

**PENGEMBANGAN SISTEM PENJAMIN MUTU JURUSAN ILMU
KOMPUTER MODUL MANAJEMEN *USER* DAN *FILE* LAPORAN**

(Skripsi)

Oleh

**DHIAURRAHMAN RAZIQ RAMADHAN
NPM 2117051048**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025**

**PENGEMBANGAN SISTEM PENJAMIN MUTU JURUSAN ILMU
KOMPUTER MODUL MANAJEMEN *USER* DAN *FILE* LAPORAN**

Oleh

DHIAURRAHMAN RAZIQ RAMADHAN

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA ILMU KOMPUTER**

Pada

**Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Lampung**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025**

ABSTRAK

PENGEMBANGAN SISTEM PENJAMIN MUTU JURUSAN ILMU KOMPUTER MODUL MANAJEMEN *USER* DAN *FILE* LAPORAN

Oleh

DHIAURRAHMAN RAZIQ RAMADHAN

Penelitian ini berfokus pada pengembangan Sistem Penjaminan Mutu untuk Jurusan Ilmu Komputer Universitas Lampung, khususnya peningkatan modul manajemen pengguna dan pembuatan *file* laporan. Sistem sebelumnya memiliki keterbatasan dalam pembagian peran dan tidak mendukung pembuatan laporan yang dapat diunduh. Tujuan utama adalah mengembangkan modul manajemen pengguna berbasis *Role-Based Access Control* (RBAC) untuk berbagai peran hierarkis, serta menambahkan fitur laporan PDF untuk visualisasi Capaian Pembelajaran Lulusan (CPL) dan Capaian Pembelajaran Mata Kuliah (CPMK). Metode *Rapid Application Development* (RAD) digunakan dalam siklus iteratif perencanaan, desain, pengembangan *live prototype*, dan implementasi dengan Laravel. Sistem yang dikembangkan mampu mengatasi keterbatasan awal dengan menyediakan manajemen *multi-role* dan pelaporan yang fleksibel. Pengujian *black-box* menunjukkan semua fungsi berjalan sesuai kebutuhan, serta pengujian non-fungsional melalui Apache JMeter menunjukkan sistem memiliki performa respons yang stabil pada beban pengguna yang bervariasi. Sistem ini mendukung evaluasi mutu yang terdokumentasi dan sesuai struktur organisasi pendidikan tinggi.

Kata Kunci : CPL dan CPMK; Laravel; Multi-otoritas; Pendidikan Tinggi; *Rapid Application Development*.

ABSTRACT

DEVELOPMENT OF A QUALITY ASSURANCE SYSTEM FOR THE COMPUTER SCIENCE DEPARTMENT USER AND FILE MANAGEMENT MODULE REPORT

By

DHIAURRAHMAN RAZIQ RAMADHAN

This study focuses on the development of a Quality Assurance System for the Computer Science Department at the University of Lampung, particularly improving the user management and report generation modules. The previous system lacked detailed role differentiation and did not support downloadable report generation. The main objectives were to implement a Role-Based Access Control (RBAC) user management module for various hierarchical roles and to add PDF reporting features for visualizing Graduate Learning Outcomes (CPL) and Course Learning Outcomes (CPMK). The development followed the Rapid Application Development (RAD) methodology with iterative planning, design, live prototyping, and implementation using the Laravel framework. The resulting system addressed prior limitations by enabling flexible multi-role management and customizable reporting. Functional testing confirmed expected behavior, while non-functional testing with Apache JMeter showed stable system performance under varying user loads. This system supports documented quality assurance aligned with institutional structures and facilitates learning outcome evaluation.

Keywords : Course and Graduate Outcomes; Higher Education; Laravel; Multi-authority; Rapid Application Development.

Judul Skripsi

: **PENGEMBANGAN SISTEM
PENJAMIN MUTU JURUSAN ILMU
KOMPUTER MODUL MANAJEMEN
USER DAN FILE LAPORAN**

Nama Mahasiswa

: **Dhiaurrahman Raziq Ramadhan**

Nomor Pokok Mahasiswa

: 2117051048

Program Studi

: Ilmu Komputer

Fakultas

: Matematika dan Ilmu Pengetahuan Alam



1. Komisi Pembimbing

Tristiyanto, S.Kom., M.I.S., Ph.D

NIP. 19810414 200501 1 001

2. Ketua Jurusan Ilmu Komputer

Dwi Sakethi, S.Si., M.Kom.

NIP. 19680611 199802 1 001

MENGESAHKAN

1. Tim Penguji

Ketua : Tristiyanto, S.Kom., M.I.S., Ph.D

Sekretaris : Dwi Sakethi, S.Si, M.Kom.

Penguji Utama : Didik Kurniawan, S.Si., M.T.

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



Dr. Eng Heri Satria, S.Si., M.Si.

NIP. 19711001 200501 1 002

Tanggal Lulus Ujian Skripsi: 18 Juni 2025

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Dhiaurrahman Raziq Ramadhan

NPM : 2117051048

Dengan ini menyatakan bahwa skripsi saya yang berjudul **“Pengembangan Sistem Penjamin Mutu Jurusan Ilmu Komputer Modul Manajemen User dan File Laporan”** merupakan karya saya sendiri dan bukan karya orang lain. Semua tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya tulis ilmiah Universitas Lampung. Apabila di kemudian hari terbukti skripsi saya merupakan hasil jiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang saya terima.

Bandar Lampung, 20 Juni 2025

Penulis,



Dhiaurrahman Raziq Ramadhan

NPM. 2117051048

RIWAYAT HIDUP



Lahir di Tangerang pada 8 November 2003 sebagai anak kedua dari pasangan Bapak Agung Hartadi dan Ibu Muliq Amindari. Penulis telah menyelesaikan pendidikan formal di SD Wijaya Kusuma pada tahun 2015. Kemudian SMPN 161 Jakarta pada tahun 2018 dan SMAN 29 Jakarta pada tahun 2021. Penulis terdaftar sebagai mahasiswa Program Studi S1 Ilmu Komputer, Universitas Lampung, melalui jalur Seleksi Bersama Masuk Perguruan Tinggi Negeri (SBMPTN) pada tahun 2021.

Selama menjadi mahasiswa di Ilmu Komputer, penulis aktif di dalam berbagai kegiatan baik di dalam maupun di luar Universitas Lampung. Kegiatan yang dilakukan adalah sebagai berikut:

1. Menjadi Asisten Dosen untuk mata kuliah Dasar-dasar Pemrograman di Jurusan Ilmu Komputer pada tahun 2022.
2. Menjadi Asisten Dosen untuk mata kuliah Pemrograman Terstruktur di Jurusan Ilmu Komputer pada tahun 2023.
3. Menjadi Asisten Dosen untuk mata kuliah Basis Data di Jurusan Ilmu Komputer pada tahun 2023.
4. Mengikuti program Magang MBKM *Batch* 6 di Yayasan Kazeto Putra Perkasa pada divisi *Programmer* pada tahun 2024.
5. Melaksanakan Kuliah Kerja Nyata (KKN) Periode II Universitas Lampung di Desa Way Mili pada tahun 2024.

PERSEMBAHAN

Segala puji saya panjatkan kepada Allah Subhanahu wa Ta'ala segala limpahan rahmat dan karunia-Nya. Shalawat serta salam semoga senantiasa tercurah kepada Nabi Muhammad Shallallahu'alaihi wa sallam, sehingga penulis dapat menyelesaikan skripsi ini.

Skripsi ini saya persembahkan kepada:

Kedua orang tua saya tercinta, atas cinta, doa, dan dukungan yang menjadi fondasi keberhasilan ini dan seluruh Keluarga Besar Ilmu Komputer 2021 Universitas Lampung yang telah bersama dalam proses penyelesaian skripsi ini. Serta untuk diri saya sendiri, Dhiaurrahman Raziq Ramadhan, atas ketekunan dan semangat yang tak pernah padam dalam menyelesaikan karya ini.

MOTTO

“...Sesungguhnya Allah beserta orang-orang yang sabar.”

(QS. Al-Baqarah: 153)

“Success is the sum of small efforts, repeated day in and day out.”

(Robert Collier)

“Knowledge is the best investment for the future.”

(Dhiaurrahman Raziq Ramadhan)

SANWACANA

Puji syukur kehadiran Allah Subhanahu wa Ta'ala, yang telah memberikan rahmat dan hidayah-Nya kepada penulis, sehingga penulis dapat menyelesaikan skripsi ini dengan tepat waktu. Shalawat serta salam penulis sanjungkan kepada Nabi Muhammad Shallallahu'alaihi wa sallam yang penulis harapkan syafaatnya di hari akhir kelak.

Skripsi yang berjudul **“Pengembangan Sistem Penjamin Mutu Jurusan Ilmu Komputer Modul Manajemen *User* dan *File* Laporan”** ini disusun sebagai salah satu syarat untuk memperoleh gelar sarjana pada Program Studi S1 Ilmu Komputer di Universitas Lampung.

Pada kesempatan ini penulis mengucapkan terima kasih kepada semua pihak yang telah berkontribusi dalam penyusunan skripsi ini, yaitu:

1. Kedua orang tua penulis, Bapak Agung Hartadi dan Ibu Muliq Amindari selaku kedua orang tua serta kakak saya Fadhillah Grandy Radiant, yang senantiasa memberikan semangat, dukungan, dan doa serta membantu dalam segala hal dalam penyelesaian skripsi ini.
2. Bapak Dr. Eng. Heri Satria, S.Si., M.Si. selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
3. Bapak Dwi Sakethi, S.Si., M.Kom. selaku Dosen Pembahas Utama sekaligus Ketua Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung yang telah memberikan banyak masukan serta kritik membangun demi penyempurnaan skripsi ini.
4. Bapak Rico Andrian, S.Si., M.Kom. selaku Dosen Pembimbing Akademik penulis.

