

**RANCANG BANGUN APLIKASI *MONITORING* SKRIPSI
DI PROGRAM STUDI PENDIDIKAN
TEKNOLOGI INFORMASI**

(Skripsi)

**Oleh
DEWI AYU LESTARI**



**FAKULTAS KEGURUAN DAN ILMU PENDIDIKAN
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2024**

ABSTRAK

RANCANG BANGUN APLIKASI *MONITORING* SKRIPSI DI PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI

Oleh

DEWI AYU LESTARI

Teknologi informasi saat ini telah mengalami perkembangan yang semakin menampakkan eksistensinya di berbagai bidang kehidupan, salah satunya di bidang pendidikan. Penerapan Teknologi Informasi dalam konteks pendidikan memberikan manfaat utama berupa kemudahan, baik dalam akses maupun proses pembelajaran. Penelitian ini bertujuan untuk merancang dan membangun aplikasi *monitoring* skripsi di Program Studi Pendidikan Teknologi Informasi, Fakultas Keguruan dan Ilmu Pendidikan, Universitas Lampung. Aplikasi ini menyediakan fitur-fitur seperti registrasi *user*, pengajuan judul skripsi, pengajuan dosen pembimbing dan pembahas, serta bimbingan dengan dosen pembimbing yang telah ditetapkan. Selain itu, mahasiswa juga dapat mengakses revisi yang diberikan oleh dosen pembimbing, dan memberikan revisi yang telah diperbaiki kepada dosen pembimbing, serta fitur-fitur lainnya. Pengembangan penelitian ini menggunakan metode *Waterfall* yang terdiri dari tahapan *Analysis*, *Design*, *Implementation*, dan *Testing*. Penelitian ini mengimplementasikan gaya arsitektur *Representational State Transfer* (REST) dalam mengembangkan API sebagai *backend* pada aplikasi *monitoring* skripsi. *Application Programming Interface* (API) yang dikembangkan menggunakan *JSON Web Token* (JWT) sebagai metode otentikasi *user*. Hasil penelitian ini menunjukkan bahwa pengembangan API pada aplikasi *monitoring* skripsi berhasil dilakukan, dan penerapan arsitektur REST memudahkan pengembangan struktur API. Pengujian fungsionalitas fitur aplikasi dilakukan menggunakan *postman*, *user testing*, dan *browser*. Selain itu, pengujian keamanan juga dilakukan menggunakan perangkat lunak seperti OWASP ZAP untuk memastikan sistem tidak rentan terhadap serangan atau pelanggaran keamanan. Hasil pengujian dengan *postman*, *user testing*, dan *browser* menunjukkan bahwa aplikasi berjalan dengan baik sesuai harapan, dan hasil pengujian keamanan menegaskan bahwa aplikasi ini memenuhi standar keamanan tingkat 2 (medium).

Kata Kunci: JWT, Metode *Waterfall*, *Monitoring* Skripsi, *Postman*, REST API

ABSTRACT

DESIGN AND DEVELOPMENT OF THESIS MONITORING APPLICATIONS IN THE INFORMATION TECHNOLOGY EDUCATION STUDY PROGRAM

By

DEWI AYU LESTARI

Information technology has now experienced developments that increasingly reveal its existence in various fields of life, one of which is in the field of education. The application of Information Technology in the context of education provides the main benefits in the form of convenience, both in access and the learning process. This research aims to design and build a thesis monitoring application in the Information Technology Education Study Program, Faculty of Teacher Training and Education, University of Lampung. This application provides features such as user registration, submission of thesis titles, submission of supervisors and discussants, and guidance with predetermined supervisors. In addition, students can also access revisions given by the supervisor, and provide revisions that have been corrected to the supervisor, as well as other features. The development of this research uses the Waterfall method which consists of the stages of Analysis, Design, Implementation, and Testing. This research implements the Representational State Transfer (REST) architectural style in developing an Application Programming Interface (API) as a backend in the thesis monitoring application. The Application Programming Interface (API) developed uses JSON Web Token (JWT) as a user authentication method. The results of this study indicate that the development of APIs in thesis monitoring applications is successful, and the application of REST architecture facilitates the development of API structures. Functionality testing of application features is done using postman, user testing, and browsers. In addition, security testing is also carried out using software such as OWASP ZAP to ensure the system is not vulnerable to attacks or security breaches. The results of testing with postman, user testing, and browsers show that the application runs well as expected, and the results of security testing confirm that this application meets level 2 (medium) security standards.

Keywords : JWT, Waterfall Method, Thesis Monitoring, Postman, REST API

**RANCANG BANGUN APLIKASI *MONITORING* SKRIPSI
DI PROGRAM STUDI PENDIDIKAN
TEKNOLOGI INFORMASI**

Oleh:

Dewi Ayu Lestari

Skripsi

**sebagai salah satu syarat untuk mencapai gelar
SARJANA PENDIDIKAN**

pada

**Program Studi Pendidikan Teknologi Informasi
Jurusan Pendidikan Matematika dan Ilmu Pengetahuan Alam**



**FAKULTAS KEGURUAN DAN ILMU PENDIDIKAN
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2024**

Judul Skripsi : **RANCANG BANGUN APLIKASI
MONITORING SKRIPSI DI PROGRAM
STUDI PENDIDIKAN TEKNOLOGI
INFORMASI**

Nama Mahasiswa : **Dewi Ayu Testari**

Nomor Pokok Mahasiswa : **1813025008**

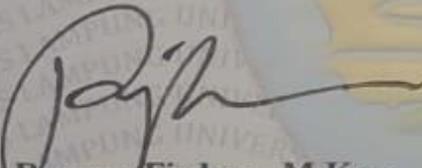
Program Studi : **Pendidikan Teknologi Informasi**

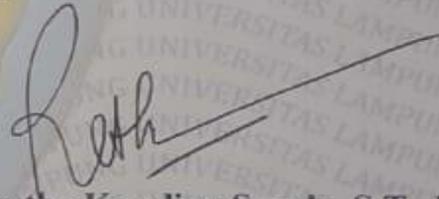
Jurusan : **Pendidikan MIPA**

Fakultas : **Keguruan dan Ilmu Pendidikan**

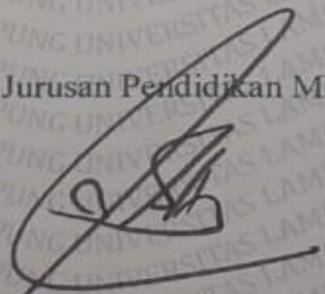
MENYETUJUI

1. **Komisi Pembimbing**


Dr. Rangga Firdaus, M.Kom.
NIP 197410102008011015


Margaretha Karolina Sagala, S.T., M.Pd.
NIP 198803092022032008

2. **Ketua Jurusan Pendidikan MIPA**


Prof. Dr. Undang Rosidin, M.Pd.
NIP 196003011985031003

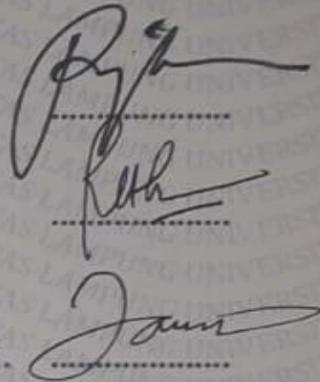
MENGESAHKAN

I. Tim Penguji

Ketua : **Dr. Rangga Firdaus, M.Kom.**

Sekretaris : **Margaretha Karolina Sagala, S.T., M.Pd.**

Penguji
Bukan Pembimbing : **Ir. Meizano Ardhi Muhammad, S.T., M.T., IPM.**



Dekan Fakultas Keguruan dan Ilmu Pendidikan

Prof. Dr. Sunyono, M.Si.
NIP. 196512301991111001

Tanggal Lulus Ujian Skripsi: 21 Maret 2024

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Dewi Ayu Lestari
NPM : 1813025008
Fakultas/Jurusan : KIP/Pendidikan MIPA
Program Studi : Pendidikan Teknologi Informasi
Alamat : Dusun Purwodadi, Kecamatan Kotagajah, Kabupaten Lampung Tengah, Provinsi Lampung

menyatakan bahwa skripsi saya yang berjudul "Rancang Bangun Aplikasi *Monitoring* Skripsi di Program Studi Pendidikan Teknologi Informasi" merupakan karya sendiri dan bukan karya orang lain. Semua tulisan yang tertuang dalam skripsi ini sudah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau telah dibuat oleh orang lain sebelumnya, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandarlampung, 2 April 2024

Dewi Ayu Lestari

RIWAYAT HIDUP

Penulis lahir di Desa Purwodadi, Lampung Tengah pada 27 November 2000. Penulis adalah anak kedua dari dua bersaudara, dengan ayah bernama Sukiman dan Ibu bernama Muntinah. Pendidikan formal pertama penulis tempuh di Taman Kanak-Kanak (TK) Pertiwi Purworejo pada tahun 2006, lalu melanjutkan ke Sekolah Dasar (SD) di SD Negeri 6 Kotagajah dan menyelesaikannya pada tahun 2012. Setelah itu, penulis melanjutkan studi ke Madrasah Tsanawiyah (MTS) di MTS Ma'arif 02 Kotagajah dan menyelesaikannya pada tahun 2015. Pendidikan Menengah Atas (SMA) penulis tempuh di SMA Negeri 1 Kotagajah dan berhasil menyelesaikannya pada tahun 2018.

Tahun 2018, penulis diterima di Universitas Lampung, Fakultas Keguruan dan Ilmu Pendidikan, Jurusan Pendidikan Matematika dan Ilmu Pengetahuan Alam, Program Studi Pendidikan Teknologi Informasi melalui seleksi SBMPTN. Selama menjadi mahasiswa, penulis aktif di berbagai organisasi, seperti Himpunan Mahasiswa Pendidikan Eksakta (HIMASAKTA) sebagai Anggota Divisi Media Center Eksakta Muda dari tahun 2018 hingga 2019, Badan Eksekutif Mahasiswa Fakultas FKIP Unila (BEM FKIP UNILA) sebagai Brigda Dinas Sosial Masyarakat dari tahun 2018 hingga 2019, Forum Mahasiswa Pendidikan Teknologi Informasi (FORMATIF) sebagai Sekretaris Divisi Kerohanian dari tahun 2020 hingga 2021, dan Bina Rohani Islam Mahasiswa (BIROHMAH) sebagai Anggota Departemen Musabaqoh Tilawatil Qur'an dan Seni islam dari tahun 2020 hingga 2021.

Tahun 2020, penulis mengikuti Kompetisi Film Pendek Sains HIMASAKTA Tingkat Nasional Universitas Lampung yang diselenggarakan oleh Himpunan Mahasiswa Pendidikan Eksakta (HIMASAKTA) dan berhasil meraih juara 3

bersama tim. Bulan Januari hingga Maret 2021, penulis juga mengikuti Kuliah Kerja Nyata (KKN) di Desa Tanggul Rejo, Kecamatan Kota Gajah, Kabupaten Lampung Tengah selama 40 hari dan melaksanakan kegiatan Pengenalan Lapangan Persekolahan (PLP) di SMA Negeri 1 Kotagajah selama 50 hari. Tahun yang sama, yaitu tahun 2021, penulis juga mengikuti Program Mahasiswa Wirausaha (PMW) dan berhasil mendapatkan bantuan modal. Bulan Agustus 2021, penulis melaksanakan Praktik Industri (PI) di Badan Pengelolaan Keuangan dan Aset Daerah (BPKAD) Kabupaten Pesawaran, tepatnya di bagian Sekretariat selama 21 hari. Tahun 2022, penulis bergabung dalam *bootcamp Fullstack Javascript* di Hactiv8 selama 4 bulan, beasiswa IDCamp yang diselenggarakan oleh Indosat Ooredoo Hutchison untuk mencetak *developer/programming* muda Indonesia, penulis juga beruntung mendapatkan beasiswa *DevOps Engineer* dan *Backend Developer*. Selain itu, pada tahun 2023, penulis juga menjadi bagian dari beasiswa program *bootcamp for university* yang diselenggarakan oleh Dicoding, beasiswa *Fullstack Javascript Expert* serta *Fullstack MEVN* yang bersertifikat internasional yang diselenggarakan oleh Codepolitan dan Alibaba *Cloud* yang memberikan peluang penulis untuk mengasah keahlian dan mendapatkan pemahaman mendalam tentang teknologi yang relevan di tingkat global.

MOTTO HIDUP

“Hidup adalah tentang menciptakan diri sendiri, jadi jangan biarkan orang lain menulis cerita hidupmu.”

“Jangan pernah meragukan dirimu sendiri. Kamu mampu lebih dari yang kamu pikirkan dan mampu melakukan lebih dari yang kamu bayangkan.”
(Roy T. Bennett)

“Perjalanan hidup adalah samudra yang luas, dan kita adalah kapal yang berlayar dengan impian sebagai kompas dan tekad sebagai angin.”

“Dan jika kamu menghitung nikmat-nikmat Allah, niscaya kamu tidak akan dapat menentukannya.”
(Surah An-Nahl, 16:18)

“Tidak ada *lift* menuju sukses. Kamu harus naik tangga.”
(Zig Ziglar)

“Ketika satu pintu kebahagiaan tertutup, pintu lain terbuka.
Tetapi seringkali kita terpaku pada pintu yang tertutup sehingga kita tidak melihat pintu lain yang telah terbuka.”
(Helen Keller)

“Seperti badai yang mengguncang bumi, kehadirannya memaksa dunia untuk memperhatikan kehadirannya yang luar biasa.”

PERSEMBAHAN

Assalamualaikum Warahmatullahi Wabarakatuh

Segala puji bagi Allah SWT, yang telah melimpahkan karunia, rahmat, dan hidayah-Nya. Semoga sholawat senantiasa tercurah kepada Nabi Muhammad SAW. Penulis dengan rendah hati menyampaikan penghargaan dan persembahan ini sebagai wujud rasa pengabdian dan kasih sayang yang mendalam kepada:

1. Allah SWT, yang telah melimpahkan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan penulisan skripsi ini dengan baik.
2. Orang tua tersayang penulis, telah sepenuh hati membesarkan, mendidik, memberikan dukungan, masukan, motivasi, serta doa yang tak terhingga. Semoga Allah SWT senantiasa memberikan berkah, kesehatan, umur yang panjang, dan kesempatan bagi penulis untuk membalas kebahagiaan mereka.
3. Teman-teman angkatan 2018 Pendidikan Teknologi Informasi, yang telah memberikan semangat, kerjasama, dan dukungan selama perjalanan penulisan skripsi ini.
4. Keluarga besar Forum Mahasiswa Pendidikan Teknologi Informasi FKIP Universitas Lampung, yang telah memberikan lingkungan yang inspiratif, kesempatan berkolaborasi, dan pembelajaran yang berharga.
5. Almamater tercinta, Universitas Lampung, yang telah memberikan pendidikan dan sarana untuk berkembangnya penulis dalam bidang Pendidikan Teknologi Informasi.

Semoga persembahan sederhana ini dapat menjadi kontribusi yang bermanfaat bagi pembaca dan seluruh insan yang terlibat. Semoga Allah SWT menerima segala ikhtiar dan memperkenankan segala doa yang terkandung dalam persembahan ini. Aamiin.

Wassalamualaikum Warahmatullahi Wabarakatuh.

SANWACANA

Assalamualaikum Warahmatullahi Wabarakatuh

Alhamdulillah, segala puji bagi Allah SWT, dengan berkah dan rahmat-Nya, skripsi ini dapat diselesaikan.

Judul skripsi ini adalah “Rancang Bangun Aplikasi *Monitoring* Skripsi di Program Studi Pendidikan Teknologi Informasi”. Skripsi ini merupakan salah satu bagian dari hasil penulisan skripsi yang telah selesai. Skripsi ini merupakan syarat untuk memperoleh gelar Sarjana Pendidikan di Universitas Lampung.

Pelaksanaan serta penyusunan skripsi ini, penulis ingin menyampaikan rasa terima kasih yang tulus kepada semua pihak yang telah membantu penyelesaian skripsi ini. Oleh karena itu, dengan kerendahan hati, penulis ingin menyampaikan terima kasih kepada:

1. Allah SWT, yang telah memberikan rahmat dan karunia-Nya selama penulis mengerjakan skripsi ini hingga dapat tersusun dengan baik.
2. Orang tua penulis, yang selalu memberikan dukungan, masukan, motivasi, dan doa yang tak terhingga.
3. Prof. Dr. Sunyono, M.Si. selaku Dekan FKIP Universitas Lampung.
4. Prof. Dr. Undang Rosidin, M.Pd. selaku Ketua Jurusan Pendidikan Matematika dan Ilmu Pengetahuan Alam Fakultas Keguruan dan Ilmu Pendidikan Universitas Lampung.
5. Ibu Dr. Pramudiyanti, S.Si., M.Si. selaku Ketua Program Studi Pendidikan Teknologi Informasi Fakultas Keguruan dan Ilmu Pendidikan Universitas Lampung.
6. Bapak Dr. Rangga Firdaus, M.Kom. selaku Dosen Pembimbing I, atas kesediaan, kesabaran, dan keikhlasannya dalam memberikan dukungan,

bimbingan, nasihat, saran, arahan, dan masukan dalam proses penyelesaian skripsi ini. Terima kasih juga karena telah memberikan semangat sebagai Pembimbing Akademik penulis.

7. Ibu Margaretha Karolina Sagala, S.T., M.Pd. selaku Dosen Pembimbing II, yang berkenan memberikan saran, arahan, bimbingan, dan motivasi dalam proses penyelesaian skripsi ini.
8. Bapak Ir. Meizano Ardhi Muhammad, S.T., M.T., IPM. selaku dosen pembahas yang memberikan saran dan perbaikan dalam proses penyelesaian skripsi ini.
9. Bapak dan Ibu Dosen Program Studi Pendidikan Teknologi Informasi FKIP Universitas Lampung, yang telah memberikan ilmu yang bermanfaat dan pengalaman hidup kepada penulis selama perkuliahan.
10. Bapak dan Ibu Staff Administrasi FKIP Unila.
11. Rekan-rekan Program Studi Pendidikan Teknologi Informasi FKIP Unila, yang merupakan *support system* penulis dan memberikan dukungan, bantuan, masukan, serta kebersamaannya selama pengerjaan dan penyelesaian skripsi.
12. Almamater tercinta, Universitas Lampung, yang telah memberikan kesempatan kepada penulis untuk menempuh pendidikan dalam jenjang S-1 dengan baik.
13. Semua pihak yang tidak dapat disebutkan satu per satu, tetapi telah memberikan kontribusi dan dukungan dalam penyelesaian skripsi ini.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan dan masih terdapat kesalahan karena keterbatasan kemampuan, pengalaman, dan pengetahuan penulis. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun dari berbagai pihak. Semoga skripsi ini bermanfaat bagi pembaca secara umum dan penulis khususnya.

Bandarlampung, 2 April 2024

Penulis,

Dewi Ayu Lestari

DAFTAR ISI

	Halaman
ABSTRAK	ii
ABSTRACT	iii
RIWAYAT HIDUP	viii
MOTTO HIDUP	x
PERSEMBAHAN	xi
SANWACANA	xii
DAFTAR ISI	xiv
DAFTAR TABEL	xvi
DAFTAR GAMBAR	xviii
DAFTAR LAMPIRAN	xxi
I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian.....	3
1.4 Manfaat Penelitian.....	3
1.5 Ruang Lingkup.....	4
II. TINJAUAN PUSTAKA	7
2.1 Aplikasi <i>Monitoring</i>	7
2.2 Skripsi.....	6
2.3 Pemrograman <i>Frontend</i>	7
2.3.1 <i>Javascript</i>	7
2.3.2 <i>HyperText Markup Language 5 (HTML5)</i>	9
2.3.3 <i>Cascading Style Sheets (CSS)</i>	9
2.3.4 <i>Bootstrap</i>	10
2.4 Pemrograman <i>Backend</i>	10
2.4.1 <i>Representational State Transfer (REST)</i>	11
2.4.2 <i>Application Programming Interface (API)</i>	11
2.4.3 <i>Node JS</i>	12
2.4.4 <i>Express JS</i>	13
2.4.5 <i>PostgreSQL</i>	14
2.4.6 <i>Sequelize</i>	14
2.4.7 <i>JavaScript Object Notation (JSON)</i>	15
2.4.8 <i>Group Inclusive Tour (GIT)</i>	15

2.4.9 <i>Node Package Manager (NPM)</i>	16
2.4.10 <i>JSON Web Token (JWT)</i>	16
2.4.11 <i>Bcrypt</i>	17
2.5 <i>Unified Modeling Language (UML)</i>	17
2.6 <i>Postman</i>	18
2.7 Metode Pengembangan <i>Waterfall</i>	19
2.8 Penelitian Terkait	20
III. METODE PENELITIAN	22
3.1 Metode Penelitian.....	22
3.2 Tempat dan Waktu Penelitian.....	22
3.3 Alat dan Bahan.....	23
3.3.1 Perangkat Keras (<i>Hardware</i>).....	23
3.3.2 Perangkat Lunak (<i>Software</i>).....	23
3.4 Metode Pengumpulan Data.....	24
3.4.1 Observasi.....	24
3.4.2 Wawancara (<i>Interview</i>)	25
3.5 Metode Pengembangan Sistem	25
3.5.1 <i>Analysis</i>	25
3.5.2 <i>Design</i>	28
3.5.3 <i>Implementation</i>	78
3.5.4 <i>Testing</i>	78
IV. HASIL DAN PEMBAHASAN.....	79
4.1 Hasil Penelitian	79
4.2 Pengujian Sistem	97
V. PENUTUP.....	102
5.1 Kesimpulan	102
5.2 Saran.....	103
DAFTAR PUSTAKA	104

DAFTAR TABEL

Tabel	Halaman
1. Penelitian yang Relevan	21
2. Jadwal Penelitian	22
3. Perangkat Keras	23
4. Perangkat Lunak	23
5. Kebutuhan Fungsional	27
6. Kebutuhan Non-Fungsional.....	27
7. Daftar Aktor atau <i>Role</i>	28
8. Perancangan API.....	30
9. <i>Activity Diagram Register</i>	53
10. <i>Activity Diagram Login</i>	54
11. <i>Activity Diagram</i> Tahun Akademik	55
12. <i>Activity Diagram</i> Pengajuan Judul Skripsi.....	56
13. <i>Activity Diagram</i> Pembagian Dosen	57
14. <i>Activity Diagram</i> Bimbingan Skripsi	58
15. <i>Users</i>	62
16. <i>CoordinatorUser</i>	62
17. <i>LecturerUser</i>	63
18. <i>StudentUser</i>	64
19. <i>ThesisRegistrations</i>	65
20. <i>TitleSubmissions</i>	65
21. <i>TitleSubmission2s</i>	66
22. <i>ListOfLecturer1s</i>	66
23. <i>ListOfLecturer2s</i>	66
24. <i>ListOfLecturer3s</i>	67
25. <i>DevisionOfLecturers</i>	67

26. <i>Proposals</i>	67
27. <i>Results</i>	68
28. <i>Comprehensives</i>	69
29. <i>Pengujian Browser</i>	100

DAFTAR GAMBAR

Gambar	Halaman
1. Hasil Survei <i>Stack Overflow Javascript</i> 2022	8
2. Hasil Survei <i>Stack Overflow Node JS</i> 2022	13
3. Metode <i>Waterfall</i>	19
4. Pengambilan Data Menggunakan REST API.....	29
5. <i>Use Case Diagram</i>	52
6. <i>Sequence Diagram Login</i>	59
7. <i>Design Diagram Model View Controller (MVC)</i>	60
8. <i>Entity Relationship Diagram</i>	61
9. <i>Design</i> Halaman Awal	70
10. <i>Design</i> Halaman <i>Register</i> Mahasiswa.....	70
11. <i>Design</i> Halaman <i>Login</i>	71
12. <i>Design</i> Halaman <i>Dashboard</i> Admin.....	71
13. <i>Design</i> Halaman <i>List</i> Tahun Akademik	72
14. <i>Design</i> Halaman <i>Data Master</i>	72
15. <i>Design</i> Halaman <i>Pengajuan Judul</i>	73
16. <i>Design</i> Halaman <i>List</i> <i>Pembagian Dosen</i>	73
17. <i>Design</i> Halaman <i>Tampilan Koordinator</i>	74
18. <i>Design</i> Halaman <i>Tampilan Dosen</i>	74
19. <i>Design</i> Halaman <i>Bimbingan pada Dosen</i>	75
20. <i>Design</i> Halaman <i>Jadwal pada Dosen</i>	75
21. <i>Design</i> Halaman <i>Riwayat pada Dosen</i>	76
22. <i>Design</i> Halaman <i>Tampilan Mahasiswa</i>	76
23. <i>Design</i> Halaman <i>Pengajuan Judul</i>	77
24. <i>Design</i> Halaman <i>Bimbingan Mahasiswa</i>	77
25. Halaman <i>Utama</i>	79

26. Halaman Panduan Aplikasi	80
27. Halaman <i>About</i>	80
28. Halaman <i>Contact Us</i>	81
29. Halaman <i>Register</i>	81
30. Halaman Verifikasi <i>Email</i>	82
31. Halaman <i>Login</i>	82
32. Halaman <i>Forgot Password</i>	83
33. Halaman <i>Link Reset Password</i>	83
34. Halaman <i>Reset Password</i>	84
35. Halaman <i>Dashboard Admin</i>	84
36. Halaman <i>List Tahun Akademik</i>	85
37. Halaman <i>Data Master</i>	86
38. Halaman Judul Masuk	86
39. Halaman Judul diterima.....	87
40. Halaman Pembagian Dosen.....	87
41. Halaman Daftar Dosen	88
42. Halaman Bimbingan Masuk	88
43. Halaman Bimbingan Terjadwal	89
44. Halaman Bimbingan Selesai.....	89
45. Halaman <i>Dashboard</i> Koordinator.....	90
46. Halaman <i>Dashboard</i> Dosen.....	91
47. Halaman Permintaan Bimbingan pada Dosen	91
48. Halaman Bimbingan pada Dosen.....	92
49. Halaman Jadwal	93
50. Halaman Riwayat	93
51. Halaman <i>Dashboard</i> Mahasiswa	94
52. Halaman Pengajuan Judul	94
53. Halaman Pengajuan Dosen.....	95
54. Halaman Bimbingan Mahasiswa	95
55. Halaman <i>Profile</i> Mahasiswa.....	96
56. Halaman Ubah <i>Password</i> Mahasiswa.....	97
57. Halaman <i>Logout</i> Mahasiswa	97

58. Halaman Pengujian Pendaftaran Admin	98
59. Halaman Pendaftaran <i>User</i> Mahasiswa.....	99
60. Halaman Verifikasi <i>Email</i>	99
61. Pengujian Security menggunakan OWASP ZAP	101

DAFTAR LAMPIRAN

Lampiran	Halaman
1. Daftar Pertanyaan Wawancara Penelitian kepada Admin.....	1
2. Daftar Pertanyaan Wawancara Penelitian kepada Dosen 1.....	5
3. Daftar Pertanyaan Wawancara Penelitian kepada Dosen 2.....	12
4. Daftar Pertanyaan Wawancara Penelitian kepada Dosen 3.....	16
5. Daftar Pertanyaan Wawancara Penelitian kepada Mahasiswa 1	19
6. Daftar Pertanyaan Wawancara Penelitian kepada Mahasiswa 2	24
7. Daftar Pertanyaan Wawancara Penelitian kepada Mahasiswa 3	29
8. Daftar Pertanyaan Wawancara Penelitian kepada Mahasiswa 4	33
9. Daftar Pertanyaan Wawancara Penelitian kepada Mahasiswa 5	37
10. Pengujian Administrator dengan Admin	42
11. Pengujian <i>User</i> Koordinator Skripsi	60
12. Pengujian <i>User</i> Dosen.....	77
13. Pengujian <i>User</i> Mahasiswa	83

I. PENDAHULUAN

1.1 Latar Belakang

Teknologi informasi saat ini telah mengalami perkembangan yang semakin menampakkan eksistensinya di berbagai bidang kehidupan. Perguruan tinggi, sebagai salah satu bidang yang harus memanfaatkan teknologi informasi secara optimal, menjadi contoh yang nyata (Mulyani & Haliza, 2021). Meningkatkan investasinya dalam bidang teknologi informasi merupakan langkah yang penting untuk mendapatkan manfaat utama dari teknologi. Salah satu manfaat tersebut adalah kemudahan yang dapat diperoleh (Sari, dkk., 2019).

Pengaruh teknologi tersebut juga masuk ke sektor pendidikan, khususnya pada Program Studi Pendidikan Teknologi Informasi, Fakultas Keguruan dan Ilmu Pendidikan, Universitas Lampung. Mahasiswa diharuskan mengerjakan skripsi setelah memenuhi syarat yang telah ditetapkan oleh Universitas Lampung. Skripsi/laporan tugas akhir, tesis, dan disertasi adalah karya ilmiah individual yang disusun pada akhir studinya oleh mahasiswa program diploma, sarjana, profesi, magister, dan doktor (Peraturan Akademik Universitas Lampung, 2020).

