of Artificial Intelligence and Digital Business*, vol. 4, no. 2, hlm. 6018–6027, Jul. 2025.
- [9] W. Yanto, H. Alhaq, R. S. Sari, dan M. Juanda, “Implementasi UML dalam Desain Sistem Informasi Program Studi Sistem Informasi di Universitas Merangin,” *Jurnal Pendidikan dan Informatika*, vol. 4, no. 2, hlm. 88–96, 2023.
- [10] S. Al-Fedaghi, “Conceptual Data Modeling: Entity–Relationship Models as Thinging Machines,” *International Journal of Computer Science and Network Security*, vol. 21, no. 9, hlm. 247–260, Sep. 2021.
- [11] E. Anthony dan T. Tony, “Perancangan Aplikasi Manajemen Data Publikasi dan Penelitian,” *Jurnal Teknologi Informasi*, vol. 12, no. 2, hlm. 134–142, 2024.
- [12] F. Widoyutomo, H. Ajie, dan Widodo, “Pengembangan Web Service Modul Mahasiswa pada Sistem Informasi Akademik Universitas Negeri Jakarta,” *PINTER: Jurnal Pendidikan Teknik Informatika dan Komputer*, vol. 5, no. 1, hlm. 68–75, Jun. 2021.
- [13] S. B. Uzayr, Ed., *Mastering Visual Studio Code: A Beginner’s Guide*, 1st ed. Boca Raton, FL, USA: CRC Press, 2023.
- [14] K. Hartomo, C. Arthur, dan I. W. Rabbani, “Analisis Kerangka Kerja Scrum pada Sistem Informasi SIGAP sebagai Model Transformasi Digital Kepolisian,” *Jurnal Algoritma*, vol. 22, no. 2, hlm. 211–220, Nov. 2025.
- [15] K. Budiman, *Rekayasa Perangkat Lunak untuk Mahasiswa Ilmu Komputer*. Bandung, Indonesia: Pustaka Rumah C1nta, 2022.
- [16] A. A. F. Amarta dan I. G. Anugrah, “Implementasi Agile Scrum Menggunakan Trello sebagai Manajemen Proyek di PT Andromedia,” *Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)*, vol. 4, no. 6, hlm. 528–534, Des. 2021.
- [17] N. M. K. Koneru, *Containerization Best Practices: Using Docker and Kubernetes for Enterprise Applications*. New York, NY, USA: Apress, 2025.

- [18] I. A. K. P. Paramitha, D. M. Wiharta, dan I. M. A. Suyadnya, “Perancangan dan Implementasi RESTful API pada Sistem Informasi Manajemen Dosen Universitas Udayana,” *Jurnal SPEKTRUM*, vol. 9, no. 3, hlm. 15–22, Sep. 2022.
- [19] D. R. Prasetyo, F. Fatoni, M. Nasir, dan I. Effendy, “Implementasi Redis untuk Mempercepat Pengambilan Data (Studi Kasus: Sistem Dasawisma PKK Sumsel),” *Explore: Jurnal Sistem Informasi dan Telematika*, vol. 15, no. 2, hlm. 214–223, Des. 2024.
- [20] P. H. Saputro, “Penerapan GTMetrix dan K6 dalam Pengujian Performa dan Tingkat Stres pada Website POS (Point of Sale),” *Jurnal Teknologi Informasi*, vol. 2, no. 1, hlm. 33–41, 2023.
- [21] S. Sibuea, Y. B. Widodo, dan M. N. Khaliq, “Penggunaan Software Nginx sebagai Load Balancing Web Server Clustering,” *Jurnal Teknologi Informasi dan Komputer*, vol. 10, no. 1, hlm. 326–341, Mar. 2024.
- [22] Salmi dan Darmatasia, “Sistem Manajemen dan Monitoring Bimbingan Tugas Akhir Berbasis Web,” *AGENTS: Journal of Artificial Intelligence and Data Science*, vol. 3, no. 1, hlm. 1–8, Feb. 2023.
- [23] H. Santoso, W. Widodo, dan M. A. Rizky, “Sistem Monitoring Skripsi Berbasis Progressive Web Application dengan Push Notification,” *Jurnal Informatika, Komputer, Bisnis dan Manajemen*, vol. 20, no. 1, hlm. 32–41, Nov. 2023.
- [24] B. Rahman dan S. A. Ningsi, “Sistem Informasi Monitoring Skripsi Mahasiswa Berbasis Website,” *Simtek: Jurnal Sistem Informasi dan Teknik Komputer*, vol. 7, no. 2, hlm. 166–170, Okt. 2022.
- [25] J. J. E. Macrohon, A. S. Bulaong, J. C. A. Bonsol, J. A. C. Gayeta, dan J.-H. Jeng, “Design and Development of a System for Managing Student Research: A Case Study for the School of Information Technology, Mapua University,” dalam *Proceedings of the IEEE International Conference on Educational Technology (ICET)*, Beijing, China, Jun. 2021, hlm. 107–111.
- [26] A. O. Abdul-Salaam, “Development of a Framework to Implement a Web-Based Thesis Management System in Colleges of Education in Nigeria,”

*American Journal of Educational Technology*, vol. 3, no. 4, hlm. 60–71, Nov. 2024.

- [27] M. E. B. Chio, “THESISIT: Web-Based University Thesis Management Portal with a Defense Scheduling System,” *International Journal of Information Systems and Software Engineering*, vol. 6, no. 2, hlm. 98–107, 2022.
- [28] Y. F. Kurniawan dan A. A. Riadi, “Website-Based Management and Financial Information System Using Prototyping Method at GITJ Puncel Church,” *Journal of Information Systems and Technology*, vol. 10, no. 2, hlm. 145–154, 2025.