5. Bapak Tristiyanto, S.Kom., M.I.S., Ph.D selaku Dosen Pembimbing Utama yang telah membimbing dan memberikan banyak arahan kepada penulis sehingga penelitian yang penulis lakukan dapat berjalan dengan baik.
6. Bapak Didik Kurniawan, S.Si., M.T. selaku Dosen Pembahas Kedua yang telah memberikan kritik dan saran yang bermanfaat sehingga penulisan skripsi ini dapat diselesaikan dengan baik.
7. Bapak/Ibu Dosen dan seluruh staf Jurusan Ilmu Komputer FMIPA Universitas Lampung yang telah memberikan ilmu dan membantu dalam berbagai urusan akademik maupun administratif.
8. Seluruh teman-teman seperjuangan di Jurusan Ilmu Komputer Universitas Lampung angkatan 2021 yang telah menjadi keluarga dan saling mendukung selama masa perkuliahan.
9. Untuk diri penulis, Dhiaurrahman Raziq Ramadhan, terima kasih telah bertahan dan berjuang dalam setiap proses, hingga berhasil menyelesaikan pendidikan di Program Studi S1 Ilmu Komputer dengan tepat waktu.

Dalam proses penyusunan skripsi ini penulis menyadari bahwa terdapat banyak kekurangan disebabkan oleh keterbatasan pengetahuan serta pengalaman. Oleh karena itu penulis mengharapkan kritik dan saran yang membangun demi perbaikan di masa mendatang. Semoga skripsi ini dapat memberikan manfaat bagi semua pihak.

Bandar Lampung, 20 Juni 2025

Dhiaurrahman Raziq Ramadhan
NPM. 2117051048

DAFTAR ISI

	Halaman
DAFTAR ISI.....	XIII
DAFTAR TABEL	XVI
DAFTAR GAMBAR.....	XVIII
DAFTAR PSEUDOCODE	XXIV
I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Manfaat	4
1.5 Batasan Masalah	4
II. TINJAUAN PUSTAKA.....	5
2.1 Penelitian Terdahulu	5
2.2 Manajemen <i>User</i>	7
2.3 Laporan	7
2.4 PHP (<i>Hypertext Preprocessor</i>)	8
2.5 Laragon	8
2.6 Laravel	9
2.7 MySQL	10
2.8 <i>Unified Modeling Language</i> (UML).....	11
2.8.1 <i>Use Case Diagram</i>	11
2.8.2 Activity Diagram	13

2.8.3 Class Diagram	14
2.8.4 Deployment Diagram.....	15
2.9 Entity Relationship Diagram (ERD)	15
2.10 Software Development Life Cycle (SDLC).....	16
2.11 Rapid Application Development (RAD).....	16
2.12 Black-Box Testing.....	19
2.13 Apache JMeter.....	20
III. METODOLOGI PENELITIAN.....	21
3.1 Waktu dan Tempat Penelitian.....	21
3.1.1 Waktu Penelitian.....	21
3.1.2 Tempat Penelitian	22
3.2 Tahapan Penelitian.....	22
3.3 Identifikasi Masalah.....	24
3.4 Studi Literatur	24
3.5 Perencanaan Kebutuhan (<i>Rapid Application Development</i>).....	25
3.5.1 Perangkat Penelitian.....	25
3.5.2 Ringkasan Eksekutif	26
3.5.3 Kebutuhan Bisnis	27
3.5.4 Kebutuhan Fungsional dan Non-Fungsional	27
3.5.5 Nilai Bisnis.....	30
3.5.6 Definisi dan Strategi Analisis Kebutuhan.....	31
3.5.7 <i>Product Backlog</i>	34
3.5.8 <i>Use Case Diagram</i>	41
3.5.9 Spesifikasi Perangkat Lunak.....	65
3.5.10 Skenario Pengujian Fungsional.....	65
3.5.11 Skenario Pengujian Non-Fungsional	76

3.5.12 Validasi <i>Prototype</i>	77
3.5.13 <i>Activity Diagram</i>	78
3.5.14 <i>Class Diagram</i>	108
3.5.15 <i>Entity Relationship Diagram</i> (ERD).....	110
3.5.16 Arsitektur Manajemen Peran dan Keamanan	112
3.5.17 <i>Deployment Diagram</i>	113
3.5.18 Rancangan Tampilan Antarmuka Pengguna.....	114
IV. HASIL DAN PEMBAHASAN.....	135
4.1 Hasil	135
4.2 Validasi <i>Prototype</i>	136
4.3 Desain Sistem (<i>Rapid Application Development</i>).....	136
4.3.1 <i>User Design 1/Input 1</i>	136
4.3.2 <i>User Design 2/Input 2</i>	143
4.3.3 <i>User Design 3/Input 3</i>	150
4.3.4 <i>User Design 4/Input 4</i>	157
4.4 Pengembangan (<i>Rapid Application Development</i>)	168
4.5 Implementasi (<i>Rapid Application Development</i>).....	213
4.6 Pembahasan.....	214
4.6.1 Pengujian Fungsional menggunakan <i>Black-box</i>	214
4.6.2 Pengujian Non-Fungsional.....	226
V. SIMPULAN DAN SARAN.....	230
5.1 Kesimpulan	230
5.2 Saran.....	230
DAFTAR PUSTAKA.....	231

DAFTAR TABEL

Tabel	Halaman
1. Tabel 2.1 Penelitian terdahulu.....	5
2. Tabel 2.2 Komponen <i>Use Case</i>	12
3. Tabel 2.3 Komponen <i>Activity Diagram</i>	14
4. Tabel 2.4 Komponen <i>Entity Relationship Diagram</i> (ERD)	15
5. Tabel 3.1 Rencana Penelitian.....	21
6. Tabel 3.2 Tabel Kebutuhan Fungsional	28
7. Tabel 3.3 Tabel Kebutuhan Non Fungsional	29
8. Tabel 3.4 Matriks Kebutuhan Fungsional <i>As-Is</i> dan <i>To-Be</i>	31
9. Tabel 3.5 Matriks Kebutuhan Non-Fungsional <i>As-Is</i> dan <i>To-Be</i>	32
10. Tabel 3.6 <i>Product Backlog</i>	34
11. Tabel 3.7 Deskripsi <i>Use Case</i> Melihat Berita.....	44
12. Tabel 3.8 Deskripsi <i>Use Case</i> Menambahkan Registrasi Universitas	44
13. Tabel 3.9 Deskripsi <i>Use Case</i> Melakukan <i>Login</i>	45
14. Tabel 3.10 Deskripsi <i>Use Case</i> Mengelola Berita	46
15. Tabel 3.11 Deskripsi <i>Use Case</i> Mengelola Registrasi Universitas.....	47
16. Tabel 3.12 Deskripsi <i>Use Case</i> Mengelola <i>User</i>	48
17. Tabel 3.13 Deskripsi <i>Use Case</i> Mengelola Tema Universitas.....	49
18. Tabel 3.14 Deskripsi <i>Use Case</i> Mengelola Profil.....	50
19. Tabel 3.15 Deskripsi <i>Use Case</i> Mengelola Prodi	51
20. Tabel 3.16 Deskripsi <i>Use Case</i> Mengelola Fakultas	52
21. Tabel 3.17 Deskripsi <i>Use Case</i> Mengelola Universitas.....	53
22. Tabel 3.18 Deskripsi <i>Use Case</i> Melihat Mahasiswa.....	54
23. Tabel 3.19 Deskripsi <i>Use Case</i> Mengelola Mahasiswa	54
24. Tabel 3.20 Deskripsi <i>Use Case</i> Melihat Visualisasi CPL per Mahasiswa.....	55
25. Tabel 3.21 Deskripsi <i>Use Case</i> Mencetak Laporan Visualisasi CPL per Mahasiswa.....	56

26. Tabel 3.22 Deskripsi <i>Use Case</i> Melihat Visualisasi CPMK per Mahasiswa..	57
27. Tabel 3.23 Deskripsi <i>Use Case</i> Mencetak Laporan Visualisasi CPMK per Mahasiswa.....	58
28. Tabel 3.24 Deskripsi <i>Use Case</i> Melihat Visualisasi CPL per Angkatan	59
29. Tabel 3.25 Deskripsi <i>Use Case</i> Mencetak Laporan Visualisasi CPL per Angkatan	59
30. Tabel 3.26 Deskripsi <i>Use Case</i> Melihat Visualisasi CPMK per Angkatan	60
31. Tabel 3.27 Deskripsi <i>Use Case</i> Mencetak Laporan Visualisasi CPMK per Angkatan	61
32. Tabel 3.28 Deskripsi <i>Use Case</i> Melihat Visualisasi CPMK per Mata Kuliah	62
33. Tabel 3.29 Deskripsi <i>Use Case</i> Mencetak Laporan Visualisasi CPMK per Mata Kuliah.....	62
34. Tabel 3.30 Deskripsi <i>Use Case</i> Melihat Visualisasi CPMK per Mahasiswa..	63
35. Tabel 3.31 Deskripsi <i>Use Case</i> Mencetak Laporan Visualisasi CPMK per Mahasiswa.....	64
36. Tabel 3.32 Spesifikasi Perangkat Keras dan Perangkat Lunak.....	65
37. Tabel 3.33 Skenario Pengujian Sistem	65
38. Tabel 3.34 Skenario Pengujian Non-Fungsional	76
39. Tabel 4.1 Hasil pengujian <i>black-box</i>	214
40. Tabel 4.2 Langkah Pengujian menggunakan Jmeter.....	228