Secara umum, penyelesaian skripsi dalam Program Studi Pendidikan Teknologi Informasi merupakan sebuah proses yang tidak mudah. Mahasiswa harus melalui serangkaian langkah yang kompleks, dimulai dari pengajuan judul skripsi hingga proses bimbingan yang menyeluruh untuk menyelesaikan skripsi tersebut. Selama ini, proses pengajuan judul skripsi dilakukan secara manual, yang melibatkan pengisian formulir pada kertas dan penggunaan *google form*. Namun, metode ini memakan waktu lebih lama dan menghasilkan struktur data yang kurang teratur. Selain itu, sulit bagi mahasiswa untuk mengetahui topik-topik yang telah diajukan sebelumnya.

Proses bimbingan skripsi di Program Studi ini masih bervariasi, pertemuan dapat dilakukan baik secara tatap muka maupun daring melalui berbagai *platform* komunikasi seperti *email*, *Zoom*, dan *WhatsApp*. Salah satu Tantangan utama yang muncul adalah adanya perbedaan kesibukan antara jadwal mahasiswa dan dosen pembimbing. Proses bimbingan tatap muka seringkali terhambat karena ketidaksesuaian jadwal, yang akhirnya menghambat kemajuan penelitian mahasiswa.

Kurangnya sistem khusus untuk memantau perkembangan skripsi di Program Studi Pendidikan Teknologi Informasi menjadi permasalahan utama. Dosen pembimbing mengalami kesulitan dalam memonitor perkembangan setiap mahasiswa, termasuk aktivitas bimbingan dan status keaktifan. Grup *WhatsApp*, yang saat ini menjadi satu-satunya media komunikasi, terbukti tidak cukup efektif untuk memantau mahasiswa secara menyeluruh. Kendala ini menunjukkan perlunya pengembangan sistem *monitoring* yang lebih terstruktur dan efisien, yang dapat memberikan aksesibilitas yang lebih baik terhadap informasi dan memungkinkan interaksi yang lebih produktif antara dosen pembimbing dan mahasiswa.

Berdasarkan masalah yang ada, peneliti mengangkat judul **“Rancang Bangun Aplikasi *Monitoring* Skripsi di Program Studi Pendidikan Teknologi Informasi”**. Peneliti bertujuan untuk meningkatkan efisiensi dalam dua aspek utama, yaitu proses pengajuan judul skripsi dan *monitoring* bimbingan mahasiswa. Implementasi aplikasi ini diharapkan memberikan dampak positif terhadap kualitas layanan di kampus.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah dalam penelitian ini yaitu bagaimana merancang serta membangun aplikasi *monitoring* skripsi di Program Studi Pendidikan Teknologi Informasi.

1.3 Tujuan Penelitian

Tujuan dalam penelitian ini adalah merancang dan membangun aplikasi *monitoring* skripsi di Program Studi Pendidikan Teknologi Informasi.

1.4 Manfaat Penelitian

1. Manfaat Praktis

a. Bagi Mahasiswa

Penelitian ini diharapkan memudahkan mahasiswa dalam pelaksanaan proses skripsi.

b. Bagi Dosen dan Admin

Penelitian ini diharapkan membuat dosen dan admin lebih mudah dalam memonitor kemajuan mahasiswa yang sudah, sedang, atau pun belum menyelesaikan skripsi untuk meningkatkan jumlah mahasiswa lulus tepat waktu.

c. Bagi Program Studi

Penelitian ini diharapkan membuat program studi Pendidikan Teknologi Informasi dalam melakukan manajemen data dan informasi menjadi lebih baik

d. Bagi Peneliti Lain

Penelitian ini dapat dijadikan referensi mulai dari isi hingga metode yang digunakan. Selain itu, peneliti juga berharap agar penelitian ini dapat memberikan motivasi kepada peneliti lain untuk lebih baik dalam merancang bangun aplikasi inovatif lainnya.

2. Manfaat Teoritis

Penelitian ini secara teoritis bertujuan memberikan kontribusi signifikan dalam pengembangan aplikasi *monitoring* skripsi di perguruan tinggi. Implementasi yang dihasilkan diharapkan dapat memperkuat pemahaman tentang integrasi teknologi informasi yang dapat mendorong kemajuan dalam bidang tersebut dan meningkatkan kesadaran pentingnya teknologi dalam peningkatan kualitas layanan akademik.

1.5 Ruang Lingkup

Ruang lingkup penelitian ini adalah:

1. Merancang dan membangun aplikasi ini dilakukan di Program Studi Pendidikan Teknologi Informasi, Fakultas Keguruan dan Ilmu Pendidikan, Universitas Lampung.
2. Informasi yang disajikan pada aplikasi *monitoring* skripsi antara lain pendaftaran, pengajuan judul skripsi, pengajuan dosen pembimbing dan pembahas, bimbingan dengan dosen pembimbing yang telah ditetapkan, serta mahasiswa dapat mengakses revisi yang diberikan oleh dosen pembimbing.
3. Metode pengembangan sistem menggunakan metode *waterfall*.

II. TINJAUAN PUSTAKA

2.1 Aplikasi *Monitoring*

Prasetyo, dkk. (2020) mengatakan bahwa *monitoring* adalah proses yang dilakukan secara teratur dengan menggunakan pendengaran, pengamatan, dan pencatatan untuk memantau perkembangan suatu program. Pernyataan ini sejalan dengan seri monograf 3 dari UNESCO *Regional Office for Education in Asia and the Pacific*. Seri tersebut menjelaskan bahwa *monitoring* adalah upaya untuk mengidentifikasi pelaksanaan program yang telah direncanakan, dijadwalkan, dan kemajuan dalam mencapai tujuan program. Pengertian lain juga dikemukakan oleh Munazilin, dkk. (2017) yang memberikan pengertian bahwa *monitoring* berfokus pada kegiatan yang sedang berlangsung untuk memastikan kesesuaian dengan perencanaan dan prosedur yang telah ditetapkan.

Berdasarkan uraian pendapat tentang *monitoring* di atas, dapat disimpulkan bahwa aplikasi *monitoring* berarti suatu *software* yang digunakan untuk menganalisis dan mengumpulkan sebuah informasi yang telah ditentukan. Aplikasi *monitoring* yang dibuat dalam penelitian ini adalah aplikasi *monitoring* skripsi yang berguna untuk memantau proses skripsi mahasiswa. Informasi yang dimaksud mencakup pendaftaran skripsi, pengajuan judul skripsi, pengajuan dosen pembimbing dan pembahas, penerimaan judul skripsi serta penetapan dosen pembimbing dan pembahas, bimbingan dengan dosen yang telah ditetapkan, serta mahasiswa dapat mengakses revisi yang diberikan oleh dosen pembimbing melalui aplikasi *monitoring* skripsi berbasis *website*.

2.2 Skripsi

Skripsi atau karya tulis ilmiah merupakan sebuah tugas yang disusun oleh seorang mahasiswa untuk memenuhi persyaratan dalam memperoleh gelar yang sesuai dengan bidang keilmuannya (Zulkarnaini, 2020). 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.

Sementara itu, Mansyur (2018) mengatakan bahwa terdapat beberapa standar yang digunakan untuk mengukur keilmiahannya sebuah karya tulis di antaranya adalah:

1. Empiris, adanya bukti yang diperoleh dari hasil observasi, eksperimen, dan sejenisnya.
2. Objektif, yang berarti data yang ditemukan menggambarkan fakta.
3. Sistematis, analisis dan penyusunan data dilakukan dengan menggunakan sistem yang teratur. Hasil deskripsi juga disajikan secara sistematis.

Tahapan skripsi di Program Studi Pendidikan Teknologi Informasi, Fakultas Keguruan dan Ilmu Pendidikan, Universitas Lampung antara lain:

1. Pengajuan judul skripsi sekaligus pemilihan dosen pembimbing dan pembahas. Keputusan diterima, revisi, atau ditolak judul skripsi ditentukan oleh ketua program studi. Setelah ditentukan, mahasiswa dapat melanjutkan skripsinya.
2. Seminar proposal merupakan tahap setelah mahasiswa menyelesaikan proposal dan mendapatkan persetujuan dari dosen pembimbing. Seminar ini bertujuan mendapatkan masukan dari mahasiswa, dosen pembimbing, dan dosen pembahas untuk memperbaiki rencana penelitian sebelum pelaksanaan.
3. Seminar hasil merupakan tahap memaparkan hasil penelitian sesuai perencanaan dan mendapatkan masukan dari dosen pembimbing serta dosen pembahas untuk memperbaiki pembahasan hasil penelitian.

4. Ujian komprehensif adalah ujian akhir setelah tahapan seminar proposal dan seminar hasil. Ujian ini bertujuan menguji kemampuan dan penguasaan mahasiswa dalam bidang keilmuannya secara menyeluruh.

Berdasarkan definisi dan uraian di atas, dapat disimpulkan bahwa skripsi menjadi persyaratan wajib bagi mahasiswa untuk menuntaskan studi S-1 dan memperoleh gelar sarjana. Tahapannya melibatkan pengajuan judul skripsi, seminar proposal, seminar hasil, hingga sidang komprehensif, sesuai prosedur yang diterapkan di Universitas Lampung.

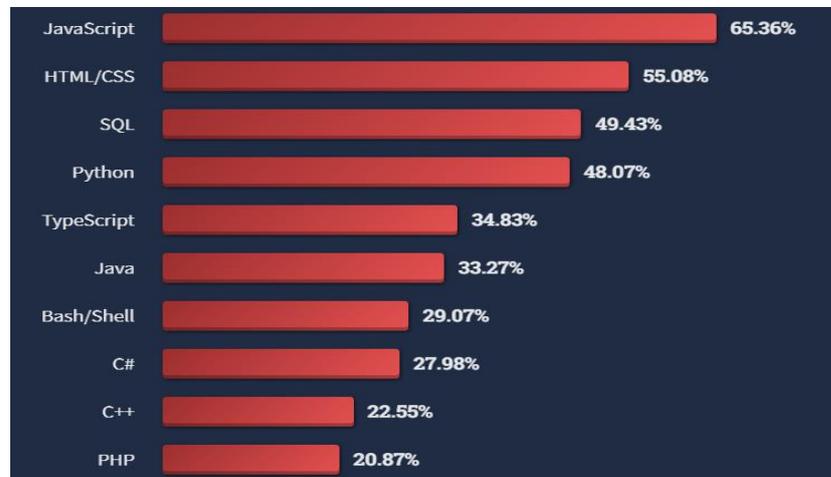
2.3 Pemrograman *Frontend*

Frontend merupakan antarmuka *interface* untuk meningkatkan tampilan *website* agar lebih menarik dan memudahkan *user* dalam berinteraksi dengan *website* (Prawastiyo & Hermawan, 2020). Pernyataan ini sejalan dengan pendapat Sari *et al.* (2022) yang mengatakan bahwa *frontend* mengacu pada desain *layout* suatu aplikasi atau *website*, *frontend* bertujuan agar desain terlihat menarik dan tidak membosankan. Pengertian lain juga diungkapkan oleh Widhyaestoeti, dkk. (2021) yang mengatakan bahwa antarmuka (*user interface*) yang disebut *frontend*, merupakan bagian antarmuka pengguna, menyediakan tampilan bagi pengguna untuk mengembangkan komponen visual pada sistem dan bertanggung jawab atas tampilan antarmuka.

2.3.1 *Javascript*

Javascript adalah bagian dari tiga teknologi penting yang harus dikuasai oleh *programmer web*, yaitu HTML untuk isi konten, CSS untuk tampilan, dan *Javascript* untuk interaksi (Kurniawan & Rosa, 2020). Pernyataan lain juga dikemukakan oleh Sari & Hidayat (2022) yang mengungkapkan bahwa dengan menggunakan *javascript* menjadi semakin mudah untuk berinteraksi dengan API. *Javascript* adalah bahasa yang berkembang, yang didefinisikan oleh serangkaian standar yang dikenal sebagai *script ECMA* oleh Ecma Internasional. *Javascript*

menjadi satu-satunya bahasa yang dipahami oleh *browser* (Matuszek, 2023).



Gambar 1. Hasil Survei Stack Overflow Javascript 2022

Kelebihan *javascript* menurut Rohman (2020) adalah:

1. Tidak memerlukan *compiler* karena dapat diinterpretasikan oleh *web browser* bersama HTML.
2. Mudah dipelajari dibandingkan dengan bahasa pemrograman lainnya.
3. Mudah menemukan dan menangani kesalahan (*error*).
4. Memungkinkan pengalihan tugas ke *event* tertentu atau elemen halaman *web*.
5. Kompatibel dengan berbagai *platform*, *browser*, dan lainnya.
6. Mengurangi pemeriksaan dan memvalidasi *input* data secara manual.
7. Membuat *website* menjadi lebih interaktif.
8. Ringan dan cepat dibandingkan bahasa pemrograman lainnya.

Kekurangan *javascript* menurut Rohman (2020) adalah:

1. Memiliki resiko tinggi dalam eksploitasi.
2. Dimanfaatkan dalam mengaktifkan sebuah kode yang memiliki kategori berbahaya pada komputer *user*.
3. Tidak selalu didukung oleh berbagai *browser*.
4. *Code snippet* lumayan banyak.
5. Dapat di-*render* pada perangkat yang berbeda.

Berdasarkan penjelasan di atas, disimpulkan bahwa *javascript* adalah bahasa pemrograman yang tidak asing bagi *frontend developer* untuk menghasilkan *website* yang lebih *powerfull*. *Javascript* merupakan pilihan yang tepat bagi *frontend web developer*, karena tidak perlu menggunakan bahasa pemrograman yang berbeda untuk mengembangkan *backend*.

2.3.2 *HyperText Markup Language 5 (HTML5)*

HTML5 telah menjadi evolusi terbaik dari HTML hingga saat ini, meningkatkan interaktivitas pengguna secara signifikan dan mengurangi beban pada perangkat (Uzayr, 2023). HTML5 sepenuhnya mendukung berbagai jenis aplikasi media dan memberikan dukungan penuh untuk menjalankan *Javascript* di latar belakang.

Sebagaimana dijelaskan dalam buku yang berjudul “HTML *The Ultimate Guide*” oleh Bloom & Reenen (2023), HTML mengandung serangkaian kode singkat yang dikenal sebagai tag. Kode ini dinormalisasi menjadi *file* teks oleh pembuat situs *web*, disimpan sebagai *file* HTML (.html), dan dapat diakses melalui berbagai *browser*.

2.3.3 *Cascading Style Sheets (CSS)*

CSS, sebagai bahasa *style sheet*, digunakan bersama HTML untuk memperbaiki tampilan *markup*. Pengguna saat menjelajahi halaman *web* hanya melihat teks yang diinginkan, bukan bahasa *markup*, karena interpretasi pengalaman tersebut dilakukan oleh perangkat (Uzayr, 2023).

Kerangka kerja CSS tidak hanya memfasilitasi desain responsif, tetapi juga menyediakan tata letak simetris yang menghemat waktu pengembang. Sistem grid dan tata letak yang efisien, bersama dengan fitur komponen antarmuka pengguna, membuat kerangka kerja ini menjadi pilihan yang ideal untuk berbagai *platform* dan ukuran layar,

sehingga dapat mempercepat alur kerja pengembangan (Bloom & Reenen, 2023). Beberapa kerangka kerja terkenal dalam dunia CSS termasuk *Bootstrap*, *Foundation*, *Semantic UI*, *Materialize*, *Material Design Lite*, dan *Lightweight*.

2.3.4 Bootstrap

Bootstrap, sebagai perpustakaan CSS, mempermudah pengembangan *frontend website* dengan menyatukan CSS, HTML, dan Javascript. Sebagai pilihan utama pengembang, *Bootstrap* tidak hanya mempercepat, tetapi juga menyederhanakan proses pembuatan *frontend* pada sebuah *website* (Ahmad Martani, dkk., 2022).

2.4 Pemrograman Backend

Pemrograman *backend* menurut Nababan, dkk. (2022) mengatakan bahwa *backend* itu bekerja di balik layar dengan mengolah *database* dan *server*. Pengertian lain juga diungkapkan oleh Mubariz, dkk. (2020) bahwa *backend* berjalan di sisi *server (server-side)* dalam aplikasi berbasis *mobile* dan *web*, dengan tugas utama berinteraksi langsung dengan basis data untuk manipulasi data, sehingga *backend* tidak berinteraksi langsung dengan pengguna. Berdasarkan penelitian Syaftiaan, dkk. (2021), diketahui bahwa *backend* sangat dibutuhkan dalam sebuah pengembangan suatu sistem dan manajemen sebuah data yang terdapat pada sistem.

Berdasarkan beberapa pendapat mengenai pengertian *backend*, dapat disimpulkan bahwa *backend* bertanggung jawab untuk memenuhi kebutuhan di balik layar (tanpa berinteraksi langsung dengan pengguna), seperti penyimpanan data, pengolahan data, dan transaksi data secara aman. Hal ini bertujuan untuk mendukung kerja *frontend* agar dapat berfungsi dengan baik sesuai dengan tugasnya.

2.4.1 *Representational State Transfer (REST)*

REST adalah sebuah gaya arsitektur yang digunakan untuk mengirimkan informasi melalui protokol HTTP. Secara umum, cara kerja REST mirip dengan sebuah *website*. *Client* mengirimkan permintaan (*request*) ke *server*, dan *server* memberikan respons (*response*) kepada *client*. REST dikembangkan oleh *co-founder* yaitu Roy Fielding dari *Apache HTTP Server Project*. Arsitektur REST, di mana *server* menyediakan sebuah sumber data (*resources*) untuk *client* mengakses dan menampilkan *resource* (sumber data) tersebut. *Resource* dapat direpresentasikan dalam format JSON (Kurniawan, dkk., 2020). Tujuan REST yaitu menciptakan sistem yang memiliki kinerja cepat, baik, dan mudah dalam pertukaran dan komunikasi data (Iswardhana & Widiono, 2021) .

REST memanfaatkan empat metode pada protokol HTTP, yaitu:

1. *GET*, membaca data dari *server* REST.
2. *POST*, membuat data baru di *server* REST.
3. *PUT*, memperbaiki data di *server* REST.
4. *DELETE*, menghapus data dari *server* REST.

2.4.2 *Application Programming Interface (API)*

Muri, dkk. (2019) mengungkapkan bahwa *Application Programming Interface (API)* adalah antarmuka pemrograman aplikasi yang memungkinkan akses dan pemanfaatan aplikasi oleh pihak lain tanpa perlu mengubah struktur kode utama atau *database* sistem. Sementara menurut Zein (2018), API dapat mengintegrasikan dari berbagai aplikasi secara bersamaan untuk bertukar data, seperti integrasi *login* menggunakan *google* atau mengirim pesan verifikasi melalui *gmail*.

Temuan tersebut sejalan dengan penelitian Hasanuddin, dkk. (2022) yang mengatakan bahwa API memberikan akses program lain untuk menggunakan fungsi dari aplikasi yang sudah ada, menghindari pembuatan ulang dari awal.

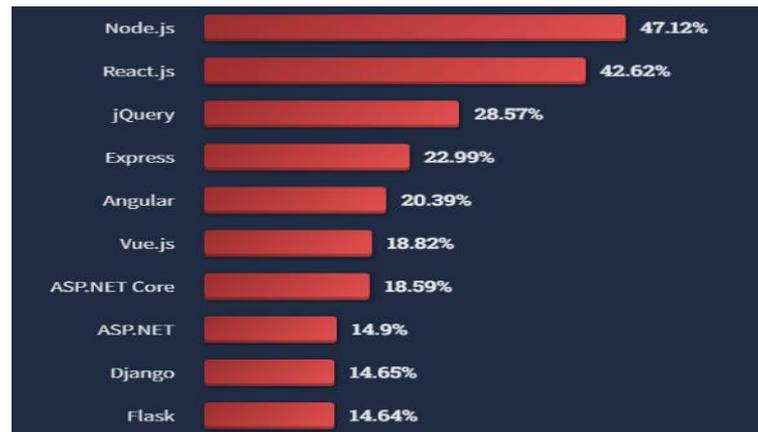
Berdasarkan uraian di atas, dapat disimpulkan bahwa API merupakan mekanisme untuk komunikasi antara dua komponen perangkat lunak melalui protokol HTTP. Arsitektur ini, yang mengirimkan *request* disebut *client*, sedangkan yang mengirimkan *response* disebut *server*. Komunikasi terjadi melalui pertukaran *request* dan *response* menggunakan HTTP.

2.4.3 Node JS

Node JS adalah sebuah *platform* pengembangan sumber terbuka yang digunakan untuk menjalankan kode sisi *server* menggunakan bahasa pemrograman *javascript*. Tahun 2009, Ryan Dahl menciptakan *Node JS* sebuah teknologi yang sangat diantisipasi oleh para pengembang *web*. Tidak disangka, teknologi ini telah menjadi sangat populer dan berguna dalam melakukan pengembangan aplikasi yang membutuhkan koneksi yang kontinu antara *browser* dan *server*.

Node JS dijalankan secara asinkron, dengan setiap permintaan ditempatkan dalam antrian acara yang terus berjalan. Pendekatan ini memungkinkan pengiriman permintaan satu per satu tanpa menunggu respons, berbeda dengan model sinkron yang membutuhkan proses yang kompleks dan menunggu respons sebelum melanjutkan permintaan lainnya (Shah & Soomro, 2017). Penelitian yang dilakukan oleh Saundariya *et al.* (2021) *Node JS* dijalankan secara asinkron, menghasilkan kinerja yang optimal dan memungkinkan untuk meningkatkan kapasitas sistem yang baik.

Bahasa pemrograman *javascript* umumnya digunakan di sisi *client* atau *browser*, namun *Node JS* memiliki peran penting dalam melengkapi *javascript* sehingga dapat berjalan di sisi *server*, seperti halnya *Ruby*, *PHP*, *Perl*, dan lainnya (Kurniawan, dkk., 2020).



Gambar 2. Hasil Survei Stack Overflow Node JS 2022

Gambar 2 menyajikan hasil survei yang dilakukan oleh *stack overflow* pada tahun 2022, bahwa popularitas *Node JS* meroket menjadi *framework* nomor satu yang banyak digunakan oleh *developer*. Berdasarkan informasi di atas, dapat disimpulkan bahwa *Node JS* telah berhasil menjadi *runtime javascript* yang memungkinkan eksekusi kode *javascript* di luar *browser*, dan menjadi pintu gerbang bagi para pengembang *javascript* untuk mengembangkan sistem di luar lingkungan *browser*.

2.4.4 Express JS

Express JS adalah sebuah *framework* yang memudahkan *web developer*, *routing*, *service API*, dan keamanan. Keunggulan *Express JS* terletak pada penggunaan pola desain yang bisa disesuaikan dengan berbagai arsitektur, membuatnya menjadi *framework* yang sangat kuat dan fleksibel. Pengembangan arsitektur dengan *Express JS* tidak memakan banyak sumber daya dan menjadi lebih ringan, sehingga dapat mengurangi biaya yang dikeluarkan dalam pengembangan situs *web* selanjutnya (Fajrin, 2017). Pengaturan *middleware* pada permintaan HTTP dalam *routing* memiliki fungsi yang berbeda dengan metode URL atau HTTP yang digunakan. *Express JS* bersifat dinamis dan kuat karena di dalam *Node Package Manager* (NPM) terdapat modul yang tersedia yang dapat diintegrasikan (Munawar, 2018).

2.4.5 PostgreSQL

PostgreSQL, menurut Obe dan Hsu (2016) dalam buku “*PostgreSQL: Up and Running*” merupakan sistem manajemen basis data (DBMS) yang didistribusikan bebas dengan lisensi BSD. *Database* ini merupakan salah satu pilihan yang populer, bersama dengan *MySQL* dan *Oracle*. Pernyataan ini selaras dengan ungkapan Rahardja (2022) yang mengatakan bahwa *postgreSQL* menjadi *database* paling banyak digunakan saat ini, bersanding dengan *MySQL* dan *Oracle*. *PostgreSQL* memiliki fitur-fitur yang berguna, termasuk fitur replikasi *database*. Berdasarkan penelitian oleh Praba & Safitri (2020), diketahui bahwa *postgreSQL* lebih baik dalam mengelola *database* skala besar, dengan beberapa *query* yang lebih unggul dibandingkan *MySQL*.

Berdasarkan pengertian di atas, dapat disimpulkan bahwa *postgreSQL* merupakan sistem manajemen *database* relasional (RDBMS) yang bersifat *open source*, yang memiliki banyak fitur canggih untuk membuat pengelolaan data lebih mudah. *postgreSQL* banyak digunakan pada *web app* dan *mobile*. Oleh karena itu, *PostgreSQL* lebih cocok untuk aplikasi yang membutuhkan pengolahan data yang lebih kompleks.

2.4.6 Sequelize

Sequelize yaitu sebuah ORM yang dapat mempermudah pemetaan *database* menjadi *object*, sehingga mempercepat dan mempermudah proses interaksi dengan *database* (Suyasa, 2018). Fitur ini menyediakan dukungan untuk transaksi yang kuat, pengambilan data (*lazy loading*), dan pengambilan data yang cepat (*eager loading*) (Hasanuddin, dkk., 2022). *Sequelize* ORM bekerja dengan *database* dan relasi-relasi di dalamnya, sehingga saat *deploy* tidak melakukan perubahan konteks saat menuliskan kode, karena melalui API sudah membuat interaksi dengan menggunakan bahasa *javascript* yang telah disediakan oleh *sequelize*.

Fitur terbaik *sequelize* yaitu dapat menggunakan API dan *database* yang berbeda (Fahmi, 2021).

2.4.7 JavaScript Object Notation (JSON)

JSON adalah format ringan untuk menyimpan data dalam variabel. Mudah diproses karena kemampuannya dalam parsing, juga mudah dipahami dan diimplementasikan oleh manusia (Brata, 2015). Pengertian lain oleh Dewi, dkk. (2019) yang mengungkapkan bahwa JSON tidak tergantung pada bahasa pemrograman tertentu, karena menggunakan gaya bahasa pemrograman yang umum digunakan oleh banyak *programmer*, seperti *C*, *C++*, *Java*, *Javascript*, *Perl*, *Python*, dan lainnya. Oleh karena itu, JSON menjadi format teks yang ideal untuk pertukaran data.

Pendapat tersebut selaras dengan pendapat Hasanuddin, dkk. (2022) yang mengungkapkan bahwa JSON termasuk turunan dari *javascript*, namun dapat digunakan pada bahasa pemrograman lainnya termasuk *Python*, *Ruby*, *PHP*, dan *Java*. File JSON memiliki ekstensi *.json* saat digunakan secara mandiri. Namun, jika didefinisikan dalam format lain seperti *.html*, JSON dapat muncul di dalam tanda petik sebagai *JSON string* atau dimasukkan ke dalam variabel. Format teks ini mudah untuk ditransfer antara *server web* dan *client/browser*.

Berdasarkan uraian di atas, dapat disimpulkan bahwa JSON adalah format pertukaran data yang digunakan untuk mengirim data dari *server* ke *client*, sehingga dapat ditampilkan pada halaman *web*, atau sebaliknya. Penyimpanan data JSON menggunakan metode *object* dan *array*.

2.4.8 Group Inclusive Tour (GIT)

GIT merupakan *version control system* yang digunakan dalam perancangan REST API. Fungsinya mencakup memberikan catatan atau tanda pada baris kode program yang mengalami perubahan,

memudahkan pelacakan perubahan pada *source code*. GIT memfasilitasi kolaborasi tim dan berfungsi sebagai perantara untuk mengirim perubahan ke *cloud server*, memungkinkan konsumsi REST API oleh *frontend*.

2.4.9 Node Package Manager (NPM)

Nasution (2021) mengungkapkan bahwa *Node Package Manager* (NPM) digunakan oleh pengembang *Javascript* untuk berbagi modul kode, mempermudah dan mempercepat proses pengembangan perangkat lunak. Melalui NPM, pengembang tidak perlu memulai kode dari awal mereka dapat menginstal paket yang dibutuhkan dan memodifikasi kode paket tersebut jika diperlukan. Implementasi NPM cukup mudah, seperti menginstal paket dengan perintah `npm install <package-name>` misalnya, untuk menginstal *framework Express.js*, dapat dilakukan dengan `npm install express` atau `npm i express`.

Penelitian oleh Alcantara, *et al.* (2021) menunjukkan bahwa NPM merender antarmuka pengguna berbasis grafis atau *Graphical User Interface* (GUI) setelah diluncurkan oleh pengguna dari dalam *Integrated Development Environment* (IDE) yang terkonfigurasi secara terintegrasi untuk menerbitkan paket NPM yang dibuat.

2.4.10 JSON Web Token (JWT)

JWT yaitu sebuah konsep token dengan format JSON yang digunakan sebagai mekanisme autentikasi antara *client* dan *server* dalam API (Satria, dkk., 2018). JWT digunakan sebagai token autentikasi yang aman dalam komunikasi antara *client* dan *server*. Penggunaan JWT memungkinkan adanya keamanan dalam *request* API, sehingga dapat mengurangi resiko terjadinya hal-hal yang tidak diinginkan saat API aktif (Chandra & Tan, 2020). Penelitian Kurniawan, dkk. (2020) menjelaskan bahwa JWT merupakan sebuah *string* panjang yang sangat acak,

digunakan untuk proses autentikasi dan pertukaran informasi. Berbeda dengan *login* pada aplikasi *website* biasa yang menggunakan *session* untuk mengingat pengguna yang sedang *login* pada JWT, pengguna dapat mengunjungi *jwt.io* untuk melakukan *login* pada *website* resmi JWT.