DAFTAR GAMBAR

Gambar	Halaman
1. Gambar 2.1 Antarmuka Laragon.....	9
2. Gambar 2.2 Contoh Kode Laravel.	10
3. Gambar 2.3 MySQL pada PHPMyAdmin.	11
4. Gambar 2.4 Tahapan <i>Rapid Application Development</i>	19
5. Gambar 3.1 Tahapan Penelitian.	22
6. Gambar 3.2 <i>Use Case Diagram</i> Global Sistem Penjamin Mutu Jurusan Ilmu Komputer.	42
7. Gambar 3.3 <i>Use Case Diagram</i> Sistem Penjamin Mutu Jurusan Ilmu Komputer.	43
8. Gambar 3.4 <i>Activity Diagram</i> Melihat Berita.	78
9. Gambar 3.5 <i>Activity Diagram</i> Menambahkan Registrasi Universitas	79
10. Gambar 3.6 <i>Activity Diagram</i> Melakukan <i>Login</i>	80
11. Gambar 3.7 <i>Activity Diagram</i> mengelola Berita.....	81
12. Gambar 3.8 <i>Activity Diagram</i> Mengelola Registrasi Universitas.	83
13. Gambar 3.9 <i>Activity Diagram</i> Mengelola <i>User</i>	84
14. Gambar 3.10 <i>Activity Diagram</i> Mengelola Tema Universitas.....	85
15. Gambar 3.11 <i>Activity Diagram</i> Mengelola Profil.	87
16. Gambar 3.12 <i>Activity Diagram</i> Mengelola Prodi.	88
17. Gambar 3.13 <i>Activity Diagram</i> Mengelola Fakultas.....	89
18. Gambar 3.14 <i>Activity Diagram</i> Mengelola Universitas.....	91
19. Gambar 3.15 <i>Activity Diagram</i> Melihat Mahasiswa.....	91
20. Gambar 3.16 <i>Activity Diagram</i> Mengelola Mahasiswa.	92
21. Gambar 3.17 <i>Activity Diagram</i> Melihat Visualisasi CPL per Mahasiswa... 93	93
22. Gambar 3.18 <i>Activity Diagram</i> Mencetak Laporan Visualisasi CPL Mahasiswa per Mahasiswa.....	94

23.	Gambar 3.19 <i>Activity Diagram</i> Melihat Visualisasi CPMK per Mahasiswa.	96
24.	Gambar 3.20 <i>Activity Diagram</i> Mencetak Laporan Visualisasi CPMK per Mahasiswa.	97
25.	Gambar 3.21 <i>Activity Diagram</i> Melihat Visualisasi CPL per Angkatan	98
26.	Gambar 3.22 <i>Activity Diagram</i> mencetak laporan CPL per Mahasiswa.	99
27.	Gambar 3.23 <i>Activity Diagram</i> melihat data CPMK per Mahasiswa.	100
28.	Gambar 3.24 <i>Activity Diagram</i> Mencetak Laporan Visualisasi CPMK per Angkatan.	102
29.	Gambar 3.25 <i>Activity Diagram</i> Melihat Visualisasi CPMK per Mata Kuliah.	103
30.	Gambar 3.26 <i>Activity Diagram</i> Mencetak Laporan Visualisasi CPMK per Mata Kuliah.	104
31.	Gambar 3.27 <i>Activity Diagram</i> melihat data CPMK per Angkatan.....	105
32.	Gambar 3.28 <i>Activity Diagram</i> Mencetak Laporan Visualisasi CPMK per Mahasiswa.	107
33.	Gambar 3.29 <i>Class Diagram</i> Pengembangan Sistem Penjamin Mutu Jurusan Ilmu Komputer Modul Manajemen <i>User</i> dan <i>File</i> Laporan.	109
34.	Gambar 3.30 ERD Pengembangan Sistem Penjamin Mutu Modul Manajemen <i>User</i> dan <i>File</i> Laporan.	111
35.	Gambar 3.31 Arsitektur Basis Data untuk Manajemen Peran (RBAC).....	112
36.	Gambar 3.32 <i>Deployment Diagram</i> Sistem Manajemen <i>User</i> dan <i>File</i> Laporan.	113
37.	Gambar 3.33 Rancangan tampilan Melihat Berita.....	114
38.	Gambar 3.34 Rancangan tampilan Menambahkan Registrasi Universitas.	115
39.	Gambar 3.35 Rancangan tampilan Melakukan <i>Login</i>	116
40.	Gambar 3.36 Rancangan tampilan Mengelola Berita bagian Melihat Berita.	117
41.	Gambar 3.37 Rancangan tampilan Mengelola Berita bagian <i>form</i> Tambah Berita.	117
42.	Gambar 3.38 Rancangan tampilan Mengelola Berita bagian <i>form</i> Ubah Berita.	118

43.	Gambar 3.39 Rancangan tampilan Mengelola Registrasi Universitas bagian Melihat Daftar Registrasi Universitas.....	119
44.	Gambar 3.40 Rancangan tampilan Mengelola Registrasi Universitas bagian Menyetujui Registrasi Universitas.....	119
45.	Gambar 3.41 Rancangan tampilan Mengelola Registrasi Universitas bagian Menolak Registrasi Universitas.	119
46.	Gambar 3.42 Rancangan tampilan Mengelola <i>User</i> bagian Melihat <i>User</i> .	120
47.	Gambar 3.43 Rancangan tampilan Mengelola <i>User</i> bagian <i>form</i> Tambah <i>User</i>	121
48.	Gambar 3.44 Rancangan tampilan Mengelola <i>User</i> bagian <i>form</i> Ubah <i>User</i>	121
49.	Gambar 3.45 Rancangan tampilan Mengelola Tema Universitas.....	122
50.	Gambar 3.46 Rancangan tampilan Mengelola Profil.....	123
51.	Gambar 3.47 Rancangan tampilan Mengelola Prodi bagian Melihat Prodi.	123
52.	Gambar 3.48 Rancangan tampilan Mengelola Prodi bagian <i>form</i> Tambah Prodi.	124
53.	Gambar 3.49 Rancangan tampilan Mengelola Fakultas bagian Melihat Fakultas.	124
54.	Gambar 3.50 Rancangan tampilan Mengelola Fakultas bagian <i>form</i> Tambah Fakultas.	124
55.	Gambar 3.51 Rancangan tampilan Mengelola Universitas bagian Melihat Universitas.	125
56.	Gambar 3.52 Rancangan tampilan Mengelola Fakultas bagian <i>form</i> Tambah Fakultas.	125
57.	Gambar 3.53 Rancangan tampilan Mengelola Mahasiswa bagian Melihat Mahasiswa.....	126
58.	Gambar 3.54 Rancangan tampilan Mengelola Mahasiswa bagian form Tambah Mahasiswa.....	126
59.	Gambar 3.55 Rancangan tampilan Mengelola Mahasiswa bagian <i>form</i> Ubah Mahasiswa.....	126
60.	Gambar 3.56 Rancangan Tampilan Visualisasi CPL per Mahasiswa.....	128

61.	Gambar 3.57 Rancangan Tampilan Visualisasi CPMK per Mahasiswa....	129
62.	Gambar 3.58 Rancangan Tampilan Visualisasi CPL per Angkatan.	131
63.	Gambar 3.59 Rancangan Tampilan Visualisasi CPMK per Angkatan.	132
64.	Gambar 3.60 Rancangan Tampilan Visualisasi CPMK per Mata Kuliah..	134
65.	Gambar 4.1 <i>Prototype</i> Melakukan <i>Login</i>	138
66.	Gambar 4.2 <i>Prototype</i> Melihat <i>User</i>	139
67.	Gambar 4.3 <i>Prototype</i> Menambahkan <i>User</i>	140
68.	Gambar 4.4 <i>Prototype</i> Mengubah <i>User</i>	140
69.	Gambar 4.5 <i>Prototype</i> Menghapus <i>User</i>	141
70.	Gambar 4.6 <i>Prototype</i> Mengelola Profil.....	141
71.	Gambar 4.7 <i>Prototype</i> Mengelola Tema Universitas.....	142
72.	Gambar 4.8 <i>Prototype</i> Melihat Berita.	144
73.	Gambar 4.9 <i>Prototype</i> Melihat Berita.	145
74.	Gambar 4.10 <i>Prototype</i> Menambahkan Berita.....	145
75.	Gambar 4.11 <i>Prototype</i> Mengubah Berita.	146
76.	Gambar 4.12 <i>Prototype</i> Menghapus Berita.....	146
77.	Gambar 4.13 <i>Prototype</i> Menambahkan Registrasi Universitas.	148
78.	Gambar 4.14 <i>Prototype</i> Melihat Daftar Registrasi Universitas.	148
79.	Gambar 4.15 <i>Prototype</i> Menyetujui Registrasi Universitas.	149
80.	Gambar 4.16 <i>Prototype</i> Menolak Registrasi Universitas.....	149
81.	Gambar 4.17 <i>Prototype</i> Melihat Prodi.	151
82.	Gambar 4.18 <i>Prototype</i> Menambahkan Prodi.....	151
83.	Gambar 4.19 <i>Prototype</i> Menghapus Prodi.....	152
84.	Gambar 4.20 <i>Prototype</i> Melihat Fakultas.	152
85.	Gambar 4.21 <i>Prototype</i> Menambahkan Fakultas.....	153
86.	Gambar 4.22 <i>Prototype</i> Menghapus Fakultas.....	153
87.	Gambar 4.23 <i>Prototype</i> Melihat Universitas.	154
88.	Gambar 4.24 <i>Prototype</i> Menambahkan Universitas.	154
89.	Gambar 4.25 <i>Prototype</i> Menghapus Universitas.	154
90.	Gambar 4.26 <i>Prototype</i> Melihat Mahasiswa.....	155
91.	Gambar 4.27 <i>Prototype</i> Menambahkan Mahasiswa.	155
92.	Gambar 4.28 <i>Prototype</i> Mengubah Mahasiswa.	156

93.	Gambar 4.29 <i>Prototype</i> Menghapus Mahasiswa.	156
94.	Gambar 4.30 <i>Prototype form</i> Visualisasi CPL per Mahasiswa.	159
95.	Gambar 4.31 <i>Prototype</i> Visualisasi CPL per Mahasiswa.	160
96.	Gambar 4.32 <i>Prototype form</i> Visualisasi CPMK per Mahasiswa.	160
97.	Gambar 4.33 <i>Prototype</i> Visualisasi CPMK per Mahasiswa.	161
98.	Gambar 4.34 <i>Prototype form</i> Visualisasi CPL per Angkatan.	163
99.	Gambar 4.35 <i>Prototype</i> Visualisasi CPL per Angkatan.	163
100.	Gambar 4.36 <i>Protoype</i> Visualisasi CPMK per Angkatan.....	164
101.	Gambar 4.37 <i>Protoype</i> Visualisasi CPMK per Angkatan.....	164
102.	Gambar 4.38 <i>Prototype form</i> Visualisasi CPMK per Mata Kuliah.	165
103.	Gambar 4.39 <i>Prototype</i> Visualisasi CPMK per Mata Kuliah.	166
104.	Gambar 4.40 Implementasi Halaman <i>Login</i>	171
105.	Gambar 4.41 Implementasi Halaman Menambahkan <i>User</i>	175
106.	Gambar 4.42 Implementasi Halaman Mengubah <i>User</i>	175
107.	Gambar 4.43 Implementasi halaman Mengelola Profil bagian Mengganti Otoritas.....	176
108.	Gambar 4.44 Implementasi halaman Mengelola Tema Universitas.	178
109.	Gambar 4.45 Implementasi halaman Melihat Berita.	179
110.	Gambar 4.46 Implementasi halaman Mengubah Berita.....	181
111.	Gambar 4.47 Implementasi halaman Menambahkan Registrasi Universitas.	183
112.	Gambar 4.48 Implementasi halaman Mengelola Registrasi Universitas bagian <i>approve</i>	186
113.	Gambar 4.49 Implementasi halaman Mengelola Registrasi Universitas bagian <i>reject</i>	186
114.	Gambar 4.50 Implementasi halaman Mengelola Prodi bagian Menambahkan Prodi.	188
115.	Gambar 4.51 Implementasi halaman Mengelola Fakultas bagian Menghapus Fakultas.	189
116.	Gambar 4.52 Implementasi halaman Mengelola Universitas bagian Melihat Universitas.	190

117. Gambar 4.53 Implementasi halaman Mengelola Mahasiswa bagian Mengubah Mahasiswa.....	192
118. Gambar 4.54 Implementasi halaman Melihat Visualisasi CPL per Mahasiswa.	194
119. Gambar 4.55 Implementasi laporan visualisasi CPL per Mahasiswa.	196
120. Gambar 4.56 Implementasi halaman Melihat Visualisasi CPMK per Mahasiswa.....	198
121. Gambar 4.57 Implementasi laporan visualisasi CPMK per Mahasiswa....	200
122. Gambar 4.58 Implementasi halaman Visualisasi CPL per Angkatan.	202
123. Gambar 4.59 Implementasi laporan visualisasi CPL per Angkatan.	204
124. Gambar 4.60 Implementasi halaman Visualisasi CPL per Angkatan.	207
125. Gambar 4.61 Implementasi laporan visualisasi CPMK per Angkatan.	209
126. Gambar 4.62 Implementasi halaman Visualisasi CPMK per Mata Kuliah.	211
127. Gambar 4.63 Implementasi laporan visualisasi CPMK per Mata Kuliah..	213
128. Gambar 4.64 Struktur <i>Test Plan</i> Uji Beban pada JMeter.....	227
129. Gambar 4.65 Konfigurasi <i>Thread Group</i> untuk Skenario Uji Beban.	227

DAFTAR PSEUDOCODE

Pseudocode	Halaman
1. Pseudocode 1. Potongan kode program Melakukan <i>Login</i> (1).	169
2. Pseudocode 2. Potongan kode program Melakukan <i>Login</i> (2).	170
3. Pseudocode 3. Potongan kode program Melakukan <i>Login</i> (3).	170
4. Pseudocode 4. Potongan kode program Menambahkan <i>User</i>	172
5. Pseudocode 5 Potongan kode program Mengubah <i>User</i>	173
6. Pseudocode 6 Potongan kode program pengecekan peran	174
7. Pseudocode 7 Potongan kode program untuk Melindungi Rute Pengelolaan Mahasiswa.....	174
8. Pseudocode 8 Potongan kode program Mengelola Profil bagian Mengganti Otoritas.....	176
9. Pseudocode 9 Potongan kode program Mengelola Tema Universitas.....	177
10. Pseudocode 10 Potongan kode program Melihat Berita.	179
11. Pseudocode 11 Potongan kode program Mengelola Berita bagian Mengubah Berita.	180
12. Pseudocode 12 Potongan kode program Menambahkan Registrasi Universitas.	182
13. Pseudocode 13 Potongan kode program Mengelola Registrasi Universitas bagian approve.	184
14. Pseudocode 14 Potongan kode program Mengelola Registrasi Universitas bagian <i>reject</i>	185
15. Pseudocode 15 Potongan kode program Mengelola Prodi bagian Menambahkan Prodi	187
16. Pseudocode 16 Potongan kode program Mengelola Fakultas bagian Menghapus Fakultas.	188
17. Pseudocode 17 Potongan kode program Mengelola Universitas bagian Melihat Universitas.	189

18. Pseudocode 18 Potongan kode program Mengelola Mahasiswa bagian Mengubah Mahasiswa.....	191
19. Pseudocode 19 Potongan kode program Melihat Visualisasi CPL per Mahasiswa.....	193
20. Pseudocode 20 Potongan kode program Mencetak Laporan Visualisasi CPL per Mahasiswa.....	195
21. Pseudocode 21 Potongan kode program Melihat Visualisasi CPMK per Mahasiswa.....	197
22. Pseudocode 22 Potongan kode program Mencetak Laporan Visualisasi CPMK per Mahasiswa.....	199
23. Pseudocode 23 Potongan kode program Melihat Visualisasi CPL per Angkatan.	201
24. Pseudocode 24 Potongan kode program Mencetak Laporan Visualisasi CPL per Angkatan.	203
25. Pseudocode 25 Potongan kode program Melihat Visualisasi CPMK per Angkatan.	206
26. Pseudocode 26 Potongan kode Program Mencetak Laporan Visualisasi CPMK per Angkatan.	208
27. Pseudocode 27 Potongan kode program Melihat Visualisasi CPMK per Mata Kuliah.....	210
28. Pseudocode 28 Potongan kode program Mencetak Laporan Visualisasi CPMK per Mata Kuliah.	212

I. PENDAHULUAN

1.1 Latar Belakang

Lembaga pendidikan memiliki peran vital sebagai fondasi dalam menciptakan dan membentuk generasi yang unggul dalam aspek pengetahuan, keterampilan, dan sikap. Generasi berkualitas ini diharapkan menjadi sumber daya manusia yang kompeten dan berdaya saing di masa depan. Sebagai institusi yang dinamis, lembaga pendidikan perlu terus melakukan pembaruan dan penyesuaian untuk mencapai tujuan pendidikan yang optimal. Saat ini, salah satu tantangan utama dalam dunia pendidikan adalah rendahnya mutu pendidikan di berbagai jenis dan jenjang pendidikan (Noly dkk., 2022).

Penjaminan mutu pendidikan tinggi merupakan aspek krusial dalam meningkatkan kualitas dan daya saing institusi pendidikan. Jurusan Ilmu Komputer, sebagai salah satu program studi yang berkembang pesat, memerlukan sistem penjaminan mutu yang efektif dan efisien. Sistem yang ada saat ini masih berupa *dashboard* yang hanya menyajikan visualisasi data berdasarkan mahasiswa, angkatan, dan mata kuliah. Meskipun *dashboard* ini telah memberikan gambaran umum tentang kinerja akademik, sistem ini masih memiliki keterbatasan signifikan. Pertama, sistem belum memiliki fitur untuk menghasilkan *file* laporan. Kedua, manajemen pengguna masih terbatas, yang mengakibatkan kesulitan dalam mengatur akses dan peran berbagai pemangku kepentingan. Ketiga, sistem belum mampu menghasilkan laporan yang disesuaikan dengan kebutuhan spesifik berbagai pemangku kepentingan. Keterbatasan ini dapat menghambat proses evaluasi yang mendalam dan upaya peningkatan kualitas pendidikan yang berkelanjutan.

Sistem Penjaminan Mutu yang telah ada di Jurusan Ilmu Komputer telah memberikan kontribusi positif dalam upaya peningkatan kualitas pendidikan. Sistem ini membantu dalam memonitor dan mengevaluasi berbagai aspek

akademik. Namun, seiring dengan bertambahnya kompleksitas proses akademik dan administratif, diperlukan pengembangan lebih lanjut untuk mengoptimalkan sistem yang ada. Khususnya, terdapat kebutuhan mendesak dalam hal manajemen pengguna dan pengelolaan *file* laporan.

Untuk mendukung pengembangan sistem penjaminan mutu ini, penulis memilih untuk menggunakan Laravel, sebuah kerangka kerja *framework* PHP yang modern dan matang. Berdasarkan dokumentasi resminya, Laravel versi 8 dibangun dengan mengadopsi pola desain *Model-View-Controller* (MVC) dan memanfaatkan komponen-komponen dari *framework* Symfony, yang membantunya menjadi fondasi yang kokoh dan aman untuk aplikasi web (Laravel, 2025). Alasan utama pemilihan Laravel adalah karena sintaksnya yang ekspresif dan sistem pakatnya yang modular, yang memungkinkan proses pengembangan menjadi lebih cepat dan efisien. Kemampuan ini secara signifikan mengurangi waktu yang dibutuhkan untuk membangun fungsionalitas dasar, sehingga penulis dapat lebih fokus pada logika bisnis yang kompleks terkait manajemen pengguna dengan berbagai peran (multi-role) dan pembuatan *file* laporan, yang menjadi inti dari penelitian ini.

Penggunaan Laravel dalam pengembangan sistem penjaminan mutu dapat meningkatkan efisiensi dan efektivitas pengelolaan pengguna dan *file* laporan. Dengan memanfaatkan Laravel, sistem ini dapat memberikan solusi yang lebih baik dalam pengelolaan data dan informasi terkait penjaminan mutu di lingkungan Jurusan Ilmu Komputer. Pengembangan ini sejalan dengan upaya untuk menciptakan sistem pendidikan yang lebih responsif terhadap kebutuhan zaman dan mampu menghasilkan lulusan berkualitas tinggi.