2.4.11 Bcrypt

Bcrypt digunakan untuk melindungi dari serangan *rainbow table*, yang merupakan tabel yang telah dihitung sebelumnya untuk membalikkan fungsi *hash kriptografi* dan sering digunakan untuk mencoba memecahkan *hash* kata sandi (Chandra & Tan, 2020). Pendapat tersebut selaras dengan Rompis & Aji (2018) yang mengatakan bahwa proses *login* melibatkan akses ke basis data serta eksekusi algoritma *bcrypt* untuk membandingkan *hash password*. *Bcrypt* adalah algoritma *kriptografi* yang lebih kompleks dibandingkan dengan verifikasi JWT.

Penelitian oleh Giffary & Ramadhani (2022) menyarankan penggunaan *password* yang terdiri dari berbagai jenis karakter, seperti huruf kecil, huruf kapital, angka, dan simbol. Penggunaan *password* dengan karakter yang beragam membuatnya lebih sulit untuk ditebak oleh peretas. Penelitian Batubara, *et al.* (2021) yang berjudul “*Analysis Performance Bcrypt Algorithm to Improve Password Security from Brute Force*”, *bcrypt* dijelaskan sebagai fungsi *hashing* yang dibuat dari algoritma *blowfish* oleh Niels Provos dan David Mazieres, dua peneliti keamanan komputer. Fungsi *hashing* ini memiliki keunggulan dalam penggunaan *salt acak* (*salt* merupakan urutan tambahan yang ditambahkan ke dalam kata sandi bertujuan agar lebih sulit untuk ditembus dengan metode *brute force*).

2.5 Unified Modeling Language (UML)

Unified Modelling Language (UML) adalah sebuah bahasa pemodelan yang digunakan dalam pengembangan sistem terintegrasi berbasis objek dalam

pengembangan perangkat lunak (Devianty & Nur Ibrahim, 2021). Aulia (2022) mengungkapkan bahwa tujuan utama UML yaitu membantu tim pengembang dalam berkomunikasi, mengeksplorasi desain, dan memvalidasi desain arsitektur perangkat lunak. Buku yang berjudul “*UML Diagramming a Case Study Approach*” menjelaskan bahwa UML adalah *diagram* untuk mendesain, membantu dalam pemahaman yang lebih baik tentang perangkat lunak atau sistem yang dikembangkan. Diagram UML seperti *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram* (Sundaramoorthy, 2022).

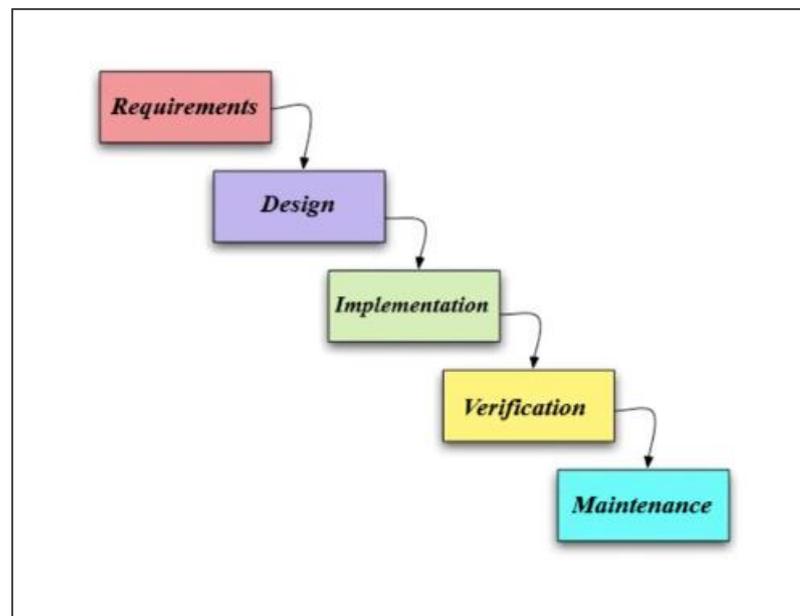
2.6 Postman

Postman adalah aplikasi yang digunakan untuk menguji API yang telah dibuat. Menggunakan *postman*, pengguna dapat dengan cepat dan secara kompleks menampilkan hasil dari *HTTP request* (Wintana, dkk., 2022). *Postman* biasanya digunakan untuk menguji berbagai *web service*, termasuk API. Pengujian yang dilakukan dengan mengakses *resources* untuk mengetahui sebuah keamanan API yang telah dibangun (Tedyyana, dkk., 2021). Sementara menurut Kore (2022), *Postman* adalah salah satu alat otomatisasi dan dokumentasi API terkemuka yang digunakan oleh 5 juta pengembang dan lebih dari 100.000 perusahaan di seluruh dunia. Terkenal sebagai solusi pengujian API lengkap, *Postman* menyediakan fungsionalitas canggih, kekuatan, dan fleksibilitas untuk memenuhi kebutuhan pengujian API.

Melalui analisis berbagai perspektif yang ada, dapat disimpulkan bahwa *postman* adalah *tools* yang sangat sesuai untuk melakukan pengujian API dengan cepat kompleks. *Tools postman* sangat *powerfull* dan mudah dipahami oleh pemula. *Postman* tersedia secara gratis dan dapat dijalankan pada berbagai *operating system*, termasuk *Linux*, *Windows*, dan *macOS*. Pengujian API ini bertujuan untuk mengetahui apakah hasil *response* sesuai dengan permintaan *HTTP request*.

2.7 Metode Pengembangan *Waterfall*

Metode Pengembangan pada penelitian ini menggunakan metode pengembangan sistem, yaitu *Waterfall*. Menurut buku *Software Engineering* yang berjudul “*A Practitioner's Approach*” yang ditulis oleh Roger S. Pressman, *waterfall* adalah salah satu pendekatan tradisional dalam pengembangan perangkat lunak yang terstruktur. Metode *waterfall* dijelaskan dengan langkah-langkah *Requirements*, *Design*, *Implementation*, *Verification*, dan *Maintenance* (Pressman, 2012).



Gambar 3. Metode *Waterfall*

Tahap-tahap dalam metode *waterfall* adalah:

1. *Requirements* (Kebutuhan)

Tahap ini fokus pada pengumpulan, analisis, dan dokumentasi kebutuhan pengguna dan sistem yang dikembangkan. Tujuannya adalah memahami secara menyeluruh kebutuhan yang mencakup deskripsi lengkap yang harus dicapai oleh sistem.

2. *Design* (Desain)

Tahap ini, setelah pengumpulan kebutuhan, dilakukan desain keseluruhan sistem dan komponennya berdasarkan kebutuhan yang telah ditetapkan. Hal ini melibatkan desain arsitektur, antarmuka, dan elemen-elemen teknis

lainnya. Tujuannya adalah merancang solusi yang memenuhi kebutuhan pengguna dengan memperhatikan aspek teknis.

3. *Implementation* (Implementasi)

Tahap ini melibatkan penulisan kode dan pengembangan perangkat lunak berdasarkan desain yang telah dibuat sebelumnya. *Developer* menerjemahkan desain menjadi kode yang dapat dieksekusi.

4. *Verification* (Verifikasi)

Pengujian perangkat lunak dilakukan pada berbagai level, mulai dari pengujian unit hingga pengujian integrasi dan sistem secara keseluruhan, untuk memastikan bahwa sistem berfungsi sesuai dengan kebutuhan yang ditetapkan. Tujuan tahap ini adalah memastikan kualitas dan keandalan sistem.

5. *Maintenance* (Pemeliharaan)

Setelah implementasi, tahap pemeliharaan dimulai. Sistem diperbaiki, ditingkatkan, dan dikelola untuk memenuhi perubahan kebutuhan pengguna atau memperbaiki *bug* yang muncul selama penggunaan. Tujuannya adalah menjaga kinerja sistem dan memenuhi kebutuhan seiring waktu. Tahap-tahap ini berjalan berurutan, di mana setiap tahap bergantung pada penyelesaian tahap sebelumnya

2.8 Penelitian Terkait

Terdapat penelitian terkait yang perlu dipertanggungjawabkan dalam hal akademis, penelitian-penelitian terkait yang telah dilakukan sebelumnya telah menjadi referensi yang digunakan oleh peneliti dalam penelitian yang sedang dilakukan disajikan oleh Tabel 1.

Tabel 1. Penelitian yang Relevan

No	Nama Peneliti	Tahun Penelitian	Judul Penelitian	Metode	Hasil Penelitian
1.	Hasanuddin, Hari Asgar, dan Budi Hartono	2022	Rancang Bangun REST API Aplikasi Weshare sebagai Upaya Mempermudah Pelayanan Donasi Kemanusiaan	Metode <i>System Development Life Cycle</i> (SDLC)	Sistem dengan menggunakan arsitektur REST dalam <i>Backend Development</i> yang berbasis API yang berguna untuk mempermudah masyarakat dalam melakukan transaksi donasi dengan menggunakan antarmuka pengguna yang tersedia pada aplikasi android dan situs <i>web</i> .
2.	Siti Sauda dan M. Barokah	2022	Penerapan <i>Node JS</i> dan <i>postgreSQL</i> sebagai <i>Backend</i> pada Aplikasi <i>Ecommerce</i> Localla	Metode <i>scrum</i>	Pengembangan <i>backend</i> dapat dilakukan dengan sukses dan REST API yang telah dibuat dapat diakses oleh <i>frontend</i> .
3.	Irfan Kurniawan, Humaira, dan Fazrol Rozi	2022	REST API Menggunakan <i>Node JS</i> pada Aplikasi Transaksi Jasa Elektronik Berbasis <i>Android</i>	Metode <i>System Development Life Cycle</i> (SDLC)	Sistem dengan arsitektur REST pada <i>backend</i> yang berbasis API yang diterapkan pada aplikasi <i>android</i> sebagai antarmuka pengguna untuk memudahkan proses transaksi jasa.
4.	I Gede Alvian Raynaldi	2021	Implementasi <i>Web Service</i> Untuk Aplikasi Proses Skripsi Menggunakan REST API	Metode <i>Waterfall</i>	Aplikasi proses skripsi dengan penerapan layanan <i>web</i> menggunakan REST API bertujuan untuk otomatisasi proses skripsi yang sebelumnya dilakukan secara manual.
5.	Nadia Mustika Sari, Erly Krisnanik, dan Lomo Mula Tua.	2019	Sistem Informasi <i>Monitoring</i> Bimbingan Skripsi / Tugas Akhir (SIMP-S/TA) Berbasis <i>Android</i>	Metode <i>Waterfall</i>	Sistem informasi yang digunakan untuk proses pengajuan proposal serta <i>monitoring progress</i> tugas akhir secara <i>online</i> .

3.3 Alat dan Bahan

Kebutuhan sistem yang digunakan dalam pembuatan Aplikasi *Monitoring Skripsi* di Program Studi Pendidikan Teknologi Informasi, peneliti menggunakan beberapa alat dan bahan digunakan untuk menunjang kebutuhan yang terdiri dari perangkat keras dan perangkat lunak.

3.3.1 Perangkat Keras (*Hardware*)

Tabel 3. Perangkat Keras

No	Perangkat Keras (<i>Hardware</i>)	Keterangan
1.	Laptop <i>personal</i>	Asus A15
2.	<i>Processor</i>	AMD <i>Ryzen 7 4800H</i> dengan <i>Radeon Graphics</i>
3.	<i>Random Access Memory (RAM)</i>	8 GB
4.	<i>Video Graphics Array (VGA)</i>	NVIDIA <i>Geforce RTX</i>
5.	<i>Solid State Drive (SSD)</i>	512 GB
6.	<i>Monitor</i>	Resolusi 1920 x 1080 px., <i>Mouse</i> dan <i>keyboard</i>
7.	Perangkat	standar

Tabel 3 menyajikan perangkat keras yang digunakan dalam pembuatan aplikasi, mencakup laptop yang digunakan, spesifikasi *processor*, RAM, VGA, SSD, *monitor*, serta perangkat *input* yang terpasang.

3.3.2 Perangkat Lunak (*Software*)

Tabel 4. Perangkat Lunak

No	Perangkat Lunak (<i>Software</i>)	Keterangan
1.	<i>Windows 11</i>	Sistem Operasi
2.	<i>Javascript, HTML</i> dan <i>CSS</i>	Bahasa pemrograman
3.	<i>PostgreSQL</i> dan <i>DBeaver</i>	<i>Database</i>
4.	<i>Chrome</i> dan <i>Firefox</i>	<i>Web browser</i>
5.	<i>Visual Studio Code</i>	<i>Text Editor</i>
6.	<i>Express JS</i> dan <i>Node JS</i>	<i>Framework</i>
7.	<i>Sequelize</i>	<i>Object Relational Mapping (ORM)</i>
8.	<i>UML</i> dan <i>Draw.io</i>	Pemodelan data dan ERD
9.	<i>Balsamiq</i>	<i>Design Antarmuka</i>
10.	<i>Postman</i>	<i>Testing</i>

Tabel 4 menyajikan informasi tentang perangkat lunak yang digunakan dalam pembuatan aplikasi, mencakup sistem operasi, bahasa pemrograman, *database*, *web browser*, *text editor*, *framework*, ORM, alat pemodelan data, alat desain antarmuka, dan alat pengujian yang digunakan dalam pengembangan aplikasi.

3.4 Metode Pengumpulan Data

Metode pengumpulan data merupakan bagian terpenting dalam sebuah penelitian itu sendiri. Metode pengumpulan data yang digunakan dalam penelitian ini adalah observasi dan wawancara. Berikut penjelasan dari metode yang peneliti lakukan.

3.4.1 Observasi

Penelitian ini mendapatkan informasi dan data dengan cara melakukan observasi. Observasi merupakan salah satu metode pengumpulan data pada penelitian ini. Observasi dilakukan pada program yang berkaitan dengan topik pembahasan dalam permasalahan penelitian ini. Hasil observasi yang telah dilakukan bahwa pada program studi pendidikan teknologi informasi dalam pelaksanaan proses skripsi belum menggunakan sistem khusus mulai dari proses pengajuan judul skripsi, proses bimbingan sampai dengan proses *monitoring* skripsi mahasiswa. Pengajuan judul yang masih dilakukan dengan menggunakan *form* yang dicetak/diprint selain itu juga mengisi pengajuan melalui *google form*. Proses bimbingan yang dilakukan secara langsung dengan mahasiswa mencetak skripsi, sedangkan bimbingan tidak langsung atau *online* dilakukan melalui *whatsapp*, *gmail*, atau *zoom*. Sementara untuk proses *monitoring* yang dilakukan masih dilakukan secara manual dengan menggunakan *whatsapp*.

3.4.2 Wawancara (*Interview*)

Metode pengumpulan data dilakukan dengan mengadakan wawancara secara langsung dengan 1 admin program studi, 3 dosen, dan 5 mahasiswa. Wawancara yang dilakukan mengenai informasi kegiatan proses pelayanan skripsi, seperti alur pengajuan judul skripsi, penerimaan judul skripsi, penentuan dosen pembimbing dan pembahas serta proses bimbingan yang dilakukan mahasiswa dengan dosen yang telah ditetapkan. Sehubungan dengan masalah yang diteliti untuk mendapatkan data yang objektif. Metode wawancara ini peneliti menggunakan wawancara yang terstruktur. Pertanyaan wawancara telah dirumuskan terlebih dahulu dengan membuat daftar pertanyaan, sehingga ketika proses wawancara berlangsung dapat menggunakan daftar pertanyaan tersebut. Daftar pertanyaan dan hasil wawancara kepada admin, dosen dan mahasiswa terlampir, yaitu pada Lampiran 1 sampai Lampiran 9.

3.5 Metode Pengembangan Sistem

Metode yang digunakan dalam perancangan aplikasi *monitoring* skripsi yaitu metode *waterfall* yang memiliki beberapa tahapan di antaranya: *Analysis*, *Design*, *Implementation*, dan *Testing*. Berikut merupakan penjelasan tentang penerapan tahapan menggunakan metode pengembangan *Waterfall*.

3.5.1 *Analysis*

Tahap analisis merupakan tahap awal dalam merancang dan membangun aplikasi *monitoring* skripsi. Proses analisis yang dilakukan, yaitu melakukan analisis mengenai kondisi sebelum adanya sebuah sistem, analisis sistem yang dibangun, dan analisis kebutuhan sistem. Proses analisis, peneliti mencari dan mengenali kebutuhan sistem yang akan dibuat, langkah yang diambil adalah mengumpulkan data dan informasi terkait dengan proses pelaksanaan skripsi pada Program Studi

Pendidikan Teknologi Informasi, Fakultas Keguruan dan Ilmu Pendidikan, Universitas Lampung untuk dijadikan sebagai referensi.

Tahap pengumpulan data, digunakan teknik observasi dan wawancara dengan admin program studi, dosen, dan mahasiswa. Informasi yang diperoleh dari hasil wawancara dan observasi kemudian dianalisis untuk mengidentifikasi data yang dibutuhkan *user*. Penjelasan mengenai admin program studi dalam konteks ini adalah yang bertanggung jawab atas administrasi skripsi mahasiswa di perguruan tinggi untuk memastikan kelancaran proses akademik.

Sementara itu, dosen yang dimaksud adalah dosen yang diberikan wewenang sebagai pembimbing atau pembahas terhadap progres skripsi mahasiswa. Dosen memiliki tanggung jawab memberikan arahan, umpan balik, serta mengawasi kemajuan mahasiswa dalam menyelesaikan skripsi. Adapun mahasiswa yang dimaksud adalah mahasiswa yang telah mencapai tahap akhir yang ditandai dengan telah menyelesaikan minimum 110 SKS, telah menyelesaikan mata kuliah wajib yang mendukung topik skripsi, dan telah mencapai IPK sementara minimal 2,00.

1. Kebutuhan Fungsional

Hasil dari pengumpulan kebutuhan dari pihak yang terlibat langsung dengan sistem, menghasilkan kebutuhan fungsional sebagaimana tercantum dalam Tabel 5. Kebutuhan Fungsional ini merupakan permintaan kepada *developer* untuk memastikan bahwa sistem yang dikembangkan dapat memenuhi kebutuhan dari *user*. Setelah mendapatkan kebutuhan fungsional ini, langkah selanjutnya adalah membuat aplikasi yang sesuai dengan kebutuhan tersebut.

Tabel 5. Kebutuhan Fungsional

ID Kebutuhan Fungsional	Kebutuhan Fungsional
KF1	Sistem memungkinkan mahasiswa memasukkan tahun periode saat melakukan proses registrasi.
KF2	Sistem memberikan kemampuan kepada admin/koordinator untuk melihat dan mengelola data mahasiswa.
KF3	Sistem menyediakan fitur <i>login</i> bagi dosen dengan proses verifikasi akun dan sandi.
KF4	Sistem memfasilitasi admin untuk melihat dan mengelola data koordinator.
KF5	Sistem menyediakan opsi bagi koordinator untuk melihat dan mengelola daftar dosen pembimbing dan pembahas.
KF6	Sistem memberikan akses kepada dosen/koordinator untuk melihat dan mengelola data pengajuan judul skripsi.
KF7	Sistem mendukung koordinator dalam melakukan verifikasi judul skripsi.
KF8	Sistem memberi kesempatan kepada mahasiswa untuk mengajukan dosen pembimbing dan pembahas.
KF9	Sistem mengizinkan koordinator untuk menentukan dosen pembimbing dan pembahas.
KF10	Sistem memberi kemampuan kepada dosen pembimbing dan mahasiswa untuk melihat dan mengelola bimbingan skripsi.
KF11	Sistem membolehkan dosen pembimbing untuk memberikan revisi pada skripsi.
KF12	Sistem memberi akses kepada mahasiswa untuk mengakses revisi dari dosen pembimbing.

2. Kebutuhan Non-Fungsional

Tahap ini, dilakukan pengelompokan kebutuhan berdasarkan area non-fungsional yang terkait dengan perangkat lunak. Tabel yang digunakan untuk menentukan kebutuhan non-fungsional dapat dilihat pada Tabel 6.

Tabel 6. Kebutuhan Non-Fungsional

ID Kebutuhan Non-Fungsional	Kebutuhan Non-Fungsional
KnF1	Sistem dapat diakses secara <i>online</i> .
KnF2	Aplikasi berbasis <i>web</i> dapat diakses menggunakan <i>browser chrome</i> dan

	<i>mozilla firefox.</i>
KnF3	Data <i>user</i> harus disandikan dan disimpan secara aman menggunakan enkripsi.
KnF4	Sistem harus mampu beroperasi secara <i>online</i> dengan menggunakan data atau jaringan yang rendah.
KnF5	Kode aplikasi harus terstruktur dengan baik dan terdokumentasi secara memadai untuk memfasilitasi pemeliharaan dan pengembangan di masa mendatang.

3. Daftar Aktor atau *Role*

Setiap aktor atau *role* memiliki peran dan tanggung jawab yang berbeda dalam sistem, yang sesuai dengan kebutuhan dan kontribusinya dalam proses pengelolaan skripsi. Daftar aktor atau *role* dan kepentingannya dapat dilihat pada Tabel 7.

Tabel 7. Daftar Aktor atau *Role*

Aktor atau <i>Role</i>	Kepentingan
Admin	Bertanggung jawab atas pengelolaan keseluruhan sistem.
Koordinator Skripsi	<ol style="list-style-type: none"> 1. Mengelola pembagian dosen pembimbing dan pembahas. 2. Mengelola permintaan bimbingan. 3. Mengelola daftar dosen pembimbing dan pembahas. 4. Mengelola pengajuan judul. 5. Mengelola bimbingan mahasiswa. 6. Mengelola akun koordinator.
Dosen	<ol style="list-style-type: none"> 1. Mengelola pengajuan judul. 2. Mengelola bimbingan mahasiswa. 3. Mengelola jadwal dan riwayat seminar atau ujian. 4. Mengelola akun dosen.
Mahasiswa	<ol style="list-style-type: none"> 1. Mengajukan judul. 2. Mengajukan dosen pembimbing dan pembahas. 3. Melakukan bimbingan proposal, bimbingan hasil skripsi, dan bimbingan ujian komprehensif. 4. Mengelola akun mahasiswa.

3.5.2 *Design*

Tahap *design* yang dilakukan yaitu perancangan *backend development*, perancangan arsitektur *system*, perancangan API, pemodelan sistem

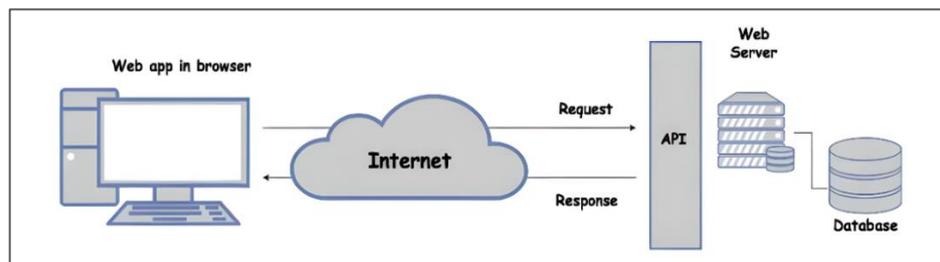
dalam bentuk *use case diagram*, *activity diagram*, *sequence diagram*, basis data, serta *design* antarmuka khusus untuk aplikasi berbasis *web*.

3.5.2.1 Perancangan *Backend Development*

Backend development merupakan tahap perancangan sebuah sistem yang dilakukan di belakang layar dari sebuah aplikasi. Bahasa pemrograman yang digunakan dalam perancangan *backend development* yaitu menggunakan bahasa *javascript* yang tentu tidak terlihat secara langsung oleh pengguna, tetapi dalam perancangan aplikasi, *backend development* memberikan kekuatan, terutama dalam pengolahan data yang ditampilkan. Perancangan *backend development* terfokus pada pengolahan *database*, pengkodean, dan arsitektur dari sebuah sistem. Pengkodean yang ditulis oleh *backend development* dapat membantu untuk mengomunikasikan informasi data yang terdapat pada *database* kepada *user*, melalui perantara *User Interface (UI)* yang disediakan *Frontend*.

3.5.2.2 Perancangan Arsitektur System

Aplikasi *client* yang dikembangkan memiliki integrasi dengan *Application Programming Interface (API)* menggunakan metode REST sebagai arsitektur utamanya.



Gambar 4. Pengambilan Data Menggunakan REST API

Gambar 4 menggambarkan proses ilustrasi pengambilan data menggunakan REST API. Pengguna mengirimkan data dalam format JSON, yang kemudian dikirimkan ke *server* API. *Server* API

melakukan permintaan kepada *database* untuk memroses data tersebut. Setelah proses permintaan selesai, hasilnya dikirimkan kembali kepada pengguna dalam format JSON. Aplikasi menguraikan data tersebut sehingga dapat digunakan dan ditampilkan kepada pengguna.

3.2.5.3 Perancangan API

Terdapat empat jenis klasifikasi berdasarkan metode permintaan (*request method*):

1. *Request method GET*, digunakan untuk untuk mengambil atau membaca data secara detail dari anggota *resource*. *Method* dalam kelompok ini biasanya mengembalikan keluaran atau hasil yang dapat dianggap sebagai fungsi.
2. *Request method POST*, digunakan untuk menambah sumber daya (*resource*).
3. *Request method PUT*, digunakan untuk memperbarui atau mengubah *item/resource* yang telah ada.
4. *Request method DELETE*, digunakan untuk menghapus *item/resource* yang telah ada.