Penelitian ini berfokus pada pengembangan modul manajemen pengguna yang komprehensif dan pembuatan sistem pelaporan yang efisien pada *website* Sistem Manajemen Mutu Jurusan Ilmu Komputer. Dalam aspek manajemen pengguna, penelitian ini bertujuan untuk memperluas struktur peran pengguna yang ada. Dari sistem yang sebelumnya hanya mencakup admin, dosen, dan penjamin mutu, akan dikembangkan menjadi struktur yang lebih kompleks dan representatif terhadap hierarki institusi pendidikan. Peran-peran baru yang akan ditambahkan meliputi

Admin Universitas, Penjamin Mutu Universitas, Wakil Rektor, Wakil Dekan, Kepala Program Studi, Penjamin Mutu Fakultas dan Penjamin Mutu Program Studi.

Selain itu, penelitian ini juga akan mengembangkan fitur pembuatan *file* laporan yang terintegrasi. Fitur ini akan memungkinkan pembuatan laporan yang disesuaikan dengan kebutuhan berbagai pemangku kepentingan, mulai dari laporan kinerja akademik hingga evaluasi program studi. Sistem pelaporan ini dapat mempercepat proses pengambilan keputusan dan meningkatkan transparansi dalam pengelolaan mutu pendidikan.

Penelitian ini bertujuan memberikan kontribusi bagi Jurusan Ilmu Komputer Universitas Lampung dalam upaya menjamin pemenuhan standar program studi. Sistem yang dihasilkan akan memfasilitasi proses evaluasi yang lebih mendalam, memungkinkan identifikasi area yang memerlukan peningkatan, serta meningkatkan kualitas pendidikan secara berkelanjutan. Dengan demikian, penelitian ini tidak hanya bertujuan untuk meningkatkan efisiensi operasional, tetapi juga untuk mendukung visi jangka panjang Jurusan Ilmu Komputer.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, penelitian ini berfokus pada pengembangan modul manajemen *user* dan *file* laporan untuk mendukung sistem penjaminan mutu di Jurusan Ilmu Komputer. Modul manajemen *user* dirancang untuk mengakomodasi berbagai peran pengguna, sementara modul *file* laporan diimplementasikan untuk mendukung proses pelaporan.

1.3 Tujuan

Adapun tujuan dari penelitian ini adalah:

1. Mengembangkan modul manajemen *user* yang dapat mengelola berbagai peran pengguna dalam sistem penjaminan mutu Jurusan Ilmu Komputer.
2. Merancang dan mengimplementasikan modul yang menyediakan fungsionalitas untuk menghasilkan *file* laporan dalam format PDF berdasarkan data visualisasi Capaian Pembelajaran Lulusan (CPL) dan Capaian Pembelajaran Mata Kuliah (CPMK) yang ditampilkan pada sistem.

1.4 Manfaat

Adapun manfaat dari penelitian ini antara lain sebagai berikut

1. Bagi Jurusan Ilmu Komputer, sistem yang dikembangkan akan mendukung pengelolaan berbagai peran pengguna dalam penjaminan mutu.
2. Bagi dosen, admin, dan penjamin mutu, sistem ini akan menyediakan *platform* yang dapat mengelola dan melihat laporan pada penjamin mutu.

1.5 Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah:

1. Sistem dikembangkan menggunakan *framework* Laravel dan diimplementasikan sebagai aplikasi web.
2. Pengembangan difokuskan pada penambahan modul manajemen *user* dengan *multi-role* dan modul *file* laporan PDF.

II. TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian terdahulu bertujuan untuk membandingkan penelitian-penelitian yang telah dilakukan sebelumnya dengan penelitian yang akan dilakukan. Selain itu, penelitian terdahulu digunakan sebagai acuan dalam mempelajari literatur yang terkait dengan penelitian ini. Tabel 2.1 berikut menyajikan ringkasan penelitian-penelitian terkait yang telah dilakukan sebelumnya, yang akan digunakan sebagai referensi dan titik awal untuk penelitian ini.

Tabel 2.1 Penelitian terdahulu

No.	Judul Penelitian		Metode	Hasil
1.	Pengembangan Admin Manajemen Perkuliahan di Jurusan Ilmu Komputer (Tamara, 2023).	Modul Sistem Mutu	Penelitian Studi Kasus	Hasil dari penelitian ini adalah Admin pada sistem dapat mengelola kurikulum, CPL (Capaian Pembelajaran Lulusan, RPS (Rencana Pembelajaran Semester), dan CPLMK (CPL yang digunakan pada Mata Kuliah.
2.	Pengembangan Manajemen Penilaian Mata Kuliah di Jurusan Ilmu Komputer (Matanari, 2024).	Sistem Mutu	Penelitian Studi Kasus	Hasil dari penelitian ini adalah menghasilkan Sistem Manajemen Mutu Penilaian Mata Kuliah dengan pengujian <i>black-box-testing</i> dengan nilai rata-rata persentase sebesar 90.19%. Sehingga pengembangan sistem ini sangat diterima dan layak untuk digunakan di Jurusan Ilmu Komputer.
3.	<i>Multi-Roles Based Rancang Bangun Surat Dengan Metode Rapid Application Development</i> (RAD) (Astuti dkk., 2025) .	Pada Surat	Penelitian Studi Kasus	Hasil dari penelitian ini adalah aplikasi pengelolaan surat perjalanan dinas berbasis web yang dilengkapi dengan manajemen pengguna berbasis multi-peran. Pengujian fungsional 99,01%, dan UAT menunjukkan penerimaan pengguna yang baik hingga sangat baik.

Penelitian pada tahun 2023 berjudul “Pengembangan Modul Admin Sistem Manajemen Mutu Perkuliahan di Jurusan Ilmu Komputer” yang diteliti oleh Anggie Tamara berfokus pada pengembangan Modul Admin Sistem Manajemen Mutu Perkuliahan berbasis web di Jurusan Ilmu Komputer Universitas Lampung menggunakan *framework* Laravel. Tujuan utama penelitian ini adalah memaksimalkan kinerja perkuliahan di setiap semester untuk setiap mata kuliah di Jurusan Ilmu Komputer Universitas Lampung secara *online*. Hasil penelitian menunjukkan bahwa admin dapat melakukan berbagai fungsi pada sistem, seperti mengelola kurikulum, CPL (Capaian Pembelajaran Lulusan), RPS (Rencana Pembelajaran Semester), dan CPLMK (CPL yang digunakan pada Mata Kuliah).

Penelitian pada tahun 2024 berjudul “Pengembangan Sistem Manajemen Mutu Penilaian Mata Kuliah di Jurusan Ilmu Komputer” yang diteliti oleh Weike Sandy Putra Matanari berfokus pada pengembangan Sistem Manajemen Mutu Penilaian Mata Kuliah di Jurusan Ilmu Komputer Universitas Lampung berbasis web menggunakan *framework* Laravel. Pengembangan sistem ini menggunakan metode *Rapid Prototyping* yang meliputi tahap perencanaan, desain, pengumpulan umpan balik, implementasi, dan pengujian. Sistem ini memungkinkan dosen untuk membuat soal, menyusun *template* penilaian, dan mengimpor data ke *database*. Hasil pengujian menunjukkan kelayakan sistem dengan skor *User Acceptance Testing* mencapai 90,19%, menandakan potensinya dalam meningkatkan kualitas pendidikan dan evaluasi kompetensi mahasiswa.

Penelitian pada tahun 2025 berjudul “*Multi-Roles Based* Pada Rancang Bangun Surat Dengan Metode *Rapid Application Development* (RAD)” yang diteliti oleh Teresa Puji Astuti dkk. berfokus pada perancangan dan implementasi aplikasi Pengelolaan Surat Perjalanan Dinas berbasis website di Dinas TPHP Kabupaten Magetan menggunakan *framework* Laravel. Tujuan utama penelitian ini adalah untuk meningkatkan kontrol dan manajemen perjalanan dinas, termasuk pengelolaan Surat Perintah Dinas (SPD), administrasi, dan biaya akomodasi, dengan sistem yang mendukung berbagai peran pengguna (*multi-roles based*). Hasil penelitian menunjukkan bahwa aplikasi yang dikembangkan menggunakan metode *Rapid Application Development* (RAD) ini dapat meningkatkan efisiensi

dan efektivitas pengelolaan perjalanan dinas, dengan hasil pengujian fungsional mencapai 99,01% kesesuaian dan hasil *User Acceptance Testing* (UAT) yang menunjukkan kategori baik hingga sangat baik untuk berbagai aspek sistem.

2.2 Manajemen User

Manajemen *user* dalam sistem informasi, seperti pada sistem manajemen dokumen laboratorium, merupakan suatu proses atau modul yang berfungsi untuk mengelola pengguna dan hak akses mereka terhadap sistem. Proses ini mencakup berbagai aktivitas, seperti menambah, menghapus, memelihara, dan mencari data pengguna sistem, mengatur dan mengkonfigurasi peran pengguna (misalnya, administrator laboratorium, dosen, dan mahasiswa) beserta izin yang dimiliki oleh masing-masing peran melalui pendekatan *Role-Based Access Control* (RBAC), serta pembaruan informasi personal pengguna. Sistem manajemen pengguna yang efektif, terutama yang mengimplementasikan RBAC, sangat penting untuk menyederhanakan kompleksitas manajemen otoritas, menjaga keamanan dan stabilitas operasional sistem melalui mekanisme seperti otentikasi dan perlindungan data, serta meningkatkan level informasi dan standardisasi manajemen secara keseluruhan (Jing dkk., 2020). Implementasi manajemen pengguna yang baik tidak hanya meningkatkan efisiensi dalam pemrosesan otoritas baru, tetapi juga memastikan bahwa setiap pengguna memiliki akses yang tepat sesuai dengan peran dan tanggung jawabnya dalam organisasi, sehingga mendukung pengelolaan sumber daya yang lebih ilmiah dan terstandar.