Tabel 8. Perancangan API

No	Routes	Endpoint	Entitas	Deskripsi
1.	POST	/api/register	Admin	Register admin.
2.	POST	/register	Mahasiswa	Mahasiswa melakukan pendaftaran ke sistem.
3.	GET	/login	Admin	Admin melakukan <i>login</i> .
4.	GET	/login	Dosen	Dosen melakukan <i>login</i> .
5.	GET	/login	Mahasiswa	Mahasiswa melakukan <i>login</i> .
6.	GET	/login	Koordinator	Koordinator melakukan <i>login</i> .
7.	GET	/dashboard	Admin	Menampilkan halaman <i>dashboard</i> admin.
8.	GET	/dashboard	Dosen	Menampilkan halaman <i>dashboard</i> dosen.
9.	GET	/dashboard	Mahasiswa	Menampilkan halaman <i>dashboard</i> mahasiswa.
10.	GET	/dashboard	Koordinator	Menampilkan halaman <i>dashboard</i> koordinator.
11.	GET	/list-tahun-akademik	Admin	Admin menampilkan halaman data tahun akademik.
12.	POST	/list-tahun-	Admin	Admin dapat menambah tahun

		akademik/tambah		akademik baru.
13.	<i>POST</i>	<i>/list-tahun-akademik/edit/:id</i>	Admin	Admin melakukan perubahan pada data tahun akademik berdasarkan id.
14.	<i>POST</i>	<i>/list-tahun-akademik/edit-status/:id</i>	Admin	Admin melakukan perubahan pada data status tahun akademik berdasarkan id.
15.	<i>GET</i>	<i>/list-tahun-akademik/delete/:id</i>	Admin	Admin menghapus data tahun akademik berdasarkan id.
16.	<i>GET</i>	<i>/mahasiswa</i>	Admin	Admin dapat menampilkan data mahasiswa.
17.	<i>POST</i>	<i>/mahasiswa/tambah</i>	Admin	Admin menambah data mahasiswa baru.
18.	<i>POST</i>	<i>/mahasiswa/edit/:id</i>	Admin	Admin melakukan perubahan data mahasiswa berdasarkan id.
19.	<i>GET</i>	<i>/mahasiswa/delete/:id</i>	Admin	Admin menghapus data mahasiswa berdasarkan id.
20.	<i>GET</i>	<i>/dosen</i>	Admin	Admin dapat menampilkan data dosen.
21.	<i>POST</i>	<i>/dosen/tambah</i>	Admin	Admin menambah data dosen baru.
22.	<i>POST</i>	<i>/dosen/edit/:id</i>	Admin	Admin melakukan perubahan data dosen berdasarkan id.
23.	<i>GET</i>	<i>/dosen/delete/:id</i>	Admin	Admin menghapus data dosen berdasarkan id.
24.	<i>GET</i>	<i>/koordinator-skripsi</i>	Admin	Admin dapat menampilkan data koordinator skripsi.
25.	<i>POST</i>	<i>/koordinator-skripsi/tambah</i>	Admin	Admin menambah data koordinator baru.
26.	<i>POST</i>	<i>/koordinator-skripsi/edit/:id</i>	Admin	Admin melakukan perubahan data koordinator berdasarkan id.
27.	<i>GET</i>	<i>/koordinator-skripsi/delete/:id</i>	Admin	Admin menghapus data koordinator berdasarkan id.
28.	<i>GET</i>	<i>/daftar-judul</i>	Mahasiswa	Menampilkan daftar pengajuan judul mahasiswa.
29.	<i>GET</i>	<i>/pengajuan-judul</i>	Admin	Admin dapat menampilkan daftar pengajuan judul dari mahasiswa.
30.	<i>GET</i>	<i>/pengajuan-judul</i>	Dosen	Dosen dapat menampilkan daftar pengajuan judul dari mahasiswa.
31.	<i>GET</i>	<i>/pengajuan-judul</i>	Mahasiswa	Menampilkan halaman pengajuan judul mahasiswa.
32.	<i>GET</i>	<i>/pengajuan-judul</i>	Koordinator	Koordinator dapat menampilkan daftar pengajuan judul dari mahasiswa.
33.	<i>POST</i>	<i>/pengajuan-judul</i>	Mahasiswa	Mahasiswa dapat mengajukan judul 1.
34.	<i>POST</i>	<i>/pengajuan-judul2</i>	Mahasiswa	Mahasiswa dapat mengajukan judul 2.
35.	<i>POST</i>	<i>/pengajuan-judul/edit/:id</i>	Admin	Admin dapat melakukan perubahan data pengajuan judul

				berdasarkan id.
36.	<i>POST</i>	/pengajuan-judul/edit/:id	Dosen	Dosen dapat melakukan perubahan data pengajuan judul berdasarkan id.
37.	<i>POST</i>	/pengajuan-judul/edit/:id	Koordinator	Koordinator dapat melakukan perubahan data pengajuan judul berdasarkan id.
38.	<i>POST</i>	/pengajuan-editJudul1	Mahasiswa	Mahasiswa dapat melakukan perubahan pengajuan judul 1.
39.	<i>POST</i>	/pengajuan-editJudul2	Mahasiswa	Mahasiswa dapat melakukan perubahan pengajuan judul 2.
40.	<i>GET</i>	/pengajuan-judul/delete/:id	Admin	Admin dapat menghapus pengajuan judul mahasiswa berdasarkan id.
41.	<i>GET</i>	/pengajuan-judul/delete/:id	Koordinator	Koordinator dapat menghapus pengajuan judul mahasiswa berdasarkan id.
42.	<i>GET</i>	/judul-diterima	Admin	Admin dapat menampilkan halaman pengajuan judul yang sudah diterima.
43.	<i>GET</i>	/judul-diterima	Dosen	Dosen dapat menampilkan halaman pengajuan judul yang sudah diterima.
44.	<i>GET</i>	/judul-diterima	Koordinator	Koordinator dapat menampilkan halaman pengajuan judul yang sudah diterima.
45.	<i>GET</i>	/judul-diterima	Mahasiswa	Menampilkan halaman pengajuan judul yang sudah diterima.
46.	<i>GET</i>	/pembagian-dosen	Admin	Admin dapat menampilkan halaman pembagian dosen.
47.	<i>GET</i>	/pembagian-dosen	Koordinator	Koordinator dapat menampilkan halaman pembagian dosen.
48.	<i>POST</i>	/pembagian-dosen/tambah	Admin	Admin dapat menambah data pembagian dosen.
49.	<i>POST</i>	/pembagian-dosen/tambah	Koordinator	Koordinator dapat menambah data pembagian dosen.
50.	<i>POST</i>	/pembagian-dosen/edit/:id	Admin	Admin melakukan perubahan pembagian dosen berdasarkan id.
51.	<i>POST</i>	/pembagian-dosen/edit/:id	Koordinator	Koordinator melakukan perubahan pembagian dosen berdasarkan id.
52.	<i>POST</i>	/pembagian-dosen/editDosen/:id	Admin	Admin melakukan perubahan pada ubah pembagian dosen berdasarkan id.
53.	<i>POST</i>	/pembagian-dosen/editDosen/:id	Koordinator	Koordinator melakukan perubahan pada ubah pembagian dosen berdasarkan id.
54.	<i>GET</i>	/pengajuan-dosen	Mahasiswa	Menampilkan halaman pengajuan dosen.

55.	<i>POST</i>	/pengajuan-dosen	Mahasiswa	Mahasiswa dapat mengajukan dosen pembimbing dan pembahas.
56.	<i>GET</i>	/permintaan-bimbingan	Admin	Admin dapat menampilkan halaman permintaan bimbingan.
57.	<i>GET</i>	/permintaan-bimbingan	Dosen	Dosen dapat menampilkan halaman permintaan bimbingan.
58.	<i>GET</i>	/permintaan-bimbingan	Koordinator	Koordinator dapat menampilkan halaman permintaan bimbingan.
59.	<i>POST</i>	/permintaan-bimbingan/edit/:id	Admin	Admin melakukan perubahan status pada permintaan bimbingan.
60.	<i>POST</i>	/permintaan-bimbingan/edit/:id	Dosen	Dosen melakukan perubahan status pada permintaan bimbingan.
61.	<i>POST</i>	/permintaan-bimbingan/edit/:id	Koordinator	Koordinator melakukan perubahan status pada permintaan bimbingan.
62.	<i>GET</i>	/daftar-dosen	Admin	Admin dapat menampilkan halaman daftar dosen pembimbing 1.
63.	<i>GET</i>	/daftar-dosen	Koordinator	Koordinator dapat menampilkan halaman daftar dosen pembimbing 1.
64.	<i>POST</i>	/daftar-dosen/tambah	Admin	Admin dapat menambah data daftar dosen pembimbing 1.
65.	<i>POST</i>	/daftar-dosen/tambah	Koordinator	Koordinator dapat menambah data daftar dosen pembimbing 1.
66.	<i>GET</i>	/daftar-dosen/delete	Admin	Admin dapat menghapus data daftar dosen pembimbing 1.
67.	<i>GET</i>	/daftar-dosen/delete	Koordinator	Koordinator dapat menghapus data daftar dosen pembimbing 1.
68.	<i>GET</i>	/daftar-dosen2	Admin	Admin dapat menampilkan halaman daftar dosen pembimbing 2.
69.	<i>GET</i>	/daftar-dosen2	Koordinator	Admin dapat menampilkan halaman daftar dosen pembimbing 2.
70.	<i>POST</i>	/daftar-dosen2/tambah	Admin	Admin dapat menambah data daftar dosen pembimbing 2.
71.	<i>POST</i>	/daftar-dosen2/tambah	Koordinator	Koordinator dapat menambah data daftar dosen pembimbing 2.
72.	<i>GET</i>	/daftar-dosen2/delete	Admin	Admin dapat menghapus data daftar dosen pembimbing 2.
73.	<i>GET</i>	/daftar-dosen2/delete	Koordinator	Koordinator dapat menghapus data daftar dosen pembimbing 2.
74.	<i>GET</i>	/daftar-dosen3	Admin	Admin dapat menampilkan halaman daftar dosen pembahas.

75.	GET	/daftar-dosen3	Koordinator	Koordinator dapat menampilkan halaman daftar dosen pembahas.
76.	POST	/daftar-dosen3/tambah	Admin	Admin dapat menambah data daftar dosen pembahas.
77.	POST	/daftar-dosen3/tambah	Koordinator	Koordinator dapat menambah data daftar dosen pembahas.
78.	GET	/daftar-dosen3/delete	Admin	Admin dapat menghapus data daftar dosen pembahas.
79.	GET	/daftar-dosen3/delete	Koordinator	Koordinator dapat menghapus data daftar dosen pembahas.
80.	GET	/proposal	Admin	Admin dapat menampilkan halaman bimbingan proposal mahasiswa.
81.	GET	/bimbingan/proposal	Dosen	Dosen dapat menampilkan halaman bimbingan proposal mahasiswa.
82.	GET	/proposal	Mahasiswa	Menampilkan halaman bimbingan proposal.
83.	POST	/proposal	Mahasiswa	Mahasiswa dapat melakukan bimbingan proposal dengan mengunggah <i>file</i> proposal.
84.	GET	/proposal	Koordinator	Koordinator dapat menampilkan halaman bimbingan proposal mahasiswa.
85.	GET	/proposal/download/:id	Admin	Admin dapat mengunduh <i>file</i> bimbingan proposal mahasiswa.
86.	GET	/bimbingan/proposal/download/:id	Dosen	Dosen dapat mengunduh <i>file</i> bimbingan proposal mahasiswa.
87.	GET	/proposal/download/:id	Koordinator	Koordinator dapat mengunduh <i>file</i> bimbingan proposal mahasiswa.
88.	GET	/proposal/download/:id	Mahasiswa	Mahasiswa dapat mengunduh <i>file</i> proposal yang telah diajukan.
89.	POST	/bimbingan/proposal-revisi/tambah/:id	Dosen	Dosen dapat memberikan revisi bimbingan proposal mahasiswa.
90.	POST	/proposal/edit	Mahasiswa	Mahasiswa dapat mengajukan ulang bimbingan proposal dengan mengunggah <i>file</i> proposal baru.
91.	POST	/proposal/status-edit	Mahasiswa	Menampilkan status bimbingan proposal.
92.	POST	/proposal/edit/:id	Admin	Admin dapat melakukan perubahan pada bimbingan proposal mahasiswa.
93.	POST	/proposal/edit/:id	Koordinator	Koordinator dapat melakukan perubahan pada bimbingan proposal mahasiswa.
94.	GET	/proposal/delete/:id	Admin	Admin dapat menghapus bimbingan proposal mahasiswa.
95.	GET	/proposal/delete/:id	Koordinator	Koordinator dapat menghapus bimbingan proposal mahasiswa.

96.	<i>POST</i>	/proposal-revisi	Mahasiswa	Mahasiswa dapat menampilkan revisi dosen pembimbing 1 dan mengunggah <i>file</i> proposal revisi.
97.	<i>POST</i>	/proposal-revisi2	Mahasiswa	Mahasiswa dapat menampilkan revisi dosen pembimbing 2 dan mengunggah <i>file</i> proposal revisi.
98.	<i>GET</i>	/proposal-revisi1/download/:id	Admin	Admin dapat mengunduh <i>file</i> proposal revisi mahasiswa dengan dosen pembimbing 1.
99.	<i>GET</i>	/proposal-revisi1/download/:id	Dosen	Dosen dapat mengunduh <i>file</i> proposal revisi mahasiswa dengan dosen pembimbing 1.
100.	<i>GET</i>	/proposal-revisi1/download/:id	Koordinator	Koordinator dapat mengunduh <i>file</i> proposal revisi mahasiswa dengan dosen pembimbing 1.
101.	<i>GET</i>	/proposal-revisi2/download/:id	Admin	Admin dapat mengunduh <i>file</i> proposal revisi mahasiswa dengan dosen pembimbing 2.
102.	<i>GET</i>	/proposal-revisi2/download/:id	Dosen	Dosen dapat mengunduh <i>file</i> proposal revisi mahasiswa dengan dosen pembimbing 2.
103.	<i>GET</i>	/proposal-revisi2/download/:id	Koordinator	Koordinator dapat mengunduh <i>file</i> proposal revisi mahasiswa dengan dosen pembimbing 2.
104.	<i>GET</i>	/proposal-masuk	Admin	Admin dapat menampilkan halaman proposal masuk.
105.	<i>GET</i>	/proposal-masuk	Koordinator	Koordinator dapat menampilkan halaman proposal masuk.
106.	<i>GET</i>	/proposal-masuk/edit/:id	Admin	Admin dapat melakukan perubahan data proposal masuk berdasarkan id.
107.	<i>GET</i>	/proposal-masuk/edit/:id	Koordinator	Koordinator dapat melakukan perubahan data proposal masuk berdasarkan id.
108.	<i>GET</i>	/proposal-terjadwal	Admin	Admin dapat menampilkan halaman proposal terjadwal.
109.	<i>GET</i>	/proposal-terjadwal	Koordinator	Koordinator dapat menampilkan halaman proposal terjadwal.
110.	<i>POST</i>	/proposal-terjadwal/edit/:id	Admin	Admin dapat melakukan perubahan data proposal terjadwal berdasarkan id.
111.	<i>POST</i>	/proposal-terjadwal/edit/:id	Koordinator	Koordinator dapat melakukan perubahan data proposal terjadwal berdasarkan id.
112.	<i>POST</i>	/proposal-terjadwal/selesai/edit/:id	Admin	Admin dapat melakukan perubahan status data proposal terjadwal berdasarkan id.
113.	<i>POST</i>	/proposal-terjadwal/selesai/edit/:id	Koordinator	Koordinator dapat melakukan perubahan status data proposal terjadwal berdasarkan id.
114.	<i>GET</i>	/proposal-selesai	Admin	Admin dapat menampilkan halaman proposal selesai.

115.	<i>GET</i>	/proposal-selesai	Koordinator	Koordinator dapat menampilkan halaman proposal selesai.
116.	<i>POST</i>	/proposal-selesai/edit/:id	Admin	Admin dapat melakukan perubahan data proposal selesai berdasarkan id.
117.	<i>POST</i>	/proposal-selesai/edit/:id	Koordinator	Koordinator dapat melakukan perubahan data proposal selesai berdasarkan id.
118.	<i>GET</i>	/hasil-skripsi	Admin	Admin dapat menampilkan halaman bimbingan hasil mahasiswa.
119.	<i>GET</i>	/bimbingan/hasil-skripsi	Dosen	Dosen dapat menampilkan halaman bimbingan hasil mahasiswa.
120.	<i>GET</i>	/hasil-skripsi	Mahasiswa	Menampilkan halaman bimbingan hasil.
121.	<i>GET</i>	/hasil-skripsi	Koordinator	Koordinator dapat menampilkan halaman bimbingan hasil mahasiswa.
122.	<i>POST</i>	/hasil-skripsi	Mahasiswa	Mahasiswa dapat melakukan bimbingan hasil dengan mengunggah <i>file</i> proposal.
123.	<i>POST</i>	/bimbingan/hasil-skripsi-revisi/tambah/:id	Dosen	Dosen dapat memberikan revisi bimbingan hasil mahasiswa.
124.	<i>GET</i>	/hasil-skripsi/download/:id	Admin	Admin dapat mengunduh <i>file</i> bimbingan hasil mahasiswa.
125.	<i>GET</i>	/bimbingan/hasil-skripsi/download/:id	Dosen	Dosen dapat mengunduh <i>file</i> bimbingan hasil mahasiswa.
126.	<i>GET</i>	/hasil-skripsi/download/:id	Koordinator	Koordinator dapat mengunduh <i>file</i> bimbingan hasil mahasiswa.
127.	<i>GET</i>	/hasil-skripsi/download/:id	Mahasiswa	Mahasiswa dapat mengunduh <i>file</i> hasil yang telah diajukan.
128.	<i>POST</i>	/hasil-skripsi/edit/:id	Admin	Admin dapat melakukan perubahan pada bimbingan hasil mahasiswa.
129.	<i>POST</i>	/hasil-skripsi/edit/:id	Koordinator	Koordinator dapat melakukan perubahan pada bimbingan hasil mahasiswa.
130.	<i>POST</i>	/hasil-skripsi/edit	Mahasiswa	Mahasiswa dapat mengajukan ulang bimbingan hasil dengan mengunggah <i>file</i> hasil skripsi baru.
131.	<i>POST</i>	/hasil-skripsi/status-edit	Mahasiswa	Menampilkan status bimbingan hasil skripsi.
132.	<i>GET</i>	/hasil-skripsi/delete/:id	Admin	Admin dapat menghapus bimbingan hasil mahasiswa.
133.	<i>GET</i>	/hasil-skripsi/delete/:id	Koordinator	Koordinator dapat menghapus bimbingan hasil mahasiswa.
134.	<i>GET</i>	/hasil-skripsi-revisi1/download/:id	Admin	Admin dapat mengunduh <i>file</i> hasil skripsi revisi mahasiswa

				dengan dosen pembimbing 1.
135.	GET	/hasil-skripsi-revisi1/download/:id	Dosen	Dosen dapat mengunduh <i>file</i> hasil skripsi revisi mahasiswa dengan dosen pembimbing 1.
136.	GET	/hasil-skripsi-revisi1/download/:id	Koordinator	Koordinator dapat mengunduh <i>file</i> hasil skripsi revisi mahasiswa dengan dosen pembimbing 1.
137.	POST	/hasil-skripsi-revisi	Mahasiswa	Mahasiswa dapat menampilkan revisi dosen pembimbing 1 dan mengunggah <i>file</i> hasil skripsi revisi.
138.	POST	/hasil-skripsi-revisi2	Mahasiswa	Mahasiswa dapat menampilkan revisi dosen pembimbing 2 dan mengunggah <i>file</i> hasil skripsi revisi.
139.	GET	/hasil-skripsi-revisi2/download/:id	Admin	Admin dapat mengunduh <i>file</i> hasil skripsi revisi mahasiswa dengan dosen pembimbing 2.
140.	GET	/hasil-skripsi-revisi2/download/:id	Dosen	Dosen dapat mengunduh <i>file</i> hasil skripsi revisi mahasiswa dengan dosen pembimbing 2.
141.	GET	/hasil-skripsi-revisi2/download/:id	Koordinator	Koordinator dapat mengunduh <i>file</i> hasil skripsi revisi mahasiswa dengan dosen pembimbing 2.
142.	GET	/hasil-masuk	Admin	Admin dapat menampilkan halaman hasil skripsi masuk.
143.	GET	/hasil-masuk	Koordinator	Koordinator dapat menampilkan halaman hasil skripsi masuk.
144.	GET	/hasil-masuk/edit/:id	Admin	Admin dapat melakukan perubahan data hasil skripsi masuk berdasarkan id.
145.	GET	/hasil-masuk/edit/:id	Koordinator	Koordinator dapat melakukan perubahan data hasil skripsi masuk berdasarkan id.
146.	GET	/hasil-terjadwal	Admin	Admin dapat menampilkan halaman hasil skripsi terjadwal.
147.	GET	/hasil-terjadwal	Koordinator	Admin dapat menampilkan halaman hasil skripsi terjadwal.
148.	POST	/hasil-terjadwal/edit/:id	Admin	Admin dapat melakukan perubahan data hasil skripsi terjadwal berdasarkan id.
149.	POST	/hasil-terjadwal/edit/:id	Koordinator	Koordinator dapat melakukan perubahan data hasil skripsi terjadwal berdasarkan id.
150.	POST	/hasil-terjadwal/selesai/edit/:id	Admin	Admin dapat melakukan perubahan status data hasil skripsi terjadwal berdasarkan id.
151.	POST	/hasil-terjadwal/selesai/edit/	Koordinator	Koordinator dapat melakukan perubahan status

		:id		data hasil skripsi terjadwal berdasarkan id.
152.	GET	/hasil-selesai	Admin	Admin dapat menampilkan halaman hasil skripsi selesai.
153.	GET	/hasil-selesai	Koordinator	Koordinator dapat menampilkan halaman hasil skripsi selesai.
154.	POST	/hasil-selesai/edit/:id	Admin	Admin dapat melakukan perubahan data hasil skripsi selesai berdasarkan id.
155.	POST	/hasil-selesai/edit/:id	Koordinator	Koordinator dapat melakukan perubahan data hasil skripsi selesai berdasarkan id.
156.	GET	/komprehensif	Admin	Admin dapat menampilkan halaman bimbingan komprehensif mahasiswa.
157.	GET	/bimbingan/ komprehensif	Dosen	Dosen dapat menampilkan halaman bimbingan komprehensif mahasiswa.
158.	GET	/komprehensif	Mahasiswa	Menampilkan halaman bimbingan komprehensif.
159.	GET	/komprehensif	Koordinator	Koordinator dapat menampilkan halaman bimbingan komprehensif mahasiswa.
160.	POST	/komprehensif	Mahasiswa	Mahasiswa dapat melakukan bimbingan hasil dengan mengunggah <i>file</i> komprehensif.
161.	GET	/komprehensif /download/:id	Admin	Admin dapat mengunduh <i>file</i> bimbingan komprehensif mahasiswa.
162.	GET	/bimbingan/ komprehensif /download/:id	Dosen	Dosen dapat mengunduh <i>file</i> bimbingan komprehensif mahasiswa.
163.	GET	/komprehensif /download/:id	Mahasiswa	Mahasiswa dapat mengunduh <i>file</i> komprehensif yang telah diajukan.
164.	GET	/komprehensif /download/:id	Koordinator	Koordinator dapat mengunduh <i>file</i> bimbingan komprehensif mahasiswa.
165.	POST	/komprehensif/edit	Mahasiswa	Mahasiswa dapat mengajukan ulang bimbingan komprehensif dengan mengunggah <i>file</i> komprehensif baru.
166.	POST	/komprehensif /edit/:id	Admin	Admin dapat melakukan perubahan pada bimbingan komprehensif mahasiswa.
167.	POST	/komprehensif /edit/:id	Koordinator	Koordinator dapat melakukan perubahan pada bimbingan komprehensif mahasiswa.
168.	POST	/komprehensif/status- edit	Mahcopy- asiswa	Menampilkan status bimbingan komprehensif.
169.	GET	/komprehensif /delete/:id	Admin	Admin dapat menghapus bimbingan komprehensif

				mahasiswa.
170.	GET	/komprehensif /delete/:id	Koordinator	Koordinator dapat menghapus bimbingan komprehensif mahasiswa.
171.	POST	/bimbingan/ komprehensif - revisi/tambah/:id	Dosen	Dosen dapat memberikan revisi bimbingan komprehensif mahasiswa.
172.	GET	/komprehensif- revisi1/download/:id	Admin	Admin dapat mengunduh <i>file</i> komprehensif revisi mahasiswa dengan dosen pembimbing 1.
173.	GET	/komprehensif- revisi1/download/:id	Dosen	Dosen dapat mengunduh <i>file</i> komprehensif revisi mahasiswa dengan dosen pembimbing 1.
174.	GET	/komprehensif- revisi1/download/:id	Koordinator	Koordinator dapat mengunduh <i>file</i> komprehensif revisi mahasiswa dengan dosen pembimbing 1.
175.	POST	/komprehensif-revisi	Mahasiswa	Mahasiswa dapat menampilkan revisi dosen pembimbing 1 dan mengunggah <i>file</i> komprehensif revisi.
176.	GET	/komprehensif- revisi2/download/:id	Admin	Admin dapat mengunduh <i>file</i> komprehensif revisi mahasiswa dengan dosen pembimbing 2.
177.	GET	/komprehensif- revisi2/download/:id	Dosen	Dosen dapat mengunduh <i>file</i> komprehensif revisi mahasiswa dengan dosen pembimbing 2.
178.	GET	/komprehensif- revisi2/download/:id	Koordinator	Koordinator dapat mengunduh <i>file</i> komprehensif revisi mahasiswa dengan dosen pembimbing 2.
179.	POST	/komprehensif-revisi2	Mahasiswa	Mahasiswa dapat menampilkan revisi dosen pembimbing 2 dan mengunggah <i>file</i> komprehensif revisi.
180.	GET	/komprehensif-masuk	Admin	Admin dapat menampilkan halaman komprehensif masuk.
181.	GET	/komprehensif-masuk	Koordinator	Koordinator dapat menampilkan halaman komprehensif masuk.
182.	GET	/komprehensif - masuk/edit/:id	Admin	Admin dapat melakukan perubahan data komprehensif masuk berdasarkan id.
183.	GET	/komprehensif - masuk/edit/:id	Koordinator	Koordinator dapat melakukan perubahan data komprehensif masuk berdasarkan id.
184.	GET	/komprehensif- terjadwal	Admin	Admin dapat menampilkan halaman komprehensif terjadwal.
185.	GET	/komprehensif- terjadwal	Koordinator	Koordinator dapat menampilkan halaman komprehensif terjadwal.
186.	POST	/komprehensif-	Admin	Admin dapat melakukan

		terjadwal/ <i>edit</i> :id		perubahan data komprehensif terjadwal berdasarkan id.
187.	<i>POST</i>	/komprehensif-terjadwal/ <i>edit</i> :id	Koordinator	Koordinator dapat melakukan perubahan data komprehensif terjadwal berdasarkan id.
188.	<i>POST</i>	/komprehensif -terjadwal/selesai/ <i>edit</i> :id	Admin	Admin dapat melakukan perubahan status data komprehensif terjadwal berdasarkan id.
189.	<i>POST</i>	/komprehensif -terjadwal/selesai/ <i>edit</i> :id	Koordinator	Koordinator dapat melakukan perubahan status data komprehensif terjadwal berdasarkan id.
190.	<i>GET</i>	/komprehensif-selesai	Admin	Admin dapat menampilkan halaman komprehensif selesai.
191.	<i>GET</i>	/komprehensif-selesai	Koordinator	Koordinator dapat menampilkan halaman komprehensif selesai.
192.	<i>POST</i>	/komprehensif-selesai/ <i>edit</i> :id	Admin	Admin dapat melakukan perubahan data komprehensif selesai berdasarkan id.
193.	<i>POST</i>	/komprehensif-selesai/ <i>edit</i> :id	Koordinator	Koordinator dapat melakukan perubahan data komprehensif selesai berdasarkan id.
194.	<i>GET</i>	/jadwal/proposal	Dosen	Dosen dapat menampilkan halaman jadwal seminar proposal mahasiswa.
195.	<i>GET</i>	/jadwal/hasil-skripsi	Dosen	Dosen dapat menampilkan halaman jadwal seminar hasil mahasiswa.
196.	<i>GET</i>	/jadwal/komprehensif	Dosen	Dosen dapat menampilkan halaman jadwal ujian komprehensif mahasiswa.
197.	<i>GET</i>	/riwayat/proposal	Dosen	Dosen dapat menampilkan halaman riwayat seminar proposal mahasiswa.
198.	<i>GET</i>	/riwayat/hasil-skripsi	Dosen	Dosen dapat menampilkan halaman riwayat seminar hasil mahasiswa.
199.	<i>GET</i>	/riwayat/ Komprehensif	Dosen	Dosen dapat menampilkan halaman riwayat ujian komprehensif mahasiswa.
200.	<i>POST</i>	/copy-mahasiswa	Admin	Admin memiliki kemampuan untuk menyalin data mahasiswa ke papan klip.
201.	<i>POST</i>	/copy	Admin	Admin memiliki kemampuan untuk menyalin data dosen ke papan klip.
202.	<i>POST</i>	/copy-koor	Admin	Admin memiliki kemampuan untuk menyalin data koordinator ke papan klip.
203.	<i>POST</i>	/copy-pengajuanJudul	Admin	Admin memiliki kemampuan untuk menyalin data pengajuan

				judul ke papan klip.
204.	POST	/copy-pengajuanJudul	Dosen	Dosen memiliki kemampuan untuk menyalin data pengajuan judul ke papan klip.
205.	POST	/copy-pengajuanJudul	Koordinator	Koordinator memiliki kemampuan untuk menyalin data pengajuan judul ke papan klip.
206.	POST	/copy-judulDiterima	Admin	Admin memiliki kemampuan untuk menyalin data judul diterima ke papan klip.
207.	POST	/copy-judulDiterima	Dosen	Dosen memiliki kemampuan untuk menyalin data judul diterima ke papan klip.
208.	POST	/copy-judulDiterima	Koordinator	Koordinator memiliki kemampuan untuk menyalin data judul diterima ke papan klip.
209.	POST	/copy-pembagianDosen	Admin	Admin memiliki kemampuan untuk menyalin data pembagian dosen ke papan klip.
210.	POST	/copy-pembagianDosen	Koordinator	Koordinator memiliki kemampuan untuk menyalin data pembagian dosen ke papan klip.
211.	POST	/copy-permintaan Bimbingan	Admin	Admin memiliki kemampuan untuk menyalin data permintaan bimbingan ke papan klip.
212.	POST	/copy-permintaan Bimbingan	Koordinator	Koordinator memiliki kemampuan untuk menyalin data permintaan bimbingan ke papan klip.
213.	POST	/copy-daftarDosen1	Admin	Admin memiliki kemampuan untuk menyalin data daftar dosen pembimbing 1 ke papan klip.
214.	POST	/copy-daftarDosen1	Koordinator	Koordinator memiliki kemampuan untuk menyalin data daftar dosen pembimbing 1 ke papan klip.
215.	POST	/copy-daftarDosen2	Admin	Admin memiliki kemampuan untuk menyalin data daftar dosen pembimbing 2 ke papan klip.
216.	POST	/copy-daftarDosen2	Koordinator	Koordinator memiliki kemampuan untuk menyalin data daftar dosen pembimbing 2 ke papan klip.
217.	POST	/copy-daftarDosen3	Admin	Admin memiliki kemampuan untuk menyalin data daftar dosen pembahas ke papan klip.
218.	POST	/copy-daftarDosen3	Koordinator	Koordinator memiliki