2.3 Laporan

Laporan merupakan sarana komunikasi di mana penyusun menyajikan informasi kepada pihak lain sebagai bagian dari tanggung jawabnya. Isinya mencakup data komprehensif yang sesuai dengan fakta dan disusun secara sistematis untuk menjamin keakuratan serta kemudahan pemahaman informasi (Hidayat & Irvanda, 2022). Laporan bertujuan menyampaikan informasi tentang kinerja suatu aktivitas yang terukur kepada atasan, serta dapat berfungsi sebagai alat evaluasi untuk perbaikan berkelanjutan, khususnya bagi instansi pemerintah dalam meningkatkan performanya. Pada hakikatnya, laporan adalah penyajian fakta mengenai hasil kerja yang berkaitan dengan tanggung jawab. Tujuan laporan adalah memberikan

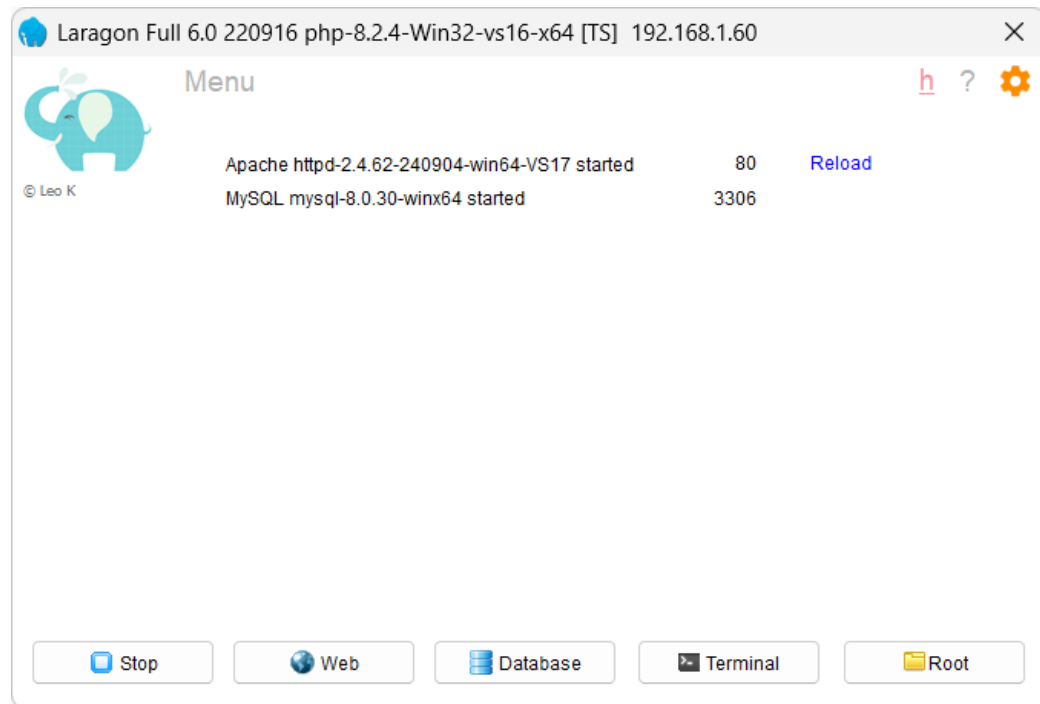
gambaran peristiwa yang terjadi, dengan kebenaran isi yang tercermin melalui struktur yang sistematis, penalaran yang jelas, dan bahasa yang kritis (Hidayat & Irvanda, 2022). Laporan yang baik tidak hanya informatif, tetapi juga dapat menjadi dasar pengambilan keputusan yang efektif bagi pihak-pihak terkait.

2.4 PHP (*Hypertext Preprocessor*)

PHP (*Hypertext Preprocessor*) dikenal luas sebagai bahasa skrip sisi server yang bersifat *open-source* dan memegang peranan penting dalam pengembangan aplikasi web dinamis. Cara kerjanya adalah dengan dieksekusi di server untuk memproses logika dan menghasilkan konten halaman web sebelum dikirimkan ke browser pengguna. Fleksibilitas PHP dalam berinteraksi dengan berbagai sistem manajemen basis data dan ekosistemnya yang didukung oleh komunitas besar telah menjadikannya pilihan populer. Untuk meningkatkan efisiensi, struktur, dan kecepatan dalam pengembangan aplikasi PHP yang kompleks, banyak pengembang memanfaatkan *framework*. *Framework* PHP, seperti yang dibahas oleh Tenzin dkk. (2022), menyediakan struktur dasar, *libraries*, dan alat bantu yang telah teruji untuk menangani tugas-tugas umum dalam pengembangan web, termasuk *routing*, interaksi basis data, manajemen sesi, dan keamanan. Penggunaan *framework* seperti Laravel, CodeIgniter, dan Symfony, memungkinkan pengembang untuk lebih fokus pada logika bisnis spesifik aplikasi mereka daripada membangun semuanya dari awal, serta mendorong penggunaan pola desain yang baik seperti *Model-View-Controller* (MVC) dan praktik pengembangan yang lebih terstruktur (Tenzin, 2022).

2.5 Laragon

Laragon adalah perangkat lunak pengembangan web yang memungkinkan pengembang menggunakan server lokal atau *localhost* tanpa memerlukan koneksi internet. Perangkat lunak gratis ini berfungsi sebagai server mandiri yang kompatibel dengan berbagai sistem operasi, menyediakan beragam layanan dan fitur seperti Apache, PHP Server, PhpMyAdmin, MySQL, Memcached, Redis, Composer, Xdebug, Cmdr, serta dukungan untuk Laravel (Budiman dkk., 2023).



Gambar 2.1 Antarmuka Laragon.

2.6 Laravel

Laravel adalah kerangka kerja PHP *open-source* yang dirancang untuk mengoptimalkan dan mempercepat proses pengembangan aplikasi web sisi server. Dengan beragam fitur bawaan yang mudah digunakan, Laravel menawarkan kemudahan bagi para pengembang. *Framework* ini mengadopsi sistem modular dengan manajemen dependensi yang efektif, memungkinkan integrasi fungsionalitas tambahan ke dalam aplikasi Laravel secara lancar. Arsitektur *Model-View-Controller* (MVC) menjadi landasan utama Laravel, yang juga dapat berfungsi sebagai *back-end* API untuk aplikasi JavaScript *single-page* (Aung dkk., 2024). Keunggulan Laravel dalam menyederhanakan tugas-tugas kompleks seperti otentikasi, *routing*, dan *caching* membuat *framework* ini semakin populer di kalangan pengembang web modern.

Penggunaan Laravel dalam penelitian ini memberikan kemampuan yang baik dalam pengembangan web karena Laravel memiliki serangkaian fitur canggih. Fitur canggih Laravel ini antara lain adalah Eloquent ORM (*Object-Relational Mapping*), yang memungkinkan pengembang menyederhanakan interaksi antar

basis data dengan menyediakan cara ekspresif dalam mengolah basis data sebagai objek sehingga meningkatkan produktivitas dan mengurangi waktu pengembangan. Laravel juga memiliki sistem *routing* yang kuat sehingga memungkinkan pembuatan URL yang bersih dan *SEO-friendly*. Laravel juga menyediakan *controller* yang memfasilitasi pengembang dalam mengatur logika pemrograman, menangani permintaan, serta mempromosikan modularitas kode dan pemeliharaan. Dengan fitur-fitur ini, yang dikombinasikan dengan dokumentasi Laravel yang sangat lengkap serta dukungan komunitasnya yang sangat aktif, Laravel menjadi solusi yang canggih dan efisien dalam pengembangan web.

```

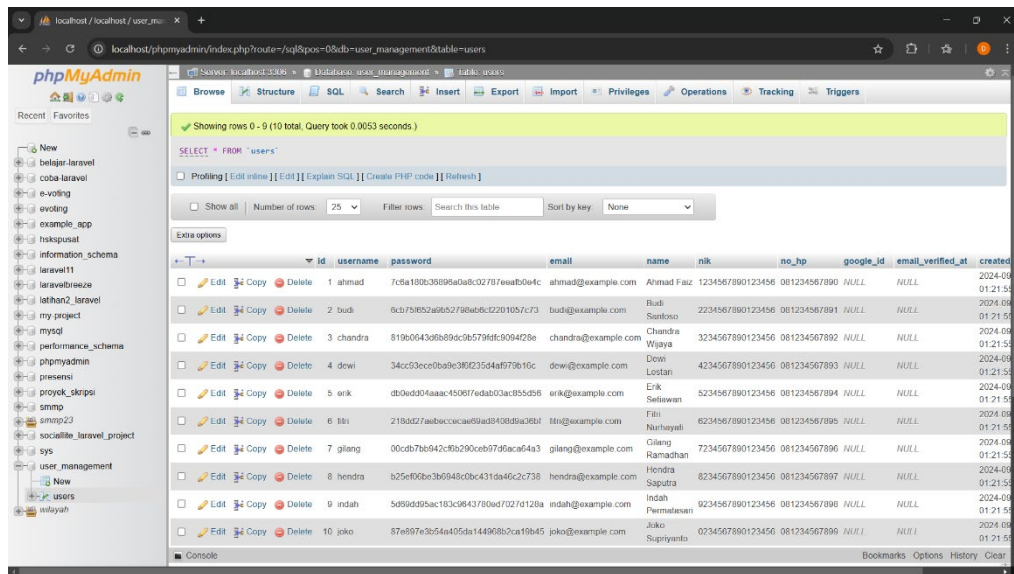
1 namespace App\Http\Controllers;
2
3 use App\Models\User;
4
5 class UserController extends Controller
6 {
7     public function index() {
8         $users = User::all();
9         return view('user.index', compact('users'));
10    }
11 }

```

Gambar 2.2 Contoh Kode Laravel.

2.7 MySQL

MySQL adalah sebuah Sistem Manajemen Basis Data (DBMS) yang bersifat *open-source* dan sering disingkat sebagai DBMS. MySQL juga termasuk dalam kategori Sistem Manajemen Basis Data Relasional (RDBMS), di mana struktur *databasenya* menggunakan metode relasional untuk proses pengambilan data dan berfungsi sebagai penghubung antara perangkat lunak dan server *database*. Salah satu hal penting dari MySQL adalah kemampuannya untuk digunakan secara gratis. Sebagai RDBMS, data yang dikelola dalam MySQL ditempatkan dalam beberapa tabel terpisah, memungkinkan manipulasi data yang jauh lebih cepat. MySQL dapat digunakan untuk membuat dan mengelola *database* beserta isinya, mulai dari skala kecil hingga sangat besar, dan telah berkembang dengan berbagai fitur seperti dukungan untuk *trigger*, beragam tipe data, serta berbagai lapisan keamanan (Wahyudi dkk., 2022).