				kemampuan untuk menyalin data daftar dosen pembahas ke papan klip.
219.	POST	/copy-proposal	Admin	Admin memiliki kemampuan untuk menyalin data bimbingan proposal ke papan klip.
220.	POST	/copy-proposal	Koordinator	Koordinator memiliki kemampuan untuk menyalin data bimbingan proposal ke papan klip.
221.	POST	/copy-proposalMasuk	Admin	Admin memiliki kemampuan untuk menyalin data proposal masuk ke papan klip.
222.	POST	/copy-proposalMasuk	Koordinator	Koordinator memiliki kemampuan untuk menyalin data proposal masuk ke papan klip.
223.	POST	/copy-proposalTerjadwal	Admin	Admin memiliki kemampuan untuk menyalin data proposal terjadwal ke papan klip.
224.	POST	/copy-proposalTerjadwal	Koordinator	Admin memiliki kemampuan untuk menyalin data proposal terjadwal ke papan klip.
225.	POST	/copy-proposalSelesai	Admin	Admin memiliki kemampuan untuk menyalin data proposal selesai ke papan klip.
226.	POST	/copy-proposalSelesai	Koordinator	Koordinator memiliki kemampuan untuk menyalin data proposal selesai ke papan klip.
227.	POST	/copy-hasil	Admin	Admin memiliki kemampuan untuk menyalin data bimbingan hasil ke papan klip.
228.	POST	/copy-hasil	Koordinator	Koordinator memiliki kemampuan untuk menyalin data bimbingan hasil ke papan klip.
229.	POST	/copy-hasilMasuk	Admin	Admin memiliki kemampuan untuk menyalin data hasil skripsi masuk ke papan klip.
230.	POST	/copy-hasilMasuk	Koordinator	Koordinator memiliki kemampuan untuk menyalin data hasil skripsi masuk ke papan klip.
231.	POST	/copy-hasilTerjadwal	Admin	Admin memiliki kemampuan untuk menyalin data hasil skripsi terjadwal ke papan klip.
232.	POST	/copy-hasilTerjadwal	Koordinator	Koordinator memiliki kemampuan untuk menyalin data hasil skripsi terjadwal ke papan klip.
233.	POST	/copy-hasilSelesai	Admin	Admin memiliki kemampuan

				untuk menyalin data hasil skripsi selesai ke papan klip.
234.	POST	/copy-hasilSelesai	Koordinator	Koordinator memiliki kemampuan untuk menyalin data hasil skripsi selesai ke papan klip.
235.	POST	/copy-komprehensif	Admin	Admin memiliki kemampuan untuk menyalin data bimbingan komprehensif ke papan klip.
236.	POST	/copy-komprehensif	Koordinator	Koordinator memiliki kemampuan untuk menyalin data bimbingan komprehensif ke papan klip.
237.	POST	/copy-komprehensif Masuk	Admin	Admin memiliki kemampuan untuk menyalin data komprehensif masuk ke papan klip.
238.	POST	/copy-komprehensif Masuk	Koordinator	Koordinator memiliki kemampuan untuk menyalin data komprehensif masuk ke papan klip.
239.	POST	/copy-komprehensif Terjadwal	Admin	Admin memiliki kemampuan untuk menyalin data komprehensif terjadwal ke papan klip.
240.	POST	/copy-komprehensif Terjadwal	Koordinator	Koordinator memiliki kemampuan untuk menyalin data komprehensif terjadwal ke papan klip.
241.	POST	/copy-komprehensif Selesai	Admin	Admin memiliki kemampuan untuk menyalin data komprehensif selesai ke papan klip.
242.	POST	/copy-komprehensif Selesai	Koordinator	Koordinator memiliki kemampuan untuk menyalin data komprehensif selesai ke papan klip.
243.	POST	/csv-mahasiswa	Admin	Admin dapat mengekspor data mahasiswa ke dalam format CSV.
244.	POST	/csv	Admin	Admin dapat mengekspor data dosen ke dalam format CSV.
245.	POST	/csv-koor	Admin	Admin dapat mengekspor data koordinator ke dalam format CSV.
246.	POST	/csv-pengajuanJudul	Admin	Admin dapat mengekspor data pengajuan judul ke dalam format CSV.
247.	POST	/csv-pengajuanJudul	Dosen	Dosen dapat mengekspor data pengajuan judul ke dalam format CSV.
248.	POST	/csv-pengajuanJudul	Koordinator	Koordinator dapat mengekspor

				data pengajuan judul ke dalam format CSV.
249.	POST	/csv-judulDiterima	Admin	Admin dapat mengekspor data judul diterima ke dalam format CSV.
250.	POST	/csv-judulDiterima	Dosen	Dosen dapat mengekspor data judul diterima ke dalam format CSV.
251.	POST	/csv-judulDiterima	Koordinator	Koordinator dapat mengekspor data judul diterima ke dalam format CSV.
252.	POST	/csv-pembagianDosen	Admin	Admin dapat mengekspor data pembagian dosen ke dalam format CSV.
253.	POST	/csv-pembagianDosen	Koordinator	Koordinator dapat mengekspor data pembagian dosen ke dalam format CSV.
254.	POST	/csv- permintaan Bimbingan	Admin	Admin dapat mengekspor data permintaan bimbingan ke dalam format CSV.
255.	POST	/csv- permintaan Bimbingan	Koordinator	Koordinator dapat mengekspor data permintaan bimbingan ke dalam format CSV.
256.	POST	/csv-daftarDosen1	Admin	Admin dapat mengekspor data daftar dosen pembimbing 1 ke dalam format CSV.
257.	POST	/csv-daftarDosen1	Koordinator	Koordinator dapat mengekspor data daftar dosen pembimbing 1 ke dalam format CSV.
258.	POST	/csv-daftarDosen2	Admin	Admin dapat mengekspor data daftar dosen pembimbing 2 ke dalam format CSV.
259.	POST	/csv-daftarDosen2	Koordinator	Koordinator dapat mengekspor data daftar dosen pembimbing 2 ke dalam format CSV.
260.	POST	/csv-daftarDosen3	Admin	Admin dapat mengekspor data daftar dosen pembahas ke dalam format CSV.
261.	POST	/csv-daftarDosen3	Koordinator	Koordinator dapat mengekspor data daftar dosen pembahas ke dalam format CSV.
262.	POST	/csv-proposal	Admin	Admin dapat mengekspor data bimbingan proposal ke dalam format CSV.
263.	POST	/csv-proposal	Koordinator	Koordinator dapat mengekspor data bimbingan proposal ke dalam format CSV.
264.	POST	/csv-proposalMasuk	Admin	Admin dapat mengekspor data proposal masuk ke dalam format CSV.
265.	POST	/csv-proposalMasuk	Koordinator	Koordinator dapat mengekspor data proposal masuk ke dalam

				format CSV.
266.	POST	/csv-proposalTerjadwal	Admin	Admin dapat mengekspor data proposal terjadwal ke dalam format CSV.
267.	POST	/csv-proposalTerjadwal	Koordinator	Koordinator dapat mengekspor data proposal terjadwal ke dalam format CSV.
268.	POST	/csv-proposalSelesai	Admin	Admin dapat mengekspor data proposal selesai ke dalam format CSV.
269.	POST	/csv-proposalSelesai	Koordinat or	Koordinator dapat mengekspor data proposal selesai ke dalam format CSV.
270.	POST	/csv-hasil	Admin	Admin dapat mengekspor data bimbingan hasil ke dalam format CSV.
271.	POST	/csv-hasil	Koordinator	Koordinator dapat mengekspor data bimbingan hasil ke dalam format CSV.
272.	POST	/csv-hasilMasuk	Admin	Admin dapat mengekspor data hasil skripsi masuk ke dalam format CSV.
273.	POST	/csv-hasilMasuk	Koordinator	Koordinator dapat mengekspor data hasil skripsi masuk ke dalam format CSV.
274.	POST	/csv-hasilTerjadwal	Admin	Admin dapat mengekspor data hasil skripsi terjadwal ke dalam format CSV.
275.	POST	/csv-hasilTerjadwal	Koordinator	Koordinator dapat mengekspor data hasil skripsi terjadwal ke dalam format CSV.
276.	POST	/csv-hasilSelesai	Admin	Admin dapat mengekspor data hasil skripsi selesai ke dalam format CSV.
277.	POST	/csv-hasilSelesai	Koordinator	Koordinator dapat mengekspor data hasil skripsi selesai ke dalam format CSV.
278.	POST	/csv-komprehensif	Admin	Admin dapat mengekspor data bimbingan komprehensif ke dalam format CSV.
279.	POST	/csv-komprehensif	Koordinator	Koordinator dapat mengekspor data bimbingan komprehensif ke dalam format CSV.
280.	POST	/csv-komprehensif Masuk	Admin	Admin dapat mengekspor data komprehensif masuk ke dalam format CSV.
281.	POST	/csv-komprehensif Masuk	Koordinator	Koordinator dapat mengekspor data komprehensif masuk ke dalam format CSV.
282.	POST	/csv-komprehensif Terjadwal	Admin	Admin dapat mengekspor data komprehensif terjadwal ke dalam format CSV.

283.	POST	/csv-komprehensif Terjadwal	Koordinator	Koordinator dapat mengekspor data komprehensif terjadwal ke dalam format CSV.
284.	POST	/csv-komprehensif Selesai	Admin	Admin dapat mengekspor data komprehensif selesai ke dalam format CSV.
285.	POST	/csv-komprehensif Selesai	Koordinator	Koordinator dapat mengekspor data komprehensif selesai ke dalam format CSV.
286.	POST	/excel-mahasiswa	Admin	Admin dapat mengekspor data mahasiswa ke dalam format Excel.
287.	POST	/excel	Admin	Admin dapat mengekspor data dosen ke dalam format Excel.
288.	POST	/excel-koor	Admin	Admin dapat mengekspor data koordinator ke dalam format Excel.
289.	POST	/excel-pengajuanJudul	Admin	Admin dapat mengekspor data pengajuan judul ke dalam format Excel.
290.	POST	/excel-pengajuanJudul Dosen	Dosen	Dosen dapat mengekspor data pengajuan judul ke dalam format Excel.
291.	POST	/excel-pengajuanJudul	Koordinator	Koordinator dapat mengekspor data pengajuan judul ke dalam format Excel.
292.	POST	/excel-judulDiterima	Admin	Admin dapat mengekspor data judul diterima ke dalam format Excel.
293.	POST	/excel-judulDiterima	Dosen	Dosen dapat mengekspor data judul diterima ke dalam format Excel.
294.	POST	/excel-judulDiterima	Koordinator	Koordinator dapat mengekspor data judul diterima ke dalam format Excel.
295.	POST	/excel- pembagianDosen	Admin	Admin dapat mengekspor data pembagian dosen ke dalam format Excel.
296.	POST	/excel- pembagianDosen	Koordinator	Koordinator dapat mengekspor data pembagian dosen ke dalam format Excel.
297.	POST	/excel- permintaan Bimbingan	Admin	Admin dapat mengekspor data permintaan bimbingan ke dalam format Excel.
298.	POST	/excel- permintaan Bimbingan	Koordinator	Koordinator dapat mengekspor data permintaan bimbingan ke dalam format Excel.
299.	POST	/excel-daftarDosen1	Admin	Admin dapat mengekspor data daftar dosen pembimbing 1 ke dalam format Excel.
300.	POST	/excel-daftarDosen1	Koordinator	Koordinator dapat mengekspor data daftar dosen pembimbing 1

				ke dalam format Excel.
301.	POST	/excel-daftarDosen2	Admin	Admin dapat mengekspor data daftar dosen pembimbing 2 ke dalam format Excel.
302.	POST	/excel-daftarDosen2	Koordinator	Koordinator dapat mengekspor data daftar dosen pembimbing 2 ke dalam format Excel.
303.	POST	/excel-daftarDosen3	Admin	Admin dapat mengekspor data daftar dosen pembahas ke dalam format Excel.
304.	POST	/excel-daftarDosen3	Koordinator	Koordinator dapat mengekspor data daftar dosen pembahas ke dalam format Excel.
305.	POST	/excel-proposal	Admin	Admin dapat mengekspor data bimbingan proposal ke dalam format Excel.
306.	POST	/excel-proposal	Koordinator	Koordinator dapat mengekspor data bimbingan proposal ke dalam format Excel.
307.	POST	/excel-proposalMasuk	Admin	Admin dapat mengekspor data proposal masuk ke dalam format Excel.
308.	POST	/excel-proposalMasuk	Koordinator	Koordinator dapat mengekspor data proposal masuk ke dalam format Excel.
309.	POST	/excel-proposalTerjadwal	Admin	Admin dapat mengekspor data proposal terjadwal ke dalam format Excel.
310.	POST	/excel-proposalTerjadwal	Koordinator	Koordinator dapat mengekspor data proposal terjadwal ke dalam format Excel.
311.	POST	/excel-proposalSelesai	Admin	Admin dapat mengekspor data proposal selesai ke dalam format Excel.
312.	POST	/excel-proposalSelesai	Koordinator	Koordinator dapat mengekspor data proposal selesai ke dalam format Excel.
313.	POST	/excel-hasil	Admin	Admin dapat mengekspor data bimbingan hasil ke dalam format Excel.
314.	POST	/excel-hasil	Koordinator	Koordinator dapat mengekspor data bimbingan hasil ke dalam format Excel.
315.	POST	/excel-hasilMasuk	Admin	Admin dapat mengekspor data hasil skripsi masuk ke dalam format Excel.
316.	POST	/excel-hasilMasuk	Koordinator	Koordinator dapat mengekspor data hasil skripsi masuk ke dalam format Excel.
317.	POST	/excel-hasilTerjadwal	Admin	Admin dapat mengekspor data hasil skripsi terjadwal ke dalam format Excel.

318.	POST	/excel-hasilTerjadwal	Koordinator	Koordinator dapat mengekspor data hasil skripsi terjadwal ke dalam format Excel.
319.	POST	/excel-hasilSelesai	Admin	Admin dapat mengekspor data hasil skripsi selesai ke dalam format Excel.
320.	POST	/excel-hasilSelesai	Koordinator	Koordinator dapat mengekspor data hasil skripsi selesai ke dalam format Excel.
321.	POST	/excel-komprehensif	Admin	Admin dapat mengekspor data bimbingan komprehensif ke dalam format Excel.
322.	POST	/excel-komprehensif	Koordinator	Koordinator dapat mengekspor data bimbingan komprehensif ke dalam format Excel.
323.	POST	/excel-komprehensif Masuk	Admin	Admin dapat mengekspor data komprehensif masuk ke dalam format Excel.
324.	POST	/excel-komprehensif Masuk	Koordinator	Koordinator dapat mengekspor data komprehensif masuk ke dalam format Excel.
325.	POST	/excel-komprehensif Terjadwal	Admin	Admin dapat mengekspor data komprehensif terjadwal ke dalam format Excel.
326.	POST	/excel-komprehensif Terjadwal	Koordinator	Koordinator dapat mengekspor data komprehensif terjadwal ke dalam format Excel.
327.	POST	/excel-komprehensif Selesai	Admin	Admin dapat mengekspor data komprehensif selesai ke dalam format Excel.
328.	POST	/excel-komprehensif Selesai	Koordinator	Koordinator dapat mengekspor data komprehensif selesai ke dalam format Excel.
329.	POST	/pdf-mahasiswa	Admin	Admin dapat mengekspor data mahasiswa ke dalam format PDF.
330.	POST	/pdf	Admin	Admin dapat mengekspor data dosen ke dalam format PDF.
331.	POST	/pdf-koor	Admin	Admin dapat mengekspor data koordinator ke dalam format PDF.
332.	POST	/pdf-pengajuanJudul	Admin	Admin dapat mengekspor data pengajuan judul ke dalam format PDF.
333.	POST	/pdf-pengajuanJudul	Dosen	Dosen dapat mengekspor data pengajuan judul ke dalam format PDF.
334.	POST	/pdf-pengajuanJudul	Koordinator	Koordinator dapat mengekspor data pengajuan judul ke dalam format PDF.
335.	POST	/pdf-judulDiterima	Admin	Admin dapat mengekspor data judul diterima ke dalam format

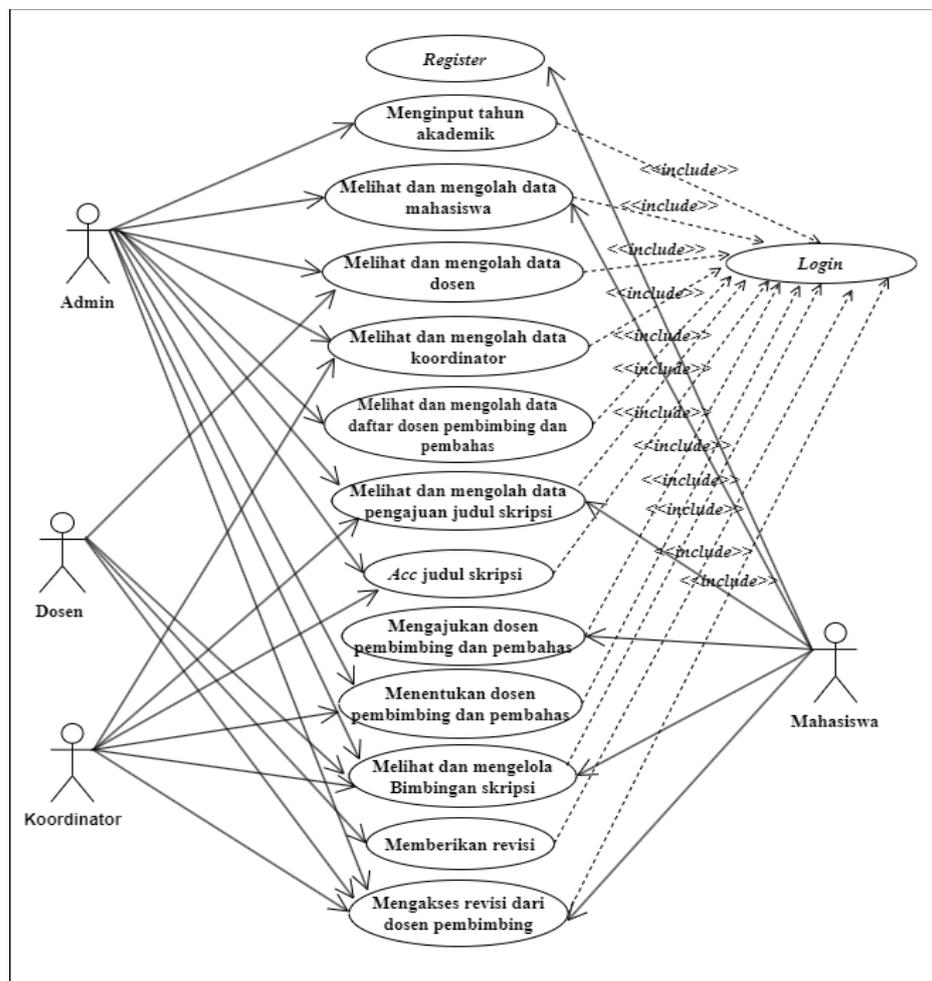
				PDF.
336.	POST	/pdf-judulDiterima	Dosen	Dosen dapat mengekspor data judul diterima ke dalam format PDF.
337.	POST	/pdf-judulDiterima	Koordinator	Koordinator dapat mengekspor data judul diterima ke dalam format PDF.
338.	POST	/pdf-pembagianDosen	Admin	Admin dapat mengekspor data pembagian dosen ke dalam format PDF.
339.	POST	/pdf-pembagianDosen	Koordinator	Koordinator dapat mengekspor data pembagian dosen ke dalam format PDF.
340.	POST	/pdf- permintaan Bimbingan	Admin	Admin dapat mengekspor data permintaan bimbingan ke dalam format PDF.
341.	POST	/pdf- permintaan Bimbingan	Koordinator	Koordinator dapat mengekspor data permintaan bimbingan ke dalam format PDF.
342.	POST	/pdf-daftarDosen1	Admin	Admin dapat mengekspor data daftar dosen pembimbing 1 ke dalam format PDF.
343.	POST	/pdf-daftarDosen1	Koordinator	Koordinator dapat mengekspor data daftar dosen pembimbing 1 ke dalam format PDF.
344.	POST	/pdf-daftarDosen2	Admin	Admin dapat mengekspor data daftar dosen pembimbing 2 ke dalam format PDF.
345.	POST	/pdf-daftarDosen2	Koordinator	Koordinator dapat mengekspor data daftar dosen pembimbing 2 ke dalam format PDF.
346.	POST	/pdf-daftarDosen3	Admin	Admin dapat mengekspor data daftar dosen pembahas ke dalam format PDF.
347.	POST	/pdf-daftarDosen3	Koordinator	Koordinator dapat mengekspor data daftar dosen pembahas ke dalam format PDF.
348.	POST	/pdf-proposal	Admin	Admin dapat mengekspor data bimbingan proposal ke dalam format PDF.
349.	POST	/pdf-proposal	Koordinator	Koordinator dapat mengekspor data bimbingan proposal ke dalam format PDF.
350.	POST	/pdf-proposalMasuk	Admin	Admin dapat mengekspor data proposal masuk ke dalam format PDF.
351.	POST	/pdf-proposalMasuk	Koordinator	Koordinator dapat mengekspor data proposal masuk ke dalam format PDF.
352.	POST	/pdf- proposalTerjadwal	Admin	Admin dapat mengekspor data proposal terjadwal ke dalam format PDF.

353.	POST	/pdf-proposalTerjadwal	Koordinator	Koordinator dapat mengekspor data proposal terjadwal ke dalam format PDF.
354.	POST	/pdf-proposalSelesai	Admin	Admin dapat mengekspor data proposal selesai ke dalam format PDF.
355.	POST	/pdf-proposalSelesai	Koordinator	Koordinator dapat mengekspor data proposal selesai ke dalam format PDF.
356.	POST	/pdf-hasil	Admin	Admin dapat mengekspor data bimbingan hasil ke dalam format PDF.
357.	POST	/pdf-hasil	Koordinator	Koordinator dapat mengekspor data bimbingan hasil ke dalam format PDF.
358.	POST	/pdf-hasilMasuk	Admin	Admin dapat mengekspor data hasil skripsi masuk ke dalam format PDF.
359.	POST	/pdf-hasilMasuk	Koordinator	Koordinator dapat mengekspor data hasil skripsi masuk ke dalam format PDF.
360.	POST	/pdf-hasilTerjadwal	Admin	Admin dapat mengekspor data hasil skripsi terjadwal ke dalam format PDF.
361.	POST	/pdf-hasilTerjadwal	Koordinator	Koordinator dapat mengekspor data hasil skripsi terjadwal ke dalam format PDF.
362.	POST	/pdf-hasilSelesai	Admin	Admin dapat mengekspor data hasil skripsi selesai ke dalam format PDF.
363.	POST	/pdf-hasilSelesai	Koordinator	Koordinator dapat mengekspor data hasil skripsi selesai ke dalam format PDF.
364.	POST	/pdf-komprehensif	Admin	Admin dapat mengekspor data bimbingan komprehensif ke dalam format PDF.
365.	POST	/pdf-komprehensif	Koordinator	Koordinator dapat mengekspor data bimbingan komprehensif ke dalam format PDF.
366.	POST	/pdf-komprehensif Masuk	Admin	Admin dapat mengekspor data komprehensif masuk ke dalam format PDF.
367.	POST	/pdf-komprehensif Masuk	Koordinator	Koordinator dapat mengekspor data komprehensif masuk ke dalam format PDF.
368.	POST	/pdf-komprehensif Terjadwal	Admin	Admin dapat mengekspor data komprehensif terjadwal ke dalam format PDF.
369.	POST	/pdf-komprehensif Terjadwal	Koordinator	Koordinator dapat mengekspor data komprehensif terjadwal ke dalam format PDF.
370.	POST	/pdf-komprehensif	Admin	Admin dapat mengekspor data

		Selesai		hasil skripsi selesai ke dalam format PDF.
371.	POST	/pdf-komprehensif Selesai	Koordinator	Koordinator dapat mengekspor data komprehensif selesai ke dalam format PDF.
372.	GET	/:id/profile	Admin	Admin dapat menampilkan halaman <i>profile</i> .
373.	GET	/:id/profile	Dosen	Dosen dapat menampilkan halaman <i>profile</i> .
374.	GET	/:id/profile	Mahasiswa	Menampilkan halaman <i>profile</i> .
375.	GET	/:id/profile	Koordinator	Koordinator dapat menampilkan halaman <i>profile</i> .
376.	POST	/:id/profile	Admin	Admin dapat melakukan perubahan data <i>profile</i> .
377.	POST	/:id/profile	Dosen	Dosen dapat melakukan perubahan data <i>profile</i> .
378.	POST	/:id/profile	Mahasiswa	Mahasiswa dapat melakukan perubahan data <i>profile</i> .
379.	POST	/:id/profile	Koordinator	Admin dapat melakukan perubahan data <i>profile</i> .
380.	GET	/:id/ubah-password	Admin	Admin dapat menampilkan halaman ubah <i>password</i> .
381.	GET	/:id/ubah-password	Dosen	Dosen dapat menampilkan halaman ubah <i>password</i> .
382.	GET	/:id/ubah-password	Mahasiswa	Menampilkan halaman ubah <i>password</i> .
383.	GET	/:id/ubah-password	Koordinator	Koordinator dapat menampilkan halaman ubah <i>password</i> .
384.	POST	/:id/ubah-password	Admin	Admin dapat melakukan perubahan data ubah <i>password</i> .
385.	POST	/:id/ubah-password	Dosen	Dosen dapat melakukan perubahan data ubah <i>password</i> .
386.	POST	/:id/ubah-password	Mahasiswa	Mahasiswa dapat melakukan perubahan data ubah <i>password</i> .
387.	POST	/:id/ubah-password	Koordinator	Koordinator dapat melakukan perubahan data ubah <i>password</i> .
388.	GET	/forgot	Mahasiswa	Menampilkan halaman lupa <i>password</i> .
389.	GET	/forgot	Mahasiswa	Mengirimkan link reset <i>password</i> ke <i>email</i> .
390.	GET	/reset/:token	Mahasiswa	Menampilkan halaman <i>reset password</i> .
391.	GET	/reset/:token	Mahasiswa	Mengatur <i>password</i> baru.
392.	GET	/logout	Admin	Admin dapat <i>logout</i> dari sistem.
393.	GET	/logout	Dosen	Dosen dapat <i>logout</i> dari aplikasi.
394.	GET	/logout	Mahasiswa	Mahasiswa dapat <i>logout</i> .
395.	GET	/logout	Koordinator	Koordinator dapat <i>logout</i> dari sistem.

3.2.5.4 Design Use Case Diagram

Diagram *Use Case* menggambarkan interaksi antara *user* dengan aplikasi itu sendiri. Tahap *design use case diagram* ini menunjukkan aktivitas yang dilakukan oleh *user* terhadap aplikasi. Aplikasi yang dibangun melibatkan empat aktor yang memiliki akses langsung ke sistem, yaitu admin, koordinator, dosen, dan mahasiswa. Setiap aktor memiliki hak dan kekuasaan sendiri dalam mengelola sistem tersebut.



Gambar 5. Use Case Diagram

Gambar 5 menyajikan *use case diagram* yang digunakan untuk menggambarkan hubungan antara pengguna dengan aplikasi yang telah dirancang. *Rules* yang berlaku pada sistem ini yaitu setiap aktor yang mengakses aplikasi harus melakukan *login* terlebih dahulu, jika tidak, maka secara otomatis sistem mengarahkan pada halaman *login*.

3.2.5.5 Design Activity Diagram

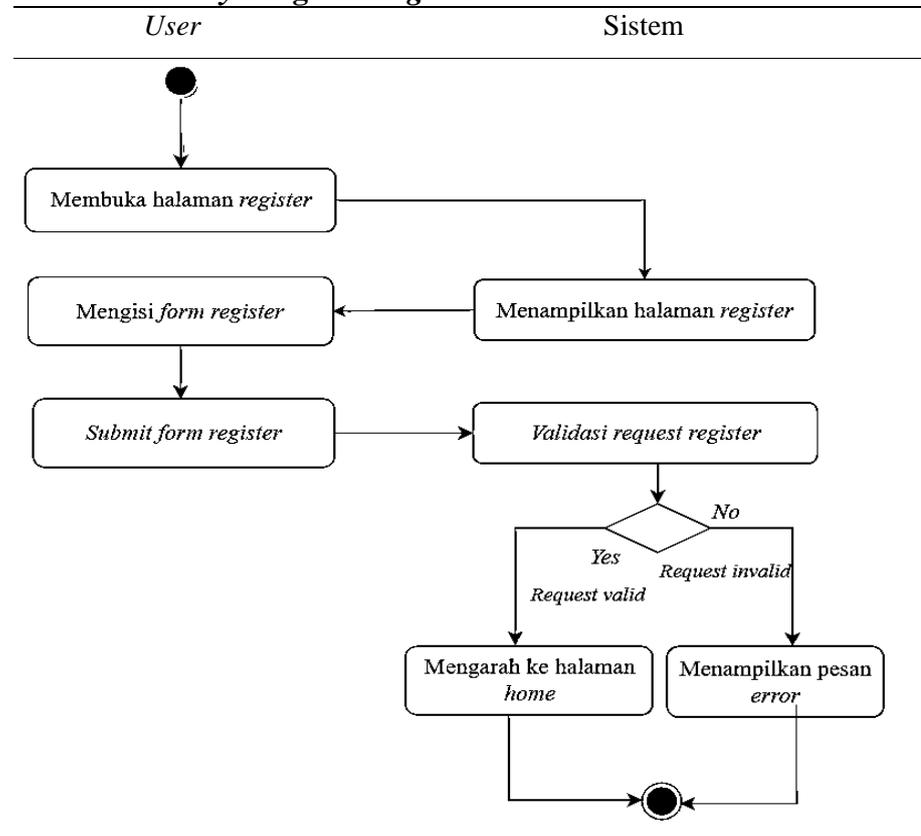
Setelah merancang diagram *use case*, langkah selanjutnya adalah merancang diagram aktivitas (*activity diagram*) berdasarkan daftar diagram *use case* yang telah dirancang sebelumnya. *Design activity diagram* merupakan sebuah *workflow* atau aktivitas pada sistem yang dijalankan, *activity diagram* juga digunakan untuk mendefinisikan tahapan pada setiap alur sistem.

Design activity diagram:

1. Activity Diagram Register

Register merupakan proses pendaftaran yang wajib dilakukan oleh *user* untuk membuat akun pada sistem. Diagram aktivitas ini menjelaskan alur kerja yang dilakukan oleh pengguna dalam melakukan registrasi akun. *Activity diagram register* ditunjukkan pada Tabel 9.

Tabel 9. Activity Diagram Register

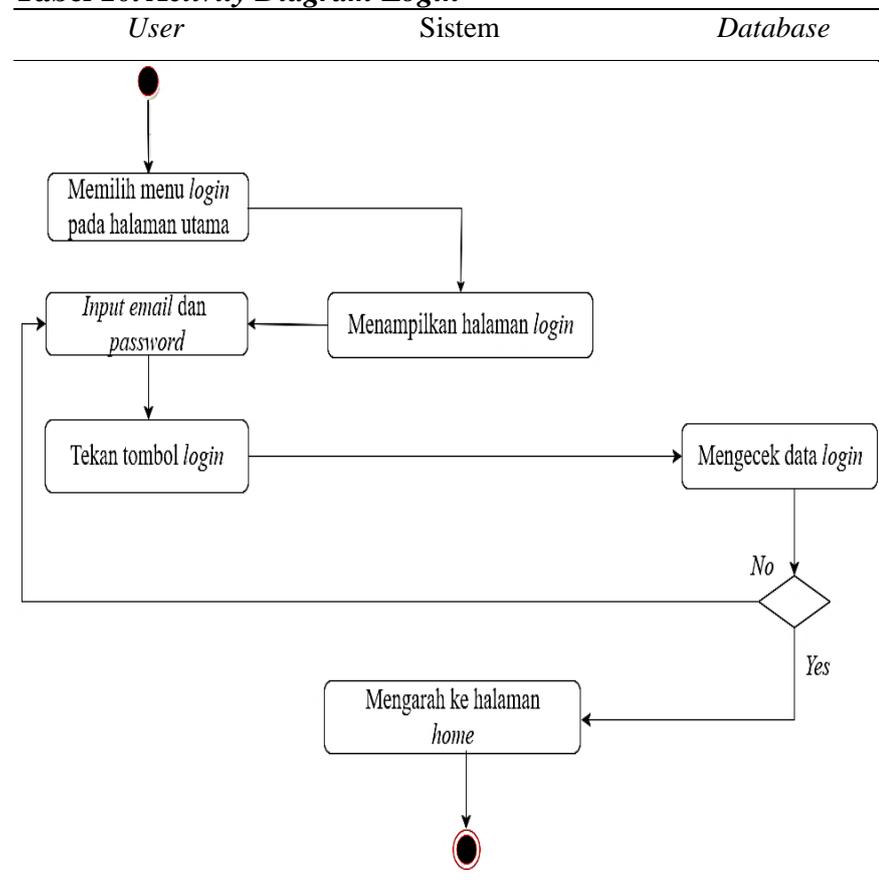


Tabel 9 menjelaskan bagaimana mahasiswa melakukan registrasi ke dalam sistem. Jika mahasiswa belum memiliki akun, maka sistem menampilkan halaman *register*. Proses registrasi, mahasiswa mengisi *form register* yang tersedia pada halaman *register*, kemudian tekan tombol *register*, maka terdapat verifikasi yang dikirimkan melalui *email* yang telah didaftarkan. Jika telah diverifikasi pada *email*, maka *user* sudah terdaftar dan dapat melanjutkan ke halaman *login*.

2. Activity Diagram Login

Login merupakan langkah yang dilakukan oleh admin dan *user* sebagai pemilik akun untuk mengakses sistem. *Activity diagram login* ini, menjelaskan alur proses *login*. Sebelum melakukan *login*, *user* harus melakukan pendaftaran terlebih dahulu untuk dapat memiliki akun.

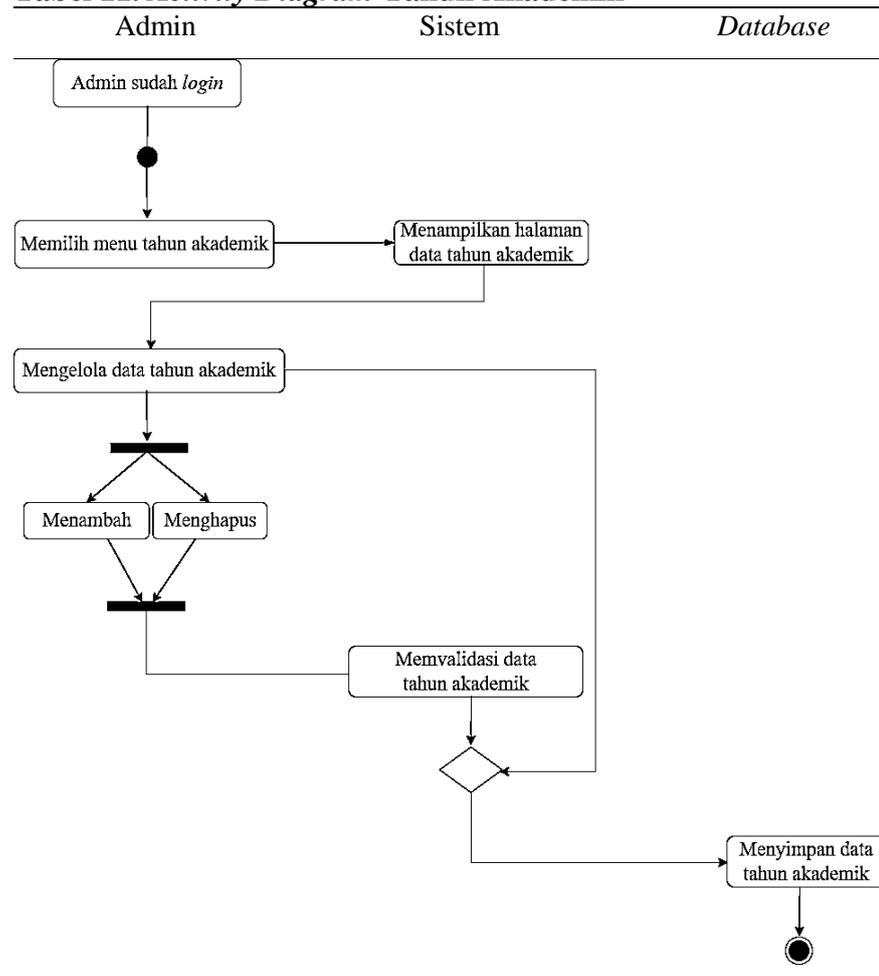
Tabel 10. Activity Diagram Login



Tabel 10 menyajikan aktivitas *login* yang dilakukan oleh pengguna sebagai bagian dari jalannya program. Pengguna harus mengakses halaman *login* terlebih dahulu. Setelah mengakses halaman *login*, sistem menampilkan formulir *login* yang dapat diakses oleh pengguna. Kemudian *user* memasukkan *email* dan *password*, ketika *user* sudah memasukkan *email* dan *password* maka sistem memvalidasi data yang dimasukkan, apabila data sesuai dengan ketentuan validasi, maka *login* dinyatakan berhasil dan *user* diarahkan ke *dashboard* sesuai perannya dalam sistem. Jika gagal maka *user* diminta untuk melakukan pengisian *email* dan *password* kembali.

3. Activity Diagram Tahun Akademik

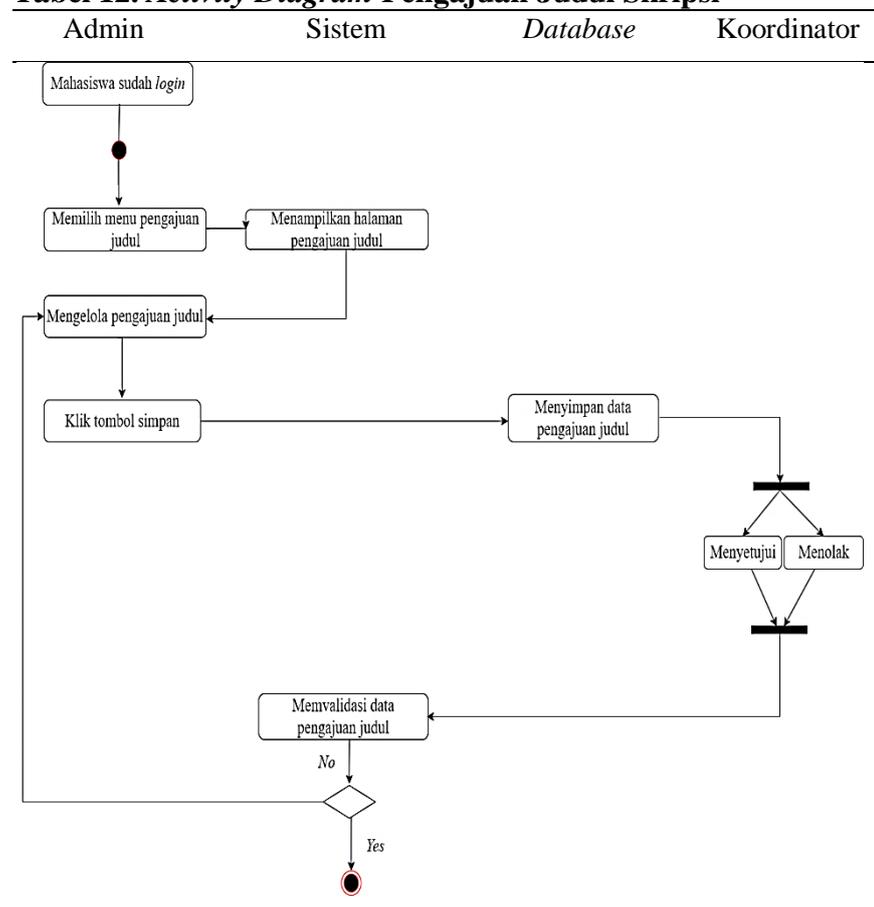
Tabel 11. Activity Diagram Tahun Akademik



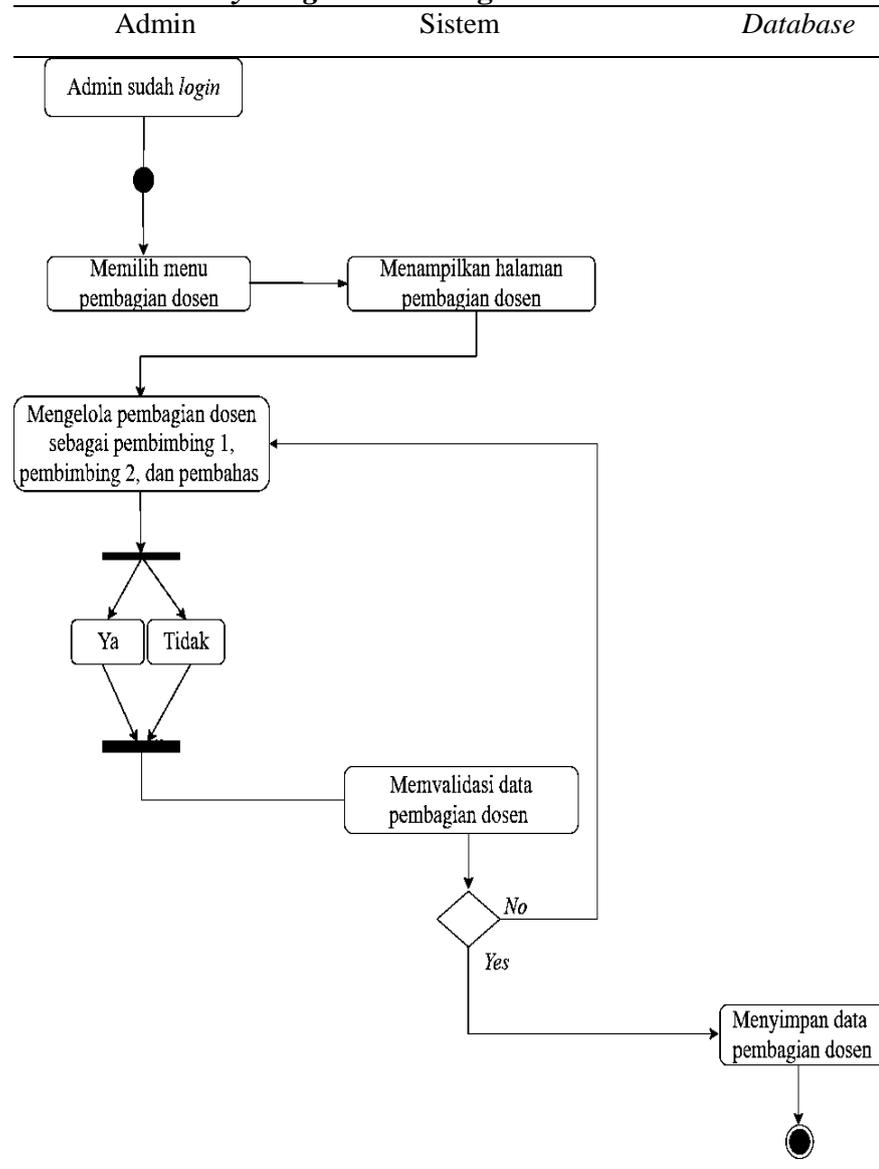
Tabel 11 menyajikan diagram aktivitas untuk halaman "Tahun Akademik" yang merupakan bagian dari data akademik yang ada pada admin, dapat dilakukan oleh admin dengan menekan tombol data akademik, yang nantinya sistem menampilkan data akademik. Setelah menampilkan data akademik, admin dapat melakukan menambah tahun akademik baru atau mengubah data akademik.

4. Activity Diagram Pengajuan Judul Skripsi

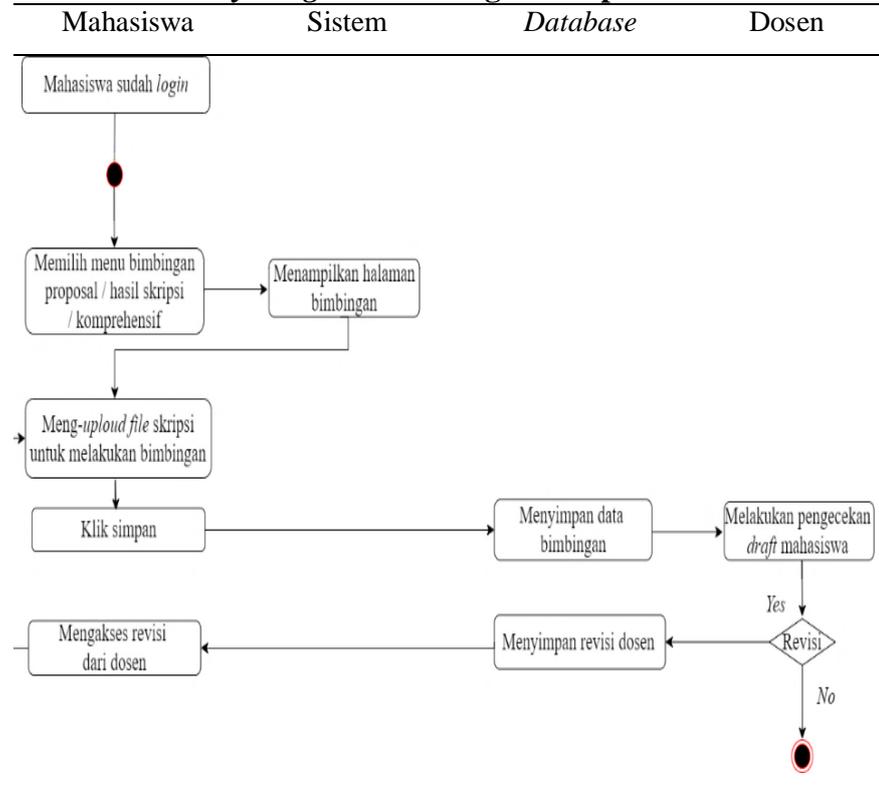
Tabel 12. Activity Diagram Pengajuan Judul Skripsi



Tabel 12 menyajikan *activity diagram* pengajuan judul skripsi, mahasiswa dapat meng-*upload* judul skripsi sesuai dengan *form* yang telah disediakan, kemudian admin memvalidasi data pengajuan judul skripsi untuk mendapat persetujuan apakah judul yang diajukan sudah layak atau belum, jika judul yang diajukan sudah layak, maka di-ACC atau disetujui dan mahasiswa dapat melanjutkan skripsinya.

5. *Activity Diagram* Pembagian Dosen**Tabel 13. Activity Diagram** Pembagian Dosen

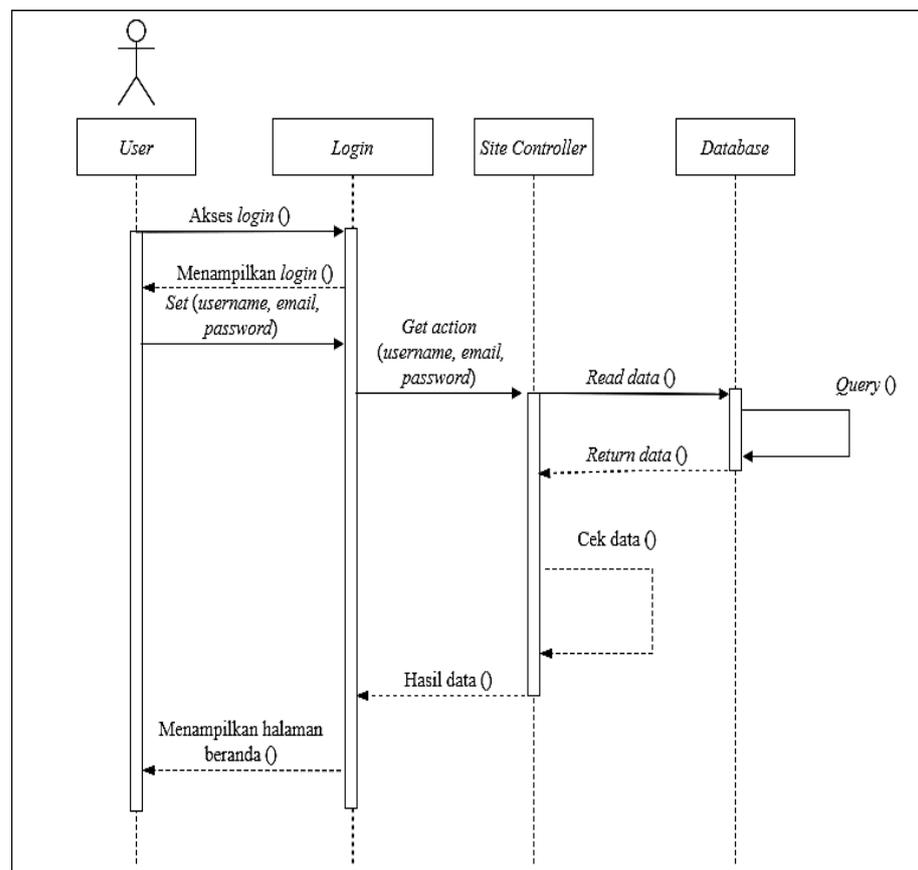
Tabel 13 menyajikan *activity diagram* pembagian dosen merupakan komponen yang sangat penting dalam proses skripsi, di mana adanya pembimbing dan pembahas untuk membantu mahasiswa dalam menyelesaikan proses skripsi. *Activity diagram* ini, admin yang memberikan penentuan kepada mahasiswa terkait dosen pembimbing 1, dosen pembimbing 2, dan dosen pembahas.

6. *Activity Diagram* Bimbingan Skripsi**Tabel 14. *Activity Diagram* Bimbingan Skripsi**

Tabel 14 menyajikan *activity diagram* bimbingan skripsi, mahasiswa meng-*upload* skripsi dan untuk menyelesaikan bimbingan, penting untuk berkonsultasi dengan dosen yang terkait. Nantinya dosen yang bersangkutan dapat men-*download file draft* skripsi yang dikirimkan oleh mahasiswa dan mengoreksi *draft* mahasiswa, jika ada revisi maka dosen memberikan revisi, dan mahasiswa dapat mengakses revisi pada aplikasi tersebut untuk diperbaharui. Jika telah selesai, mahasiswa dapat meng-*upload* kembali.

3.2.5.6 Design Sequence Diagram

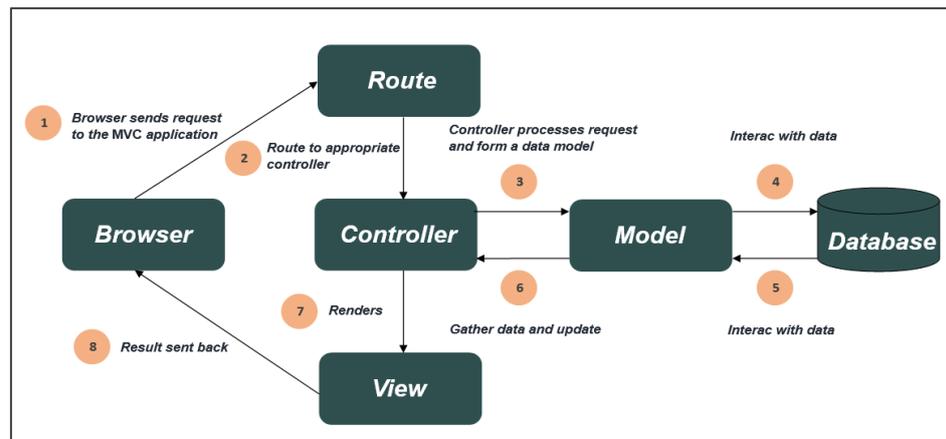
Sequence diagram menggambarkan adanya interaksi antar objek yang berada di dalam dan sekitar sistem termasuk *user*. *Sequence diagram* digunakan untuk menggambarkan rangkaian tahap yang dilakukan sebagai *respons* dari *event* untuk menghasilkan *output* tertentu. *Login* merupakan proses yang harus dilakukan ketika mengakses sebuah aplikasi atau sistem. Halaman ini dapat diakses oleh admin dan pengguna (*user*). Keduanya mengakses halaman *login* dan memasukkan *username* dan *password* mereka. Informasi *login* ini diverifikasi dengan *database*. Jika *login* berhasil, *user* diarahkan ke halaman utama aplikasi.



Gambar 6. Sequence Diagram Login

Gambar 6 menyajikan *sequence diagram login* yang menjelaskan interaksi antara objek-objek yang terlibat dalam proses *login user* ke sistem.

3.2.5.7 Design Diagram Model View Controller (MVC)



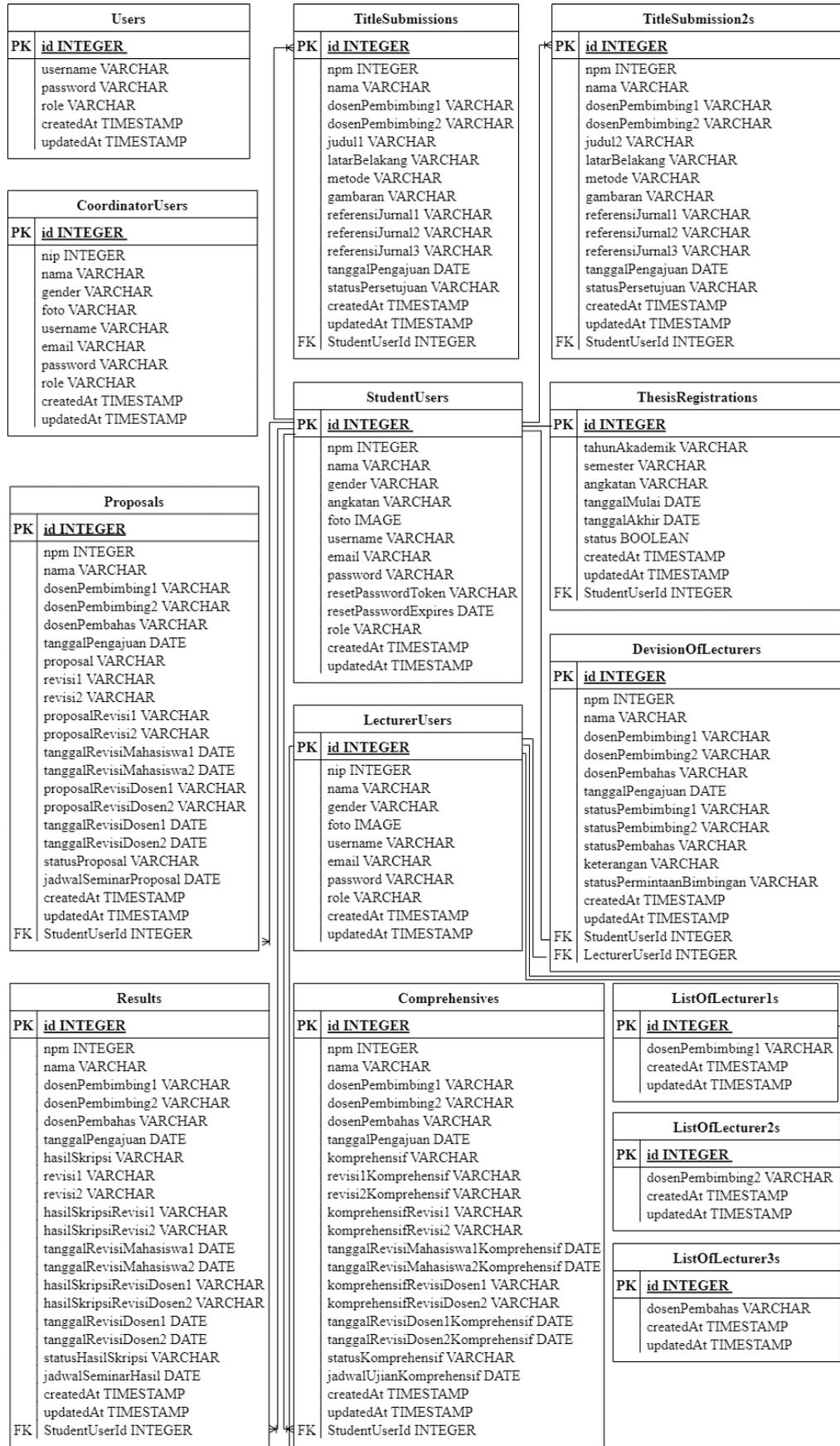
Gambar 7. Design Diagram Model View Controller (MVC)

Gambar 7 menyajikan *Design Diagram Model View Controller (MVC)*, yang menjelaskan aliran permintaan dan tanggapan dalam aplikasi MVC. Pengguna mengirimkan permintaan melalui *browser*, yang diterima oleh aplikasi dan diproses oleh *controller*. *Controller* membentuk model data, yang kemudian diinteraksi untuk mengumpulkan atau memperbarui data. Setelah itu, *controller* merender tampilan dan hasilnya dikirimkan kembali ke *browser*. Pendekatan MVC memisahkan tugas aplikasi menjadi *model*, *view*, dan *controller*, meningkatkan keterbacaan dan pemeliharaan kode.

3.2.5.8 Design Basis Data

Perancangan basis data berisi *Design* penyimpanan data pada sistem. Perancangan basis data mempunyai tujuan agar baata yang dibuat terstruktur dan sesuai dengan sistem yang dibuat. Perancangan basis data memetakan kebutuhan data agar sistem yang dibuat sesuai dengan tujuan penelitian. Perancangan basis data dalam sistem ini menggunakan *Entity Relationship Diagram (ERD)*.

ERD menjelaskan relasi antar entitas dalam perancangan suatu basis data. Setiap entitasnya mempunyai atribut yang merepresentasikan suatu tabel yang berisi berbagai kolom. Peneliti merancang sebuah ERD yang menggambarkan perancangan basis data pada sistem. ERD tersebut ditunjukkan pada Gambar 8.