Showing rows 0 - 9 (10 total. Query took 0.0053 seconds)

SELECT * FROM `users`

Prototyping [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

		id	username	password	email	name	nik	no_hp	google_id	email_verified_at	created_at
☐	Edit Copy Delete	1	ahmad	7c5a180b36895a0a8c02707eeab0e7c	ahmad@example.com	Ahmad Faiz	1234567890123456	081234567890	NULL	NULL	2024-09-01 21:55
☐	Edit Copy Delete	2	budi	8cb73852a0527f8ebdc0201057c73	budi@example.com	Budi Santoso	2234567890123456	081234567891	NULL	NULL	2024-09-01 21:55
☐	Edit Copy Delete	3	chandra	81960643d0b89dc9b5796ff909428e	chandra@example.com	Chandra Wijaya	3234567890123456	081234567892	NULL	NULL	2024-09-01 21:55
☐	Edit Copy Delete	4	dewi	34cc03ccc0ba9c30f235d4af79b10c	dewi@example.com	Dewi Lestari	4234567890123456	081234567893	NULL	NULL	2024-09-01 21:55
☐	Edit Copy Delete	5	erik	d50ed304aac4508f7edab03ac055d5b	erik@example.com	Erik Setiawan	5234567890123456	081234567894	NULL	NULL	2024-09-01 21:55
☐	Edit Copy Delete	6	lita	218a027aebccca9a89a08400d8a30b8	lita@example.com	Fili Nurhayati	6234567890123456	081234567895	NULL	NULL	2024-09-01 21:55
☐	Edit Copy Delete	7	gilang	00c0b7b042c0b290ceb07d9acaf4a3	gilang@example.com	Gilang Ramadhan	7234567890123456	081234567896	NULL	NULL	2024-09-01 21:55
☐	Edit Copy Delete	8	hendra	b25e00ba3b684c0bc431da46c2c738	hendra@example.com	Hendra Saputra	8234567890123456	081234567897	NULL	NULL	2024-09-01 21:55
☐	Edit Copy Delete	9	indah	5d59d95ac183c9843780ed7027d128a	indah@example.com	Indah Permatasari	9234567890123456	081234567898	NULL	NULL	2024-09-01 21:55
☐	Edit Copy Delete	10	joko	87e897e3b54a405da144968b2ca10b45	joko@example.com	Joko Supriyanto	0234567890123456	081234567899	NULL	NULL	2024-09-01 21:55

Gambar 2.3 MySQL pada PHPMyAdmin.

2.8 Unified Modeling Language (UML)

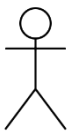
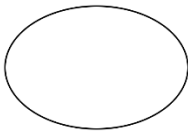
Unified Modeling Language (UML) merupakan sebuah sistem notasi visual terstandarisasi yang terdiri dari berbagai diagram yang saling berhubungan. Setiap diagram mewakili aspek tertentu dari sistem yang sedang dimodelkan, dan ketika digabungkan, diagram-diagram tersebut menawarkan pandangan terhadap keseluruhan arsitektur sistem (Elsayed dkk., 2020). Dalam penelitian ini, dua jenis diagram UML yang relevan dan akan dibahas lebih lanjut adalah *Use Case Diagram* dan *Activity Diagram*.

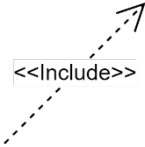
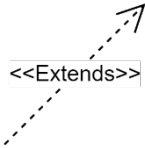
2.8.1 Use Case Diagram

Use case diagram adalah salah satu diagram dalam *Unified Modelling Language (UML)* yang sangat berguna untuk merefleksikan kebutuhan bisnis dari sebuah sistem perangkat lunak yang diusulkan. Diagram ini menggambarkan fungsionalitas sistem dari sudut pandang pengguna dan menangkap tujuan pengguna secara umum. Oleh karena itu, diagram ini sangat penting dalam fase desain perangkat lunak untuk meminimalkan ambiguitas dan meningkatkan pemahaman serta komunikasi antara pemangku kepentingan seperti pengembang, pemilik produk, dan pengguna (Molla dkk., 2024).

Use case diagram terdiri dari beberapa elemen penting, yaitu aktor, *use case*, dan relasi di antara keduanya. Aktor merupakan entitas eksternal yang berinteraksi dengan sistem, seperti pengguna, yang dalam digambarkan dengan simbol *stick figure*. *Use case* menggambarkan fungsionalitas yang disediakan sistem dalam bentuk oval. Asosiasi menghubungkan aktor dan *use case* dengan garis lurus. Generalisasi menunjukkan hierarki antar aktor atau *use case* dengan panah terbuka. *Include* menyertakan fungsionalitas *use case* lain dengan garis putus-putus dan label "<<include>>", sedangkan *Extends* menunjukkan penambahan opsional dengan simbol "<<extends>>", seperti yang dapat dilihat pada tabel 2.2.

Tabel 2.2 Komponen *Use Case*

No	Simbol	Nama	Keterangan
1	 Actor	Aktor	Aktor merepresentasikan pengguna atau entitas yang berinteraksi dengan sebuah sistem. Aktor bisa berupa pengguna manusia, perangkat, atau sistem lain yang terlibat dalam <i>use case</i> .
2		<i>Use case</i>	<i>Use case</i> menggambarkan fungsionalitas atau layanan yang disediakan oleh sistem kepada aktor. Biasanya berisi tindakan-tindakan atau skenario yang dilakukan oleh aktor dengan sistem.
3		Asosiasi	Asosiasi menggambarkan hubungan antara aktor atau <i>use case</i> .
4		Generalisasi	Generalisasi menggambarkan hubungan hierarki antara aktor atau <i>use case</i> . Aktor atau <i>use case</i> yang lebih spesifik mewarisi atau memperluas fungsionalitas dari aktor atau <i>use case</i> yang lebih umum.

No	Simbol	Nama	Keterangan
5		<i>Include</i>	<i>Include</i> menunjukkan bahwa suatu <i>use case</i> selalu menyertakan fungsionalitas <i>use case</i> lain sebagai bagian dari pelaksanaannya.
6		<i>Extends</i>	<i>Extends</i> menunjukkan bahwa suatu <i>use case</i> dapat memperpanjang fungsionalitas <i>use case</i> lain di bawah kondisi tertentu.

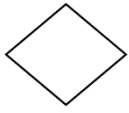
2.8.2 Activity Diagram

Activity diagram adalah salah satu jenis diagram perilaku (behavioral diagram) dalam UML yang berfungsi untuk merepresentasikan aliran aktivitas dalam sebuah sistem. Diagram ini umum digunakan untuk memodelkan proses bisnis dan membantu dalam pengembangan sistem perangkat lunak. Menurut , diagram ini menggambarkan urutan aliran aktivitas dari satu kegiatan ke kegiatan (Kulkarni & Srinivasa, 2021) lainnya. *Activity diagram* umumnya digunakan untuk pemodelan bisnis dan pengembangan sistem perangkat lunak. Komponen utamanya mencakup beberapa notasi, seperti status awal (initial node), status akhir (final node), kotak aktivitas (activity box), dan percabangan (decision box). Setiap langkah tunggal dalam alur proses direpresentasikan sebagai sebuah aksi (action) dalam persegi panjang dengan sudut membulat, sementara alur kendali antar aksi digambarkan dengan garis panah.

Activity diagram memiliki beberapa komponen utama. Status Awal digambarkan dengan lingkaran hitam penuh yang menunjukkan titik mulai dari suatu alur aktivitas. Aktivitas direpresentasikan oleh persegi panjang, yang mewakili langkah atau tindakan dalam proses. Percabangan dilambangkan dengan bentuk belah ketupat yang menunjukkan keputusan atau kondisi yang mengarah pada jalur alternatif dalam alur. Status Akhir digambarkan dengan lingkaran hitam dengan

lingkaran luar menunjukkan akhir dari proses atau alur aktivitas, seperti yang dapat dilihat pada Tabel 2.3.

Tabel 2.3 Komponen *Activity Diagram*

No	Simbol	Nama	Keterangan
1		Status Awal	Status awal merupakan titik pertama dalam diagram aktivitas, menunjukkan di mana proses dimulai.
2		Aktivitas	Aktivitas menggambarkan tindakan atau langkah yang dilakukan dalam proses. Setiap aktivitas menunjukkan suatu pekerjaan atau operasi yang dilakukan dalam alur kerja.
3		Percabangan	Percabangan menunjukkan titik keputusan di mana alur dapat bercabang menjadi dua atau lebih jalur tergantung pada kondisi tertentu.
4		Status Akhir	Status akhir menunjukkan titik akhir dari suatu alur aktivitas, menandakan bahwa proses atau aliran kerja telah selesai.

2.8.3 Class Diagram

Class Diagram adalah salah satu dari dalam *Unified Modeling Language* (UML), sebuah perangkat desain yang dikembangkan oleh IBM dan umum digunakan di industri perangkat lunak untuk merancang sistem. Diagram ini secara spesifik merepresentasikan perilaku statis dari sebuah sistem perangkat lunak, sehingga memberikan pandangan statis dari sebuah aplikasi. Fungsinya tidak hanya untuk memvisualisasikan, mendeskripsikan, dan mendokumentasikan berbagai aspek sistem, tetapi juga dapat digunakan sebagai dasar untuk menghasilkan kode struktural dan *behavioral*. Secara rinci, *Class Diagram* menggambarkan nama kelas, atribut, operasi (metode), dan visibilitasnya. Selain itu, diagram ini juga menunjukkan relasi yang ada antar kelas, seperti generalisasi, agregasi, asosiasi, komposisi, dan antarmuka (Kulkarni & Prasad, 2021). Oleh karena itu, diagram ini

menjadi fondasi penting dalam pemodelan sistem berorientasi objek dengan memberikan gambaran yang jelas mengenai struktur sistem tersebut.

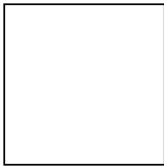
2.8.4 Deployment Diagram

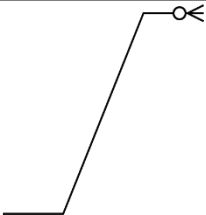
Deployment Diagram adalah salah satu jenis diagram struktural dalam UML yang menggambarkan arsitektur fisik dari suatu sistem perangkat lunak. Diagram ini memodelkan bagaimana komponen perangkat lunak (artifacts) ditempatkan atau di-deploy pada elemen-elemen perangkat keras (nodes), serta bagaimana *node-node* tersebut saling terhubung untuk menjalankan sistem. *Deployment Diagram* sangat berguna untuk memvisualisasikan aspek *runtime* dari sistem, termasuk distribusi komponen, pemrosesan, dan interaksi antar perangkat keras (Bhatt & Nandu, 2021).

2.9 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan salah satu teknik pemodelan data yang paling penting dan krusial dalam proses perancangan basis data. ERD digunakan untuk merepresentasikan data secara konseptual sebelum data tersebut diimplementasikan ke dalam skema basis data yang sebenarnya. Dengan menggunakan ERD, pengembang dapat mengilustrasikan hubungan antar berbagai elemen data atau entitas, atribut-atribut yang dimiliki oleh setiap entitas, serta jenis-jenis hubungan yang terjadi di antara entitas-entitas tersebut. Pembuatan ERD yang baik dan benar menjadi sangat kritis, karena kesalahan dalam tahap perancangan ERD dapat menyebabkan masalah serius dalam implementasi basis data, seperti terjadinya redundansi data, anomali data, atau bahkan kehilangan data (Palinggi dkk., 2024).

Tabel 2.4 Komponen *Entity Relationship Diagram (ERD)*

No	Simbol	Nama	Keterangan
1		Entitas	Entitas merepresentasikan objek atau hal penting yang ingin dimodelkan dalam basis data. Setiap entitas biasanya mewakili tabel dalam basis data, yang berisi kumpulan data yang berhubungan.

No	Simbol	Nama	Keterangan
2		Relasi	Relasi menunjukkan hubungan antara dua entitas dalam model ERD. Relasi dapat menggambarkan jenis hubungan seperti <i>one-to-one</i> , <i>one-to-many</i> , atau <i>many-to-many</i> .

2.10 Software Development Life Cycle (SDLC)

Software Development Life Cycle (SDLC) atau Siklus Hidup Pengembangan Perangkat Lunak merupakan suatu kerangka kerja konseptual yang menyusun dan mendefinisikan tahapan-tahapan utama yang terlibat dalam keseluruhan proses pengembangan sebuah sistem atau aplikasi perangkat lunak, dimulai dari analisis kebutuhan awal dan studi kelayakan hingga tahap implementasi, pengujian, penerapan, dan pemeliharaan berkelanjutan (S. Gupta dkk., 2022). SDLC menyediakan berbagai model yang menguraikan pendekatan berbeda dalam proses pengembangan, di antaranya adalah model *Waterfall* yang bersifat sekuensial dan model-model iteratif seperti *Prototyping* (A. Gupta dkk., 2021). Selain itu, terdapat pula model seperti *Rapid Application Development* (RAD) yang menekankan pada siklus pengembangan yang sangat singkat dan cepat melalui pendekatan iteratif dan konstruksi berbasis komponen. Setiap model memiliki karakteristik, kelebihan, dan kekurangan masing-masing yang menjadikannya lebih sesuai untuk jenis proyek tertentu. Pemilihan model SDLC yang tepat menjadi faktor krusial yang dapat memengaruhi efisiensi, kualitas, dan keberhasilan keseluruhan proyek pengembangan perangkat lunak, serta memastikan produk akhir selaras dengan kebutuhan pengguna dan tujuan bisnis yang ditetapkan.

2.11 Rapid Application Development (RAD)

Rapid Application Development (RAD) adalah sebuah model siklus hidup pengembangan sistem yang dirancang untuk proses pengembangan yang lebih cepat dengan menghasilkan kualitas produk yang lebih baik dibandingkan dengan metode siklus tradisional. Lebih lanjut, Pressman (2005) yang dikutip dalam penelitian (Friadi & Gulo, 2020) menyatakan bahwa RAD merupakan proses model

perangkat lunak inkremental yang mengutamakan siklus pengembangan yang singkat. Salah satu kunci keberhasilan RAD dalam mempercepat proses pengembangan adalah keterlibatan langsung pengguna dalam keseluruhan proses, di mana pengguna bertindak sebagai pengambil keputusan pada setiap tahapan sehingga pengembangan dapat lebih cepat dan lebih sesuai dengan kebutuhan pengguna. Pengembangan sistem informasi dengan model RAD umumnya melalui tiga tahap utama yang melibatkan, pengembang, dan pengguna, yaitu Perencanaan Kebutuhan (*Requirement Planning*), Desain Sistem, dan Implementasi. Pada tahap Perencanaan Kebutuhan, pengguna dan analis bertemu untuk mengidentifikasi tujuan aplikasi dan kebutuhan analisis. Selanjutnya, dalam *Workshop* Perancangan RAD, pengguna memberikan masukan dan saran terhadap *prototype* yang dikembangkan, dan analis akan memperbaiki modul yang dikembangkan sesuai dengan saran tersebut. Tahap terakhir adalah Implementasi, di mana sistem yang telah disepakati dan diuji kemudian diperkenalkan ke organisasi.

Menurut (Friadi & Gulo, 2020) yang merujuk pada Kendall & Kendall, model RAD melibatkan analis, pengembang, dan pengguna dalam tiga tahap utama yaitu Perencanaan Kebutuhan (*Requirement Planning*), *Workshop* Perancangan RAD (RAD Design *Workshop*), dan Implementasi. Meskipun demikian, tahapan-tahapan ini dapat dirinci lebih lanjut. Model RAD yang diadopsi dalam penelitian ini mengacu pada Gambar 2.4 menguraikan proses tersebut ke dalam empat tahapan utama untuk memastikan pendekatan yang lebih cepat dan lebih iteratif dalam pengembangan perangkat lunak:

1. Perencanaan Kebutuhan

Tahap ini merupakan fase di mana pengguna dan analis bertemu untuk mengidentifikasi tujuan aplikasi serta kebutuhan sistem. Tim pengembang bekerja sama dengan *stakeholder* untuk mendefinisikan tujuan, fungsi, dan kebutuhan pengguna pada proyek tersebut, dengan fokus pada pencapaian tujuan perusahaan. Keterlibatan aktif dari kedua belah pihak sangat penting dalam tahap ini.

2. Desain Sistem (RAD Design *Workshop*)

Pada tahap ini, tim pengembang mulai mendesain sistem menggunakan pendekatan iteratif dengan fokus pada tiga langkah utama:

1) *Prototype*

Pengembang membuat *prototype* awal yang mencakup elemen-elemen inti dari sistem. *Prototype* ini bersifat sederhana dan fokus pada fitur utama sistem, tanpa menghabiskan terlalu banyak waktu untuk detail. Tujuannya adalah memberikan gambaran nyata kepada pengguna mengenai bagaimana sistem akan bekerja.

2) *Test*

Prototype yang telah dibuat kemudian diuji dengan melibatkan pengguna secara langsung. Pengujian ini bertujuan untuk mengidentifikasi apakah *prototype* sudah memenuhi kebutuhan pengguna atau jika ada fitur yang perlu ditambahkan, dihapus, atau diperbaiki. Proses ini sering melibatkan diskusi dan pemberian komentar antara pengguna dan tim pengembang hingga mencapai kesepakatan.

3) *Refine*

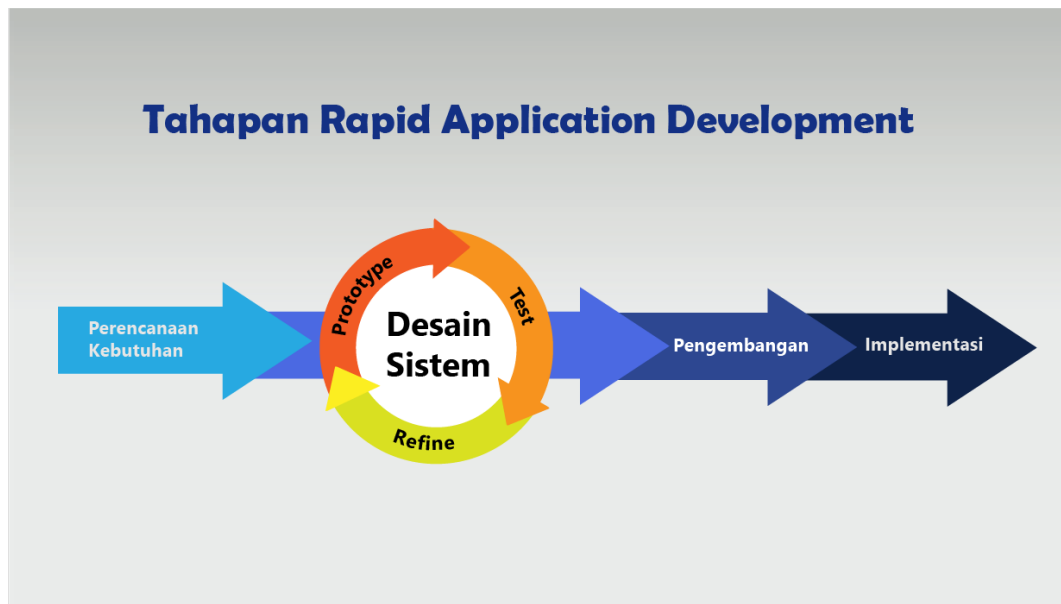
Berdasarkan umpan balik dan saran yang didapat dari pengguna selama pengujian *prototype*, *prototype* tersebut diperbaiki dan disempurnakan oleh analis. Perbaikan ini bisa mencakup perubahan pada antarmuka pengguna (UI), logika sistem, atau fitur. Proses iteratif membuat *prototype*, menguji, dan menyempurnakannya ini dilakukan berulang hingga *prototype* dianggap memadai dan disetujui untuk dilanjutkan