Gambar 8. Entity Relationship Diagram

Gambar 8 mempunyai empat belas tabel yang mempunyai relasi dengan tabel lainnya. Setiap tabel tersebut mempunyai masing-masing struktur tabelnya. Berikut ini adalah penjelasan mengenai stuktur Tabel 15, Tabel 16, Tabel 17, Tabel 18, Tabel 19, Tabel 20, Tabel 21, Tabel 22, Tabel 23, Tabel 24, Tabel 25, Tabel 26, Tabel 27, dan Tabel 28:

a. Struktur Tabel *Users*

Tabel 15. *Users*

Nama Kolom	Tipe Data	Panjang	Keterangan
<i>Id</i>	<i>Integer</i>	10	<i>Primary Key</i>
<i>username</i>	<i>Varchar</i>	255	
<i>password</i>	<i>Varchar</i>	255	
<i>Role</i>	<i>Varchar</i>		
<i>createdAt</i>	<i>Timestamp</i>		<i>Nullable</i>
<i>updatedAt</i>	<i>Timestamp</i>		<i>Nullable</i>

Tabel 15 merupakan Tabel *Users* yang digunakan untuk menyimpan data admin. Kolom *id* berperan sebagai kunci utama (*primary key*) pada Tabel *Users*. Kolom *username* digunakan untuk menyimpan data *username* dari admin. Kolom *password* digunakan untuk menyimpan data kata sandi admin. Kolom *created_at* digunakan untuk menyimpan data tanggal ketika pertama kali *user* dibuat. Kolom *updated_at* menyimpan data tanggal ketika pengguna terakhir kali melakukan perubahan data.

b. Struktur Tabel *CoordinatorUsers*

Tabel 16. *CoordinatorUser*

Nama Kolom	Tipe Data	Panjang	Keterangan
<i>Id</i>	<i>Integer</i>	10	<i>Primary Key</i>
<i>NIP</i>	<i>Integer</i>	18	
<i>Nama</i>	<i>Varchar</i>	255	
<i>Gender</i>	<i>Varchar</i>		
<i>Foto</i>	<i>Image</i>		
<i>username</i>	<i>Varchar</i>	255	
<i>Email</i>	<i>Varchar</i>	191	<i>Unique</i>
<i>password</i>	<i>Varchar</i>		
<i>role</i>	<i>Varchar</i>		
<i>createdAt</i>	<i>Timestamp</i>		<i>Nullable</i>
<i>updatedAt</i>	<i>Timestamp</i>		<i>Nullable</i>

Tabel 16 merupakan Tabel *CoordinatorUsers* yang digunakan untuk menyimpan data *user* koordinator. Kolom *id* berperan sebagai kunci utama (*primary key*) pada Tabel *CoordinatorUsers*. Kolom NIP menyimpan data NIP koordinator. Kolom nama menyimpan data nama lengkap koordinator. Kolom *password* menyimpan data kata sandi koordinator. Kolom *gender* menyimpan data jenis kelamin koordinator. Kolom foto menyimpan data foto koordinator. Kolom *username* digunakan untuk menyimpan data *username* koordinator. Kolom *email* digunakan untuk menyimpan data alamat *email* koordinator. Kolom *password* digunakan untuk menyimpan data kata sandi koordinator. Kolom *created_at* digunakan untuk menyimpan data tanggal ketika pertama kali *user* dibuat. Kolom *updated_at* menyimpan data tanggal ketika pengguna terakhir kali melakukan perubahan data.

c. Struktur Tabel *LecturerUsers*

Tabel 17. *LecturerUser*

Nama Kolom	Tipe Data	Panjang	Keterangan
<i>id</i>	<i>Integer</i>	10	<i>Primary Key</i>
NIP	<i>Integer</i>	18	
nama	<i>Varchar</i>	255	
<i>gender</i>	<i>Varchar</i>		
foto	<i>Image</i>		
<i>username</i>	<i>Varchar</i>	255	
<i>email</i>	<i>Varchar</i>	191	<i>Unique</i>
<i>password</i>	<i>Varchar</i>		
<i>role</i>	<i>Varchar</i>		
<i>createdAt</i>	<i>Timestamp</i>		<i>Nullable</i>
<i>updatedAt</i>	<i>Timestamp</i>		<i>Nullable</i>

Tabel 17 merupakan Tabel *LecturerUsers* yang digunakan untuk menyimpan data dosen. Kolom *id* berperan sebagai kunci utama (*primary key*) pada Tabel *LecturerUsers*. Kolom NIP menyimpan data NIP dosen. Kolom nama menyimpan data nama lengkap dosen. Kolom *password* menyimpan data kata sandi dosen. Kolom *gender* menyimpan data jenis kelamin dosen. Kolom foto menyimpan data foto dosen. Kolom *username* digunakan untuk menyimpan data *username* dosen. Kolom *email* digunakan untuk menyimpan data alamat *email* dosen. Kolom *password* digunakan untuk menyimpan data kata sandi dosen. Kolom

created_at digunakan untuk menyimpan data tanggal ketika pertama kali *user* dibuat. Kolom *updated_at* menyimpan data tanggal ketika pengguna terakhir kali melakukan perubahan data. Tabel *LecturerUsers* memiliki relasi 1:1 (*One to One*) terhadap Tabel *DevisionOfLecturers*. Tabel *LecturerUsers* memiliki relasi 1:1 (*One to One*) terhadap Tabel *TitleSubmission1s*, *TitleSubmission2s* dan *TitleSubmission3s*.

d. Struktur Tabel *StudentUsers*

Tabel 18. *StudentUser*

Nama Kolom	Tipe Data	Panjang	Keterangan
<i>id</i>	<i>Integer</i>	10	<i>Primary Key</i>
NPM	<i>Integer</i>	10	
nama	<i>Varchar</i>	255	
<i>gender</i>	<i>Varchar</i>		
foto	<i>Image</i>		
<i>username</i>	<i>Varchar</i>	255	
<i>email</i>	<i>Varchar</i>	191	<i>Unique</i>
<i>password</i>	<i>Varchar</i>		
<i>role</i>	<i>Varchar</i>		
<i>createdAt</i>	<i>Timestamp</i>		<i>Nullable</i>
<i>updatedAt</i>	<i>Timestamp</i>		<i>Nullable</i>

Tabel 18 merupakan Tabel *StudentUsers* yang digunakan untuk menyimpan data mahasiswa. Kolom *id* berperan sebagai kunci utama (*primary key*) pada Tabel *StudentUsers*. Kolom NPM menyimpan data NPM mahasiswa. Kolom nama menyimpan data nama lengkap mahasiswa. Kolom *password* menyimpan data kata sandi mahasiswa. Kolom *gender* menyimpan data jenis kelamin mahasiswa. Kolom foto menyimpan data foto mahasiswa. Kolom *username* digunakan untuk menyimpan data *username* mahasiswa. Kolom *email* digunakan untuk menyimpan data alamat *email* mahasiswa. Kolom *password* digunakan untuk menyimpan data kata sandi mahasiswa. Kolom *created_at* digunakan untuk menyimpan data tanggal ketika pertama kali *user* dibuat. Kolom *updated_at* menyimpan data tanggal ketika pengguna terakhir kali melakukan perubahan data. Tabel *StudentUser* memiliki relasi 1:1 (*One to One*) terhadap Tabel *LecturerProfile*, 1:1 (*One to One*) terhadap Tabel *ThesisRegistrations* dan *DevisionOfLecturers*, 1:M (*One*

to Many) terhadap Tabel *TitleSubmissions*, *TitleSubmission2*, *Proposals*, *Results*, dan *Comprehensives*.

e. Struktur Tabel *ThesisRegistrations*

Tabel 19. *ThesisRegistrations*

Nama Kolom	Tipe Data	Panjang	Keterangan
<i>id</i>	<i>Integer</i>	10	<i>Primary Key</i>
<i>tahunAkademik</i>	<i>Integer</i>		
<i>semester</i>	<i>Integer</i>		
<i>angkatan</i>	<i>Integer</i>		
<i>tanggalMulai</i>	<i>Date</i>		
<i>tanggalAkhir</i>	<i>Date</i>		
<i>StudentUserId</i>	<i>Integer</i>		<i>Foreign Key</i>
<i>createdAt</i>	<i>Timestamp</i>		<i>Nullable</i>
<i>updatedAt</i>	<i>Timestamp</i>		<i>Nullable</i>

Tabel 19 memiliki berbagai relasi dengan tabel lainnya, di antaranya relasi 1:1 (*One to One*) dengan Tabel *StudentUsers*.

f. Struktur Tabel *TitleSubmissions*

Tabel 20. *TitleSubmissions*

Nama Kolom	Tipe Data	Panjang	Keterangan
<i>Id</i>	<i>Integer</i>	10	<i>Primary Key</i>
<i>NPM</i>	<i>Varchar</i>	10	
<i>nama</i>	<i>Varchar</i>	255	
<i>dosenPembimbing1</i>	<i>Varchar</i>	255	
<i>dosenPembimbing2</i>	<i>Varchar</i>	255	
<i>judul1</i>	<i>Varchar</i>	255	
<i>latarBelakang</i>	<i>Varchar</i>	255	
<i>metode</i>	<i>Varchar</i>	255	
<i>gambaran</i>	<i>Varchar</i>	255	
<i>referensiJurnal1</i>	<i>Varchar</i>	255	
<i>referensiJurnal2</i>	<i>Varchar</i>	255	
<i>referensiJurnal3</i>	<i>Varchar</i>	255	
<i>tanggalPengajuan</i>	<i>Date</i>		
<i>statusPersetujuan</i>	<i>Varchar</i>		
<i>StudentUserId</i>	<i>Integer</i>		
<i>createdAt</i>	<i>Timestamp</i>		<i>Nullable</i>
<i>updatedAt</i>	<i>Timestamp</i>		<i>Nullable</i>

Tabel 20 memiliki relasi M:1 (*Many to One*) terhadap Tabel *StudentUsers*.

g. Struktur Tabel *TitleSubmission2s***Tabel 21. *TitleSubmission2s***

Nama Kolom	Tipe Data	Panjang	Keterangan
<i>Id</i>	<i>Integer</i>	10	<i>Primary Key</i>
NPM	<i>Varchar</i>	10	
nama	<i>Varchar</i>	255	
dosenPembimbing1	<i>Varchar</i>	255	
dosenPembimbing2	<i>Varchar</i>	255	
judul2	<i>Varchar</i>	255	
latarBelakang	<i>Varchar</i>	255	
metode	<i>Varchar</i>	255	
gambaran	<i>Varchar</i>	255	
referensiJurnal1	<i>Varchar</i>	255	
referensiJurnal2	<i>Varchar</i>	255	
referensiJurnal3	<i>Varchar</i>	255	
tanggalPengajuan	<i>Date</i>		
statusPersetujuan	<i>Varchar</i>		
<i>StudentUserId</i>	<i>Integer</i>		
<i>createdAt</i>	<i>Timestamp</i>		<i>Nullable</i>
<i>updatedAt</i>	<i>Timestamp</i>		<i>Nullable</i>

Tabel 21 memiliki relasi M:1 (*Many to One*) terhadap Tabel *StudentUsers*.

h. Struktur Tabel *ListOfLecturer1s***Tabel 22. *ListOfLecturer1s***

Nama Kolom	Tipe Data	Panjang	Keterangan
<i>Id</i>	<i>Integer</i>	10	<i>Primary Key</i>
dosenPembimbing1	<i>Varchar</i>		
<i>createdAt</i>	<i>Timestamp</i>		<i>Nullable</i>
<i>updatedAt</i>	<i>Timestamp</i>		<i>Nullable</i>

Tabel 22 memiliki relasi 1:1 (*One to One*) terhadap Tabel *LecturerUsers*.

i. Struktur Tabel *ListOfLecturer2s***Tabel 23. *ListOfLecturer2s***

Nama Kolom	Tipe Data	Panjang	Keterangan
<i>Id</i>	<i>Integer</i>	10	<i>Primary Key</i>
dosenPembimbing2	<i>Varchar</i>		
<i>createdAt</i>	<i>Timestamp</i>		<i>Nullable</i>
<i>updatedAt</i>	<i>Timestamp</i>		<i>Nullable</i>

Tabel 23 memiliki relasi 1:1 (*One to One*) terhadap Tabel *LecturerUsers*.

j. Struktur Tabel *ListOfLecturer3s***Tabel 24. *ListOfLecturer3s***

Nama Kolom	Tipe Data	Panjang	Keterangan
<i>Id</i>	<i>Integer</i>	10	<i>Primary Key</i>
<i>dosenPembahas</i>	<i>Varchar</i>		
<i>createdAt</i>	<i>Timestamp</i>		<i>Nullable</i>
<i>updatedAt</i>	<i>Timestamp</i>		<i>Nullable</i>

Tabel 24 memiliki relasi 1:1 (*One to One*) terhadap Tabel *LecturerUsers*.

k. Struktur Tabel *DevisiionOfLecturers***Tabel 25. *DevisiionOfLecturers***

Nama Kolom	Tipe Data	Panjang	Keterangan
<i>id</i>	<i>Integer</i>	10	<i>Primary Key</i>
<i>NPM</i>	<i>Integer</i>	10	
<i>nama</i>	<i>Varchar</i>	255	
<i>dosenPembimbing1</i>	<i>Varchar</i>		
<i>dosenPembimbing2</i>	<i>Varchar</i>		
<i>dosenPembahas</i>	<i>Varchar</i>		
<i>tanggalPengajuan</i>	<i>Date</i>		
<i>statusPembimbing1</i>	<i>Varchar</i>		
<i>statusPembimbing2</i>	<i>Varchar</i>		
<i>statuspembahas</i>	<i>Varchar</i>		
<i>keterangan</i>	<i>Varchar</i>		
<i>statusPermintaan</i>	<i>Varchar</i>		
<i>StudentUserId</i>	<i>Integer</i>		
<i>LecturerUserId</i>	<i>Integer</i>		
<i>createdAt</i>	<i>Timestamp</i>		<i>Nullable</i>
<i>updatedAt</i>	<i>Timestamp</i>		<i>Nullable</i>

Tabel 25 memiliki relasi 1:1 (*One to One*) terhadap Tabel *StudentUsers* dan *LecturerUsers*.

l. Struktur Tabel *Proposals***Tabel 26. *Proposals***

Nama Kolom	Tipe Data	Panjang	Keterangan
<i>Id</i>	<i>Integer</i>	10	<i>Primary Key</i>
<i>NPM</i>	<i>Integer</i>	10	
<i>nama</i>	<i>Varchar</i>	255	
<i>dosenPembimbing1</i>	<i>Varchar</i>		
<i>dosenPembimbing2</i>	<i>Varchar</i>		
<i>dosenPembahas</i>	<i>Varchar</i>		
<i>tanggalPengajuan</i>	<i>Date</i>		

proposal	Varchar	
revisi1	Varchar	
revisi2	Varchar	
proposalRevisi1	Varchar	
proposalRevisi2	Varchar	
tanggalRevisiMahasiswa1	Date	
tanggalRevisiMahasiswa2	Date	
proposalRevisiDosen1	Varchar	
proposalRevisiDosen2	Varchar	
tanggalRevisiDosen1	Date	
tanggalRevisiDosen1	Date	
statusProposal	Varchar	
jadwalSeminarProposal	Date	
StudentUserId	Integer	Foreign Key
createdAt	Timestamp	Nullable
updatedAt	Timestamp	Nullable

Tabel 26 memiliki relasi M:1 (*Many to One*) terhadap Tabel *StudentUsers*.

m. Struktur Tabel *Results*

Tabel 27. Results

Nama Kolom	Tipe Data	Panjang	Keterangan
<i>Id</i>	<i>Integer</i>	10	<i>Primary Key</i>
NPM	<i>Integer</i>	10	
nama	<i>Varchar</i>	255	
dosenPembimbing1	<i>Varchar</i>		
dosenPembimbing2	<i>Varchar</i>		
dosenPembahas	<i>Varchar</i>		
tanggalPengajuan	<i>Date</i>		
hasilSkripsi	<i>Varchar</i>		
revisi1	<i>Varchar</i>		
revisi2	<i>Varchar</i>		
hasilSkripsiRevisi1	<i>Varchar</i>		
hasilSkripsiRevisi2	<i>Varchar</i>		
tanggalRevisiMahasiswa1	<i>Date</i>		
tanggalRevisiMahasiswa2	<i>Date</i>		
hasilSkripsiRevisiDosen1	<i>Varchar</i>		
hasilSkripsiRevisiDosen2	<i>Varchar</i>		
tanggalRevisiDosen1	<i>Date</i>		
tanggalRevisiDosen1	<i>Date</i>		
statusHasilSkripsi	<i>Varchar</i>		
jadwalSeminarProposal	<i>Date</i>		
StudentUserId	<i>Integer</i>		<i>Foreign Key</i>
createdAt	<i>Timestamp</i>		<i>Nullable</i>
updatedAt	<i>Timestamp</i>		<i>Nullable</i>

Tabel 27 memiliki relasi M:1 (*Many to One*) terhadap Tabel *StudentUsers*.

n. Struktur Tabel *Comprehensives*

Tabel 28. *Comprehensives*

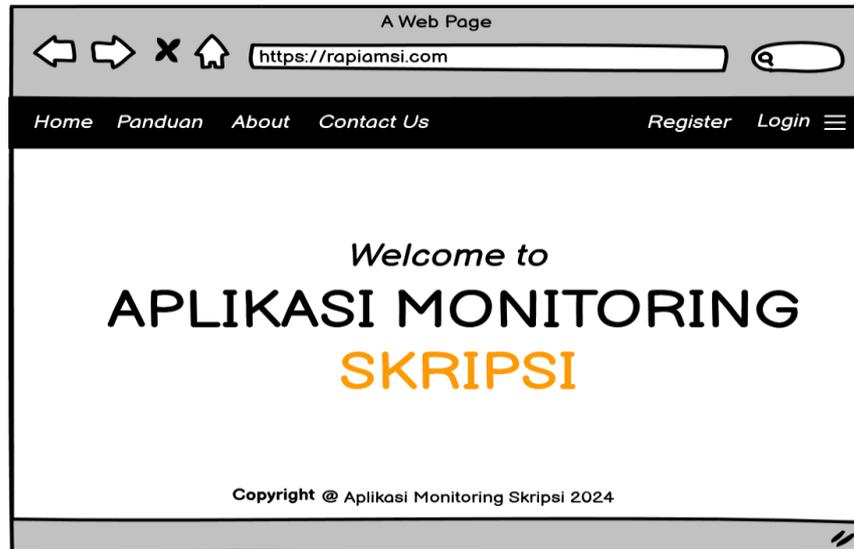
Nama Kolom	Tipe Data	Panjang	Keterangan
<i>Id</i>	<i>Integer</i>	10	<i>Primary Key</i>
<i>NPM</i>	<i>Integer</i>	10	
<i>nama</i>	<i>Varchar</i>	255	
<i>dosenPembimbing1</i>	<i>Varchar</i>		
<i>dosenPembimbing2</i>	<i>Varchar</i>		
<i>dosenPembahas</i>	<i>Varchar</i>		
<i>tanggalPengajuan</i>	<i>Date</i>		
<i>komprehensif</i>	<i>Varchar</i>		
<i>revisi1Komprehensif</i>	<i>Varchar</i>		
<i>revisi2Komprehensif</i>	<i>Varchar</i>		
<i>komprehensif Revisi1</i>	<i>Varchar</i>		
<i>komprehensif Revisi2</i>	<i>Varchar</i>		
<i>tanggalRevisiMahasiswa1</i>	<i>Date</i>		
<i>tanggalRevisiMahasiswa2</i>	<i>Date</i>		
<i>komprehensifRevisiDosen1</i>	<i>Varchar</i>		
<i>komprehensifRevisiDosen2</i>	<i>Varchar</i>		
<i>tanggalRevisiDosen1</i>	<i>Date</i>		
<i>tanggalRevisiDosen2</i>	<i>Date</i>		
<i>statusKomprehensif</i>	<i>Varchar</i>		
<i>jadwalUjianKomprehensif</i>	<i>Date</i>		
<i>StudentUserId</i>	<i>Integer</i>		<i>Foreign Key</i>
<i>createdAt</i>	<i>Timestamp</i>		<i>Nullable</i>
<i>updatedAt</i>	<i>Timestamp</i>		<i>Nullable</i>

Tabel 28 memiliki relasi M:1 (*Many to One*) terhadap Tabel *StudentUsers*.

3.2.5.9 Design Antarmuka

Design interface atau antarmuka dilakukan sebelum melakukan tahapan implementasi pada sebuah sistem dengan tujuan untuk mencapai hasil yang optimal. *Design* antarmuka mencakup sisi admin, dosen, dan mahasiswa yang menjadi dasar dalam pembuatan aplikasi berbasis *website*.

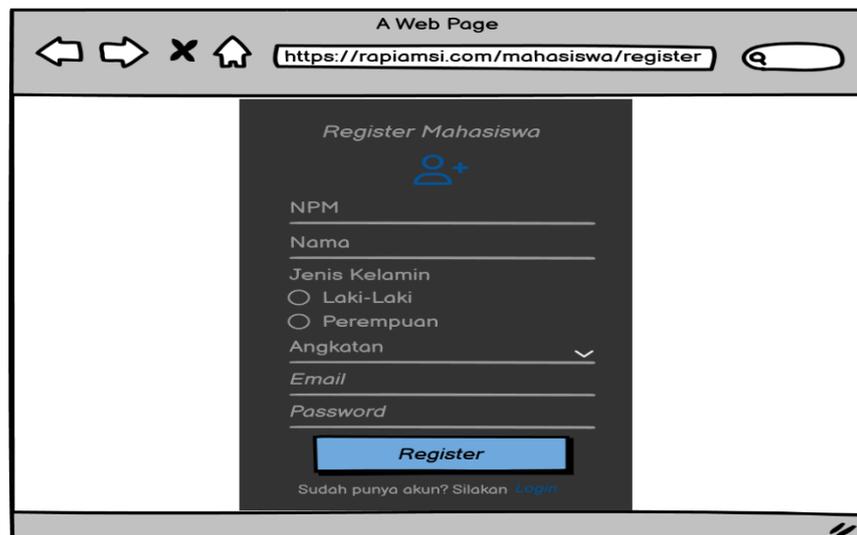
1. Halaman Awal



Gambar 9. Design Halaman Awal

Gambar 9 merupakan *design* halaman awal. Terdapat beberapa menu pada halaman tersebut, yaitu menu *home*, menu *panduan*, menu *about*, menu *contact us*, menu *register*, dan menu *login*, yang nantinya setelah *login* diarahkan pada masing-masing halaman sesuai *role*.

2. Halaman Register



Gambar 10. Design Halaman Register Mahasiswa

Gambar 10 merupakan *design* halaman *register* mahasiswa. Proses *register* mahasiswa berfungsi untuk mendaftarkan akun pada sistem. Halaman *register* menampilkan formulir yang perlu diisi oleh mahasiswa untuk melengkapi proses registrasi.

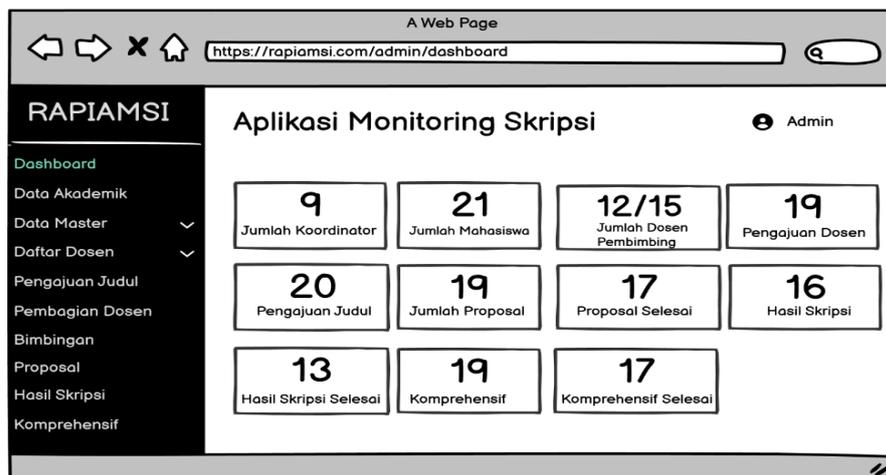
3. Halaman *Login*



Gambar 11. Design Halaman Login

Gambar 11 merupakan *design* halaman *login* (*login page*). Sebelum mengakses sistem, *user* perlu melakukan *login* dengan memasukkan *email*, *username*, dan *password* yang sudah terdaftar. Teks “*forgot password??*” berfungsi sebagai tautan yang mengarahkan *user* pada halaman *reset password* yang dikirimkan melalui *email*.

4. Halaman *Dashboard Admin*

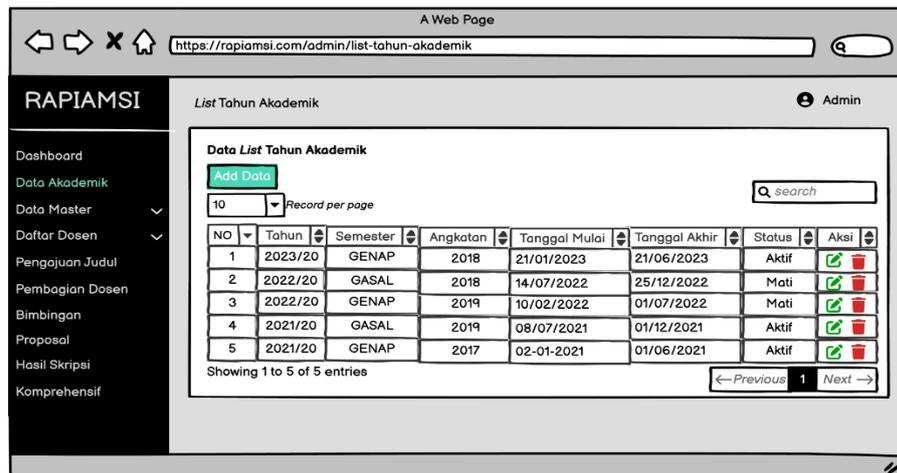


Gambar 12. Design Halaman Dashboard Admin

Gambar 12 merupakan *design* halaman *dashboard* admin yang ditampilkan setelah admin masuk ke dalam sistem, pada *dashboard* atau halaman utama, admin dapat melihat jumlah koordinator, jumlah mahasiswa, jumlah dosen pembimbing, jumlah pengajuan dosen, jumlah pengajuan judul, jumlah proposal, jumlah proposal selesai, jumlah hasil skripsi, jumlah hasil skripsi selesai, jumlah komprehensif, dan jumlah komprehensif selesai.

jumlah pengajuan judul, jumlah proposal, jumlah mahasiswa yang selesai proposal, jumlah hasil skripsi, jumlah mahasiswa yang selesai hasil skripsi, jumlah komprehensif, dan jumlah mahasiswa yang selesai sidang komprehensif atau skripsi.

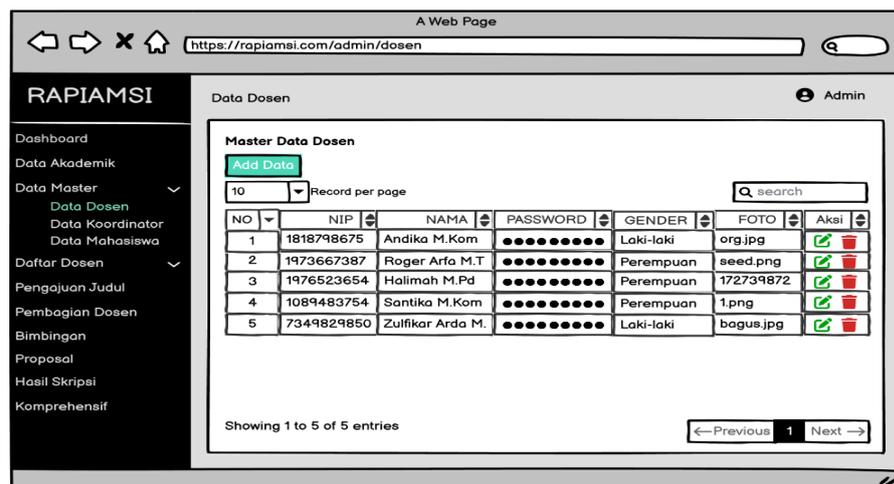
5. Halaman *List Tahun Akademik*



Gambar 13. Design Halaman *List Tahun Akademik*

Gambar 13 merupakan *design* halaman *list* tahun akademik yang menampilkan tahun akademik yang aktif atau yang sudah tidak aktif.

7. Halaman *Data Master*



Gambar 14. Design Halaman *Data Master*

Gambar 14 merupakan *design* halaman *data master* yang tersedia pada admin, admin dapat menambahkan, mengedit dan menghapus data master yang sudah terdaftar pada sistem.

8. Halaman Pengajuan Judul

https://rapiamsi.com/admin/pengajuan-judul

RAPIAMSI Admin

Data Pengajuan Judul

10 Record per page

search

NO	NPM	NAMA	PEMBIMBING 1	PEMBIMBING 2	JUDUL 1	JUDUL 2	STATUS	Aksi
1	1813025	Ariana	Dr. Ariyanto	Teddy, M.T.	Judul 1	Judul 2	Judul 1 Diterima	✓ ✗
2	1813025	Berlian	Teddy, M.T.	Elvin, M.Kom.	Judul 1	Judul 2	Judul 2 Diterima	✓ ✗
3	1813025	Davindo	Elvin, M.Kom.	Tata, M.T.	Judul 1	Judul 2	Judul 1 Diterima	✓ ✗
4	1813025	Stevi	Christanto, M.Pd.	Dr. Ariyanto	Judul 1	Judul 2	Judul 1 Diterima	✓ ✗
5	1813025	Zafira	Tata, M.T.	Christanto, M.Pd.	Judul 1	Judul 2	Judul 1 Ditolak	✓ ✗

Showing 1 to 5 of 5 entries

← Previous 1 Next →

Gambar 15. Design Halaman Pengajuan Judul

Gambar 15 merupakan *design* halaman pengajuan judul yang berisi tentang data pengajuan judul mahasiswa yang terdiri NPM, nama, pembimbing 1, pembimbing 2, judul 1, judul 2, judul revisi, status persetujuan terkait judul skripsi, dan aksi.

9. Halaman *List* Pembagian Dosen

https://rapiamsi.com/admin/pembagian-dosen

RAPIAMSI Admin

List Pembagian Dosen

10 Record per page

search

NO	NPM	NAMA	PB 1	PB 2	PEMBAHAS	TGL PENGAJUAN	STATUS PB 1	STATUS PB 2	STATUS PMBHS	Aksi
1	1813025	Ariana	Dr. Ariyanto	Teddy, M.T.	Suyatna, M.Kom.	1/2/2023	Diterima	Diterima	Diterima	Ubah
2	1813025	Berlian	Teddy, M.T.	Elvin, M.Ko	Dr. Ariyanto	2/2/2023	Diterima	Diterima	Diterima	Ubah
3	1813025	Davindo	Elvin, M.Ko	Tata, M.T.	Christanto, M.Pd.	3/2/2023	Diterima	Diterima	Diterima	Ubah
4	1813025	Stevi	Christanto,	Dr. Ariyanto	Elvin, M.Kom.	3/2/2023	Diterima	Diterima	Diterima	Ubah
5	1813025	Zafira	Tata, M.T.	Christanto,	Teddy, M.T.	4/2/2023	Diterima	Diterima	Diterima	Ubah

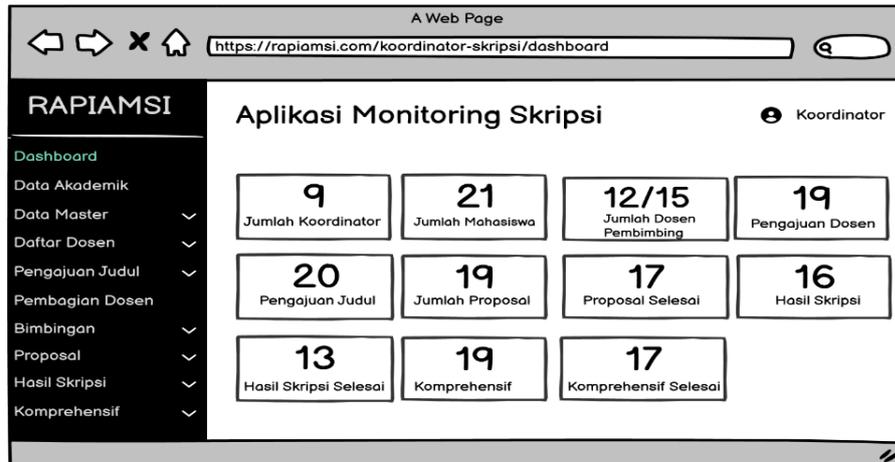
Showing 1 to 5 of 5 entries

← Previous 1 Next →

Gambar 16. Design Halaman *List* Pembagian Dosen

Gambar 16 merupakan *design* halaman *list* pembagian dosen, menampilkan *list* pembagian dosen yang terdiri dari NPM, nama, pembimbing 1, pembimbing 2, pembahas, tanggal pengajuan, status pembimbing1, status pembimbing 2, status pembahas, dan aksi.

10. Halaman Tampilan Koordinator



Gambar 17. Design Halaman Tampilan Koordinator

Gambar 17 merupakan *design* halaman tampilan koordinator yang menampilkan jumlah koordinator, jumlah mahasiswa, jumlah dosen pembimbing, jumlah pengajuan dosen, jumlah pengajuan judul, jumlah proposal, jumlah mahasiswa yang selesai proposal, jumlah hasil skripsi, jumlah mahasiswa yang selesai hasil skripsi, jumlah komprehensif, dan jumlah mahasiswa yang selesai sidang komprehensif atau skripsi.

11. Halaman Tampilan Dosen



Gambar 18. Design Halaman Tampilan Dosen

Gambar 18 merupakan *design* halaman tampilan dosen yang ditampilkan setelah dosen masuk ke dalam sistem, pada *dashboard* atau halaman utama, dosen dapat melihat informasi yang ditampilkan.

12. Bimbingan Mahasiswa dengan Dosen

A Web Page
https://rapiamsi.com/dosen/bimbingan/proposal

RAPIAMSI Dosen

Data Bimbingan Proposal

10 Record per page

Q search

NO	NPM	NAMA	JUDUL	PEMBIMBING 1	PEMBIMBING 2	PROPOSAL	STATUS	AKSI
1	1813025008	Dewi Ayu Lestari	Implementasi REST	Julian Roger, M.T.	Holimah, M.Pd.	Skripsi.pdf	Diterima	
2	1813025034	Stevani	Rancang Bangun Ap	Holimah, M.Pd.	Julian Roger, M.T.	Proposal.pdf	Ditolak	

Showing 1 to 5 of 5 entries

← Previous 1 Next →

Gambar 19. Design Halaman Bimbingan pada Dosen

Gambar 19 merupakan *design* halaman bimbingan pada dosen, pada halaman ini dosen pembimbing dapat melihat *file* mahasiswa, memberikan revisi, dan status persetujuan bimbingan sesuai dengan pencapaian mahasiswa.

13. Halaman Jadwal pada Dosen

A Web Page
https://rapiamsi.com/dosen/jadwal/proposal

RAPIAMSI Dosen

Jadwal Seminar Proposal

10 Record per page

Q search

NO	NPM	NAMA	JUDUL	TANGGAL	SEBAGAI	PROPOSAL	AKSI
1	1813025008	Dewi Ayu Lestari	Implementasi REST	21/04/2023	Pembimbing 1		Lihat Data
2	1813025034	Stevani	Rancang Bangun Ap	06/03/2023	Pembahas		Lihat Data

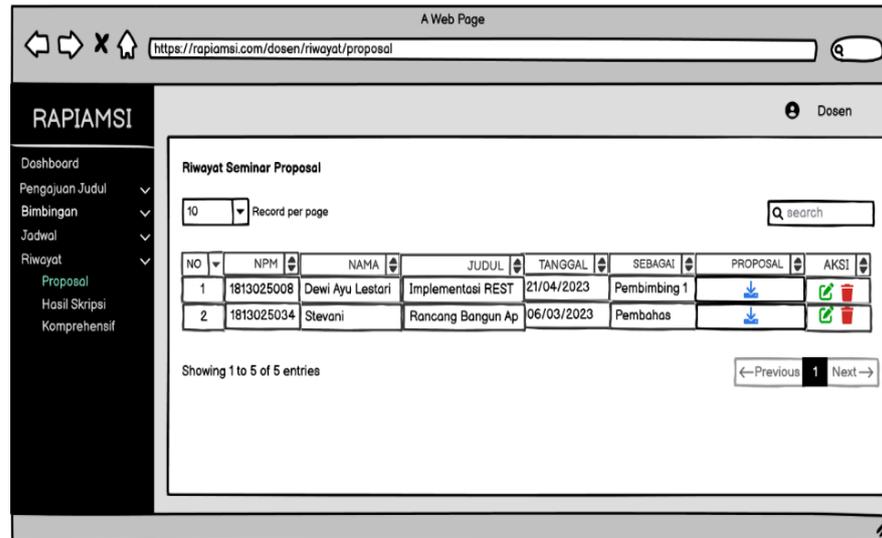
Showing 1 to 5 of 5 entries

← Previous 1 Next →

Gambar 20. Design Halaman Jadwal pada Dosen

Gambar 20 merupakan *design* halaman jadwal pada dosen, halaman jadwal hanya diberikan kepada pembahas ketika melakukan seminar, baik seminar proposal, seminar hasil, atau komprehensif. Dosen pembimbing dan pembahas dapat diberitahukan dan dapat *download file* berkas mahasiswa.

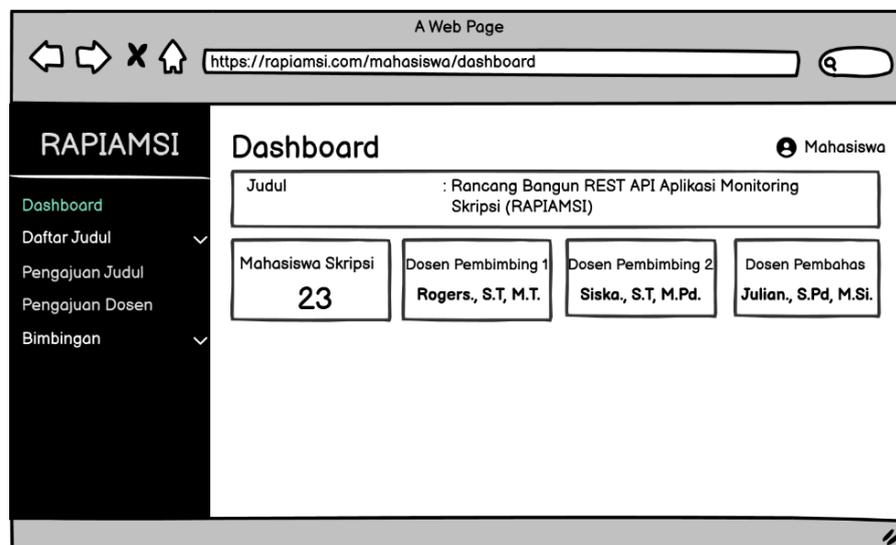
14. Halaman Riwayat



Gambar 21. Design Halaman Riwayat pada Dosen

Gambar 21 merupakan *design* halaman riwayat pada dosen baik sebagai pembimbing atau pembahas.

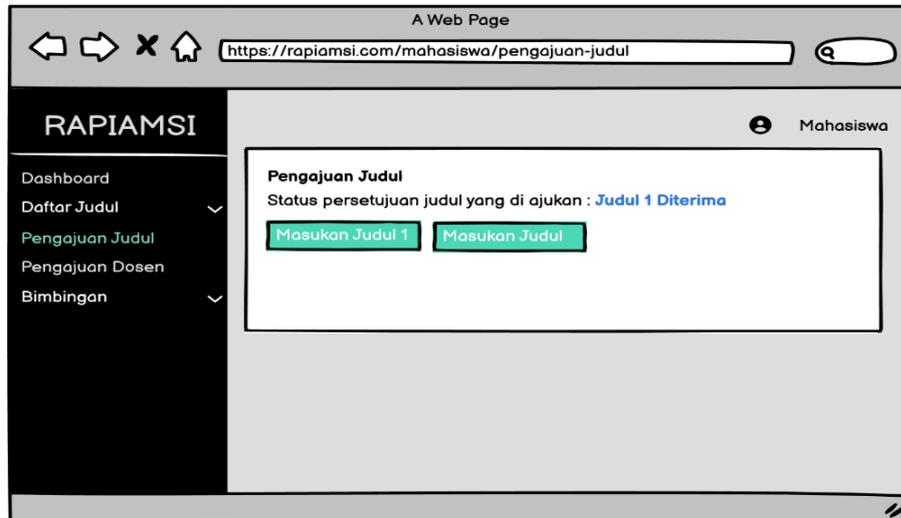
15. Halaman Tampilan Mahasiswa



Gambar 22. Design Halaman Tampilan Mahasiswa

Gambar 22 merupakan *design* halaman tampilan mahasiswa. Halaman ini menampilkan *profile* mahasiswa, judul skripsi mahasiswa, dosen pembimbing dan dosen pembahas setelah pembagian dosen oleh admin, mengetahui status skripsi mahasiswa, dan fitur yang dapat digunakan untuk melakukan bimbingan dengan dosen pembimbing nantinya.

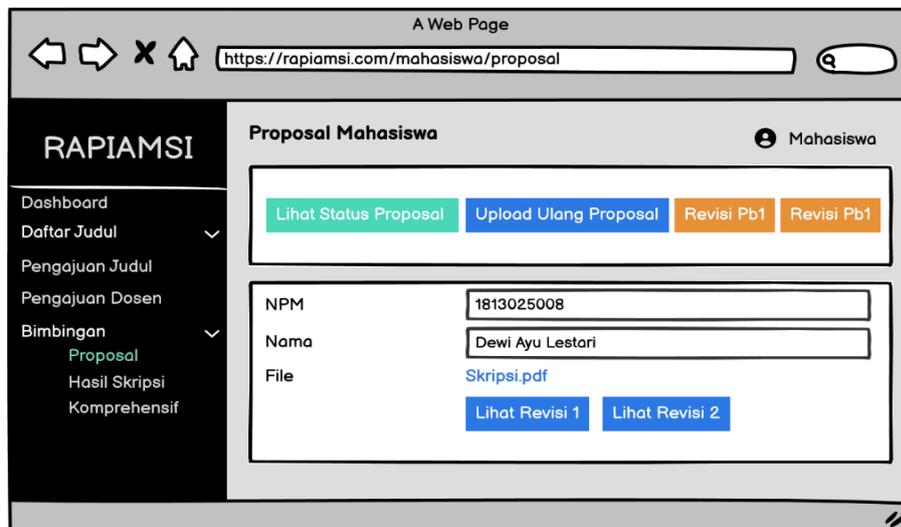
16. Pengajuan Judul Skripsi Mahasiswa



Gambar 23. Design Halaman Pengajuan Judul

Gambar 23 merupakan *design* halaman pengajuan judul. Mahasiswa dapat mengajukan judul skripsi 1, judul skripsi 2 dan dapat melihat *history* pengajuan judul skripsinya.

17. Halaman Bimbingan



Gambar 24. Design Halaman Bimbingan Mahasiswa

Gambar 24 merupakan *design* halaman bimbingan mahasiswa yang menampilkan informasi *history* bimbingan mahasiswa, di dalam halaman ini terdapat menu yang memungkinkan mahasiswa untuk melakukan bimbingan baru, *upload* ulang, melihat status bimbingan yang sedang berlangsung, melihat revisi, dan *upload* revisi.

3.5.3 Implementation

Implementasi merupakan tahap pengkodean dari seluruh tahapan *design*. Tahapan ini dilakukan pembuatan *backend* aplikasi *monitoring* skripsi menggunakan bahasa pemrograman *javascript*, *framework Node JS* dan *Express JS*, *database postgresQL* dengan menggunakan *Dbeaver*, *Sequelize ORM* untuk mempermudah lalu lintas data ke dalam *database*, *GIT* sebagai *Version Control System*, *Web browser chrome*, menggunakan aplikasi *postman* yang merupakan *tools* untuk mempermudah *testing* REST API.

3.5.4 Testing

Tahap uji coba/*testing* yaitu melakukan uji kelayakan sistem yang difokuskan pada fungsi program. Tujuan utamanya adalah untuk memastikan bahwa semua perintah yang telah di-*input* ke dalam program menghasilkan *output* seperti yang diharapkan dengan berbagai pengaksesan serta pemrosesan seperti menambah data, mengubah data dan menghapus data. Proses ini dilakukan melalui akses yang beragam terhadap masing-masing REST API yang telah dibuat pada situs *Uniform Resource Locator* (URL) dengan menggunakan bantuan *postman*.

Selain itu, pengujian keamanan juga penting untuk memastikan bahwa sistem tidak rentan terhadap serangan atau pelanggaran keamanan. Untuk itu, digunakan perangkat lunak seperti *OWASP ZAP*. Penggunaan kedua alat ini membantu memastikan bahwa sistem tidak hanya berfungsi dengan baik secara fungsional, tetapi juga terlindungi dari serangan keamanan yang berpotensi merugikan. Oleh karena itu, pengujian dilakukan secara menyeluruh untuk memastikan bahwa API dapat beroperasi secara optimal, baik dari segi fungsionalitas maupun keamanan.

V. PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan dapat disimpulkan:

1. Perancangan dan pembangunan dalam penelitian ini, peneliti menggunakan metode *waterfall* yang melibatkan empat tahapan utama: *Analysis, Design, Implementation, dan Testing*. Setiap tahapan dijalankan secara terstruktur dan berurutan untuk memastikan kelancaran proses pengembangan aplikasi *monitoring* skripsi.
2. Peneliti berhasil merancang dan membangun aplikasi *monitoring* skripsi dengan mengimplementasikan arsitektur REST yang dilakukan dari sisi *Backend Development*. Aplikasi ini berhasil dibangun untuk empat peran utama, yaitu admin, koordinator skripsi, dosen, dan mahasiswa. Terdapat total 395 *endpoint* yang dirancang telah berhasil dibuat, dengan 139 *endpoint* menggunakan metode GET dan 256 *endpoint* menggunakan metode POST.
3. Fitur-fitur yang disediakan oleh aplikasi mencakup proses registrasi, *login*, pengajuan judul skripsi, pengajuan dosen pembimbing dan pembahas, serta bimbingan dengan dosen pembimbing yang telah ditetapkan. Mahasiswa juga diberikan akses untuk melihat revisi yang diberikan oleh dosen pembimbing, serta memberikan revisi yang telah diperbaiki kepada dosen pembimbing. Selain itu, aplikasi ini dilengkapi dengan fitur tambahan, seperti kemampuan untuk menyalin data ke *clipboard*, ekspor data dalam format CSV, *Excel*, dan PDF, yang memudahkan *user*, terutama admin dalam pengelolaan dan analisis data.

4. Pengujian fungsionalitas dan keamanan aplikasi dilakukan menggunakan *postman*, *user testing*, *browser*, dan OWASP ZAP. Hasilnya menunjukkan bahwa aplikasi berjalan sesuai harapan dan memenuhi standar keamanan tingkat 2 (medium). Pengujian dilakukan 5 kali percobaan dengan *postman*, 1 kali percobaan *user testing*, 6 kali percobaan *browser*, dan 3 kali percobaan dengan OWASP ZAP. Aplikasi telah diuji secara menyeluruh dan dapat berfungsi baik dalam mengelola data. Pengujian keamanan juga dilakukan untuk memastikan sistem tahan terhadap ancaman dan serangan, menjaga kestabilan operasional serta keamanan data secara keseluruhan.
5. Aplikasi ini menyajikan informasi tahapan proses pelaksanaan skripsi, sekaligus memantau status mahasiswa yang sedang menempuh skripsi. Melalui aplikasi ini, program studi dapat memantau kemajuan mahasiswa secara efisien dan langsung secara *online*. Aplikasi ini akan membantu meningkatkan efisiensi dalam pemantauan progres skripsi mahasiswa secara keseluruhan.

5.2 Saran

Pembuatan aplikasi *monitoring* skripsi, terdapat beberapa saran sebagai masukan agar dapat dilakukan pengembangan selanjutnya, yaitu:

1. Penambahan fitur pesan atau ide skripsi dari dosen atau koordinator skripsi untuk memudahkan mahasiswa yang kesulitan dalam mencari topik skripsi.
2. Status lulus tepat waktu, untuk memastikan keakuratan informasi yang disajikan kepada mahasiswa dan pihak terkait, serta membantu mahasiswa untuk mengikuti perkembangan skripsi dengan lebih baik.

DAFTAR PUSTAKA

- Ahmad Martani, Saripuddin M, & Nurul Ikhsan. (2022). Rancang Bangun Website Company Profile Berbasis Framework Bootstrap dan Framework Codeigniter Pada Yayasan Khalifah Cendekia Mandiri. *Jurnal Multidisiplin Madani*, 2(6), 2895–2912.
- Alcantara, P., Campbell, R., & Carolina, N. (2021). System, Method, and GUI Client For Publishing Node Package Manager (NPM) Packages On NPM Repository. *United States Patent*, 1(16), 1–7.
- Aulia, S. C. I. (2022). Pemanfaatan Uml (Unified Modeling Language) dalam Perancangan Sistem Informasi Rekam Medis Sederhana Pada Kegiatan Posbindu Ptm. *Jurnal Ilmiah Sains Dan Teknologi*, 6(1), 38–44.
- Batubara, T. P., Efendi, S., & Nababan, E. B. (2021). Analysis Performance Bcrypt Algorithm to Improve Password Security from Brute Force. *Journal of Physics: Conference Series (ICOSTA)*, 1811(1), 1–7.
- Bloom, N., & Reenen, J. Van. (2023). HTML The Ultimate Guide. In *NBER Working Papers* (First edit). Taylor & Francis Group.
- Brata, D. W. (2015). Perancangan Sistem KHS Mobile di STMIK ASIA Malang Menggunakan Android Programming dan JSON. *Jurnal Ilmiah Teknologi Dan Informasi ASIA (JITIKA)*, 9(2), 30–39.
- Chandra, J. V., & Tan, R. (2020). Rancang Bangun Web API untuk Akses Informasi Kerja Sama Pendidikan Tinggi. *Strategy Journal*, 2(2), 510–520.
- Devianty, D., & Nur Ibrahim, R. (2021). Perancangan Sistem E-Arsip Menggunakan Subject Filing System Berbasis Framework Codeigniter (Studi Kasus Stmik Mardira Indonesia). *Jurnal Computech & Bisnis*, 15(2), 100–107.
- Dewi, E. K., Rahmalisa, U., & Febriani, A. (2019). Aplikasi Kuesioner Pra Skrining Perkembangan Anak Berbasis Android di Hompimpa Center Bengkalis. *Jurnal Informatika Polinema (JIP)*, 6(1), 71–80.
- Fajrin, R. (2017). Pengembangan Sistem Informasi Geografis Berbasis Node.JS untuk Pemetaan Mesin dan Tracking Engineer dengan Pemanfaatan Geolocation pada PT IBM Indonesia. *Jurnal Informatika*, 11(2), 40–47.

- Giffary, R. S., & Ramadhani, E. (2022). Implementasi Bcrypt dengan SHA-256 Pada Password Pengguna Aplikasi Golek Kost. *Jurnal Sistem Komputer Dan Informatika (JSON)*, 3(4), 543.
- Hasanuddin, Asgar, H., & Hartono, B. (2022). Rancang Bangun Rest Api Aplikasi Weshare Sebagai Upaya Mempermudah Pelayanan Donasi Kemanusiaan. *Jurnal Ilmu Pengetahuan Dan Teknologi Informatika (JINTEKS)*, 4(1), 8–14.
- Iswardhana, M. R., & Widiono, S. (2021). *Diplomasi Siber dan Teknologi Mobile Pada Multi disiplin*. Kemitraan untuk Tindakan Pendidikan Masyarakat.
- Kore, P. (2022). API Testing Using Postman Tool. *International Journal For Research (IJRASET)*, 10(12), 841–843.
- Kurniawan, A., & Rosa, S. E. (2020). Rancang Bangun Aplikasi Ujian Online Madrasah Aliyah Negeri Surabaya Berbasis Website dengan Framework Laravel. *Jurnal Manajemen Informasi*, 11(1), 1–11.
- Kurniawan, I., Humaira, & Rozi, F. (2020). REST API Menggunakan NodeJS pada Aplikasi Transaksi Jasa Elektronik Berbasis Android. *Jurnal Ilmiah Teknologi Sistem Informasi (JITSI)*, 1(4), 127–132.
- Matuszek, D. (2023). *Quick Javascript* (First edit). A Chapman & Hall Book.
- Mubariz, A., Nur, D., Tungadi, E., & Utomo, M. N. Y. (2020). Perancangan Back-End Server Menggunakan Arsitektur Rest dan Platform Node.JS (Studi Kasus : Sistem Pendaftaran Ujian Masuk Politeknik Negeri Ujung Pandang). *Seminar Nasional Teknik Elektro Dan Informatika (SNTEI)*, 72–77.
- Mulyani, F., & Haliza, N. (2021). Analisis Perkembangan Ilmu Pengetahuan dan Teknologi (Iptek) dalam Pendidikan. *Jurnal Pendidikan Dan Konseling (JPDK)*, 3(1), 101–109.
- Munawar, G. (2018). Analisa Penerapan MEAN Stack Dalam Pengembangan Web Berbasis Deklaratif. *9th Industrial Research Workshop and National Seminar*, 251–257.
- Munazilin, A., Homaidi, A., & Ghofur, A. (2017). *Monitoring Keaktifan Tenaga Pengajar Bidang Pendidikan Pp. Salafiyah Syafi'iyah Sukorejo Situbondo*. *Jurnal Ilmiah Informatika*, 2(2), 121–129.
- Muri, M. F. A., Utomo, H. S., & Sayyidati, R. (2019). Search Engine Get Application Programming Interface. *Jurnal Sains Dan Informatika*, 5(2), 88–97.
- Nababan, A. A., Jannah, M., & Sianturi, F. A. (2022). Pelatihan Sistem Informasi Penelitian Dan Pengabdian Kepada Masyarakat (SIM-PPM) STMIK Pelita Nusantara. *Jurnal Pengabdian Kepada Masyarakat Nusantara*, 3(1), 241–251.

- Nasution. (2021). *Implementation of MongoDB, Express Js, React Js and Node Js (Mern) on the Development of Online Forms, Quizzes and Surveys Applications*.
- Nugraha, Y., Muslihah, N. N., Sary, Y. S., & Herman, S. (2022). Development (Frontend) of the Mobile-based Beliin Yuks Application using the Flutter Framework. *Research in Information Systems and Technology (RISTEC)*, 3(1), 52–60.
- Prasetyo, B. D., Witanti, W., Sabrina, P. N., Sains, F., Jenderal, U., Yani, A., & Cimahi, K. (2020). Sistem *Monitoring* Bimbingan Mahasiswa Dengan Menggunakan Metode Service Quality. *Prosiding Seminar Nasional Sistem Sistem Informasi Dan Teknologi (SISFOTEK)*, 118–122.
- Prawastiyo, C. A., & Hermawan, I. (2020). Pengembangan Front-End Website Perpustakaan Politeknik Negeri Jakarta dengan menggunakan Metode User Centered Design. *Jurnal Teknologi Terpadu*, 6(2), 89–95.
- Pressman. (2012). Book review: Software Engineering: a Practitioner's Approach. In *Software Engineering Journal* (Vol. 10, Issue 6).
- Rahardja, U. (2022). Meningkatkan Kualitas Sumber Daya Manusia Dengan Sistem Pengembangan Fundamental Agile. *ADI Bisnis Digital Interdisiplin Jurnal*, 3(1), 63–68.
- Rompis, A. C., & Aji, R. F. (2018). Perbandingan Performa Kinerja Node.js, PHP, dan Python dalam Aplikasi REST. *CogITO Smart Journal*, 4(1), 171–187.
- Sari, A. S., & Hidayat, R. (2022). Designing Website Vaccine Booking System Using Golang Programming Language and Framework React JS. *Journal of Information System, Informatics and Computing*, 6(1), 22–39.
- Sari, N. M., Tua, L. M., & Krisnanik, E. (2019). Sistem Informasi *Monitoring* Pembimbingan Skripsi / Tugas Akhir (SIMP-S / TA) Berbasis Android. *Seminar Nasional Informatika, Sistem Informasi Dan Keamanan Siber (SEINASI-KESI)*, 1–10.
- Satria, B., Kusyanti, A., & Yahya, W. (2018). Implementation of Blake2s Algorithm on JSON Web Token (JWT) as Hashing Algorithm for REST-API Service Authentication Mechanism. *Journal of Information Technology and Computer Science Development (J-PTIHK) Universitas Brawijaya*, 2(12), 6269–6276.
- Saundariya, K., Abirami, M., Senthil, K. R., Prabakaran, D., Srimathi, B., & Nagarajan, G. (2021). Webapp service for booking handyman using MongoDB, express JS, React JS, node JS. *Proceeding International Conference on Signal Processing and Communication (ICPSC)*, 13, 180–183.

- Shah, H., & Soomro, T. R. (2017). Node.js Challenges in Implementation. *Global Journals of Computer Science and Technology: E Network, Web & Security*, 17(2), 73–83.
- Sundaramoorthy, S. (2022). *UML Diagramming A Case Study Approach*. CRC Press Taylor & Francis Group.
- Syaftiaan, B., Safira, N. F., Rizky, F., & Telkom, U. (2021). Rancang Bangun Backend Aplikasi Jobbie: Pencarian Dan Penyedia. *E-Proceeding of Engineering*, 8(6), 12441–12448.
- Tedyyana, A., Fauzi, M., & Ratnawati, F. (2021). Revamp Keamanan Web Service Milik PT XYZ Menggunakan REST API. *Jurnal Teknologi Informasi Dan Komunikasi*, 12(1), 1–10.
- Uzayr, S. bin. (2023). *Mastering HTML a Beginner's Guide* (First Edit, Vol. 1). Mastering Computer Science.
- Widhyaestoeti, D., Iqram, S., Mutiyah, S. N., & Khairunnisa, Y. (2021). Black Box Testing Equivalence Partitions untuk Pengujian Front-End Pada Sistem Akademik Sitoda. *Jurnal Ilmiah Teknologi Informasi Terapan*, 7(3), 211–216.
- Wintana, D., Pribadi, D., & Nurhadi, M. Y. (2022). Analisis Perbandingan Efektifitas White-Box Testing dan Black-Box Testing. *Jurnal Larik: Ladang Artikel Ilmu Komputer*, 2(1), 8–16.
- Zulkarnaini, I. (2020). Sistem Informasi Pengajuan Judul Skripsi Pada Fakultas Ilmu Komputer Universitas Almuslim. *Jurnal Ilmiah Sains, Teknologi, Ekonomi, Sosial Dan Budaya*, 4(4), 9–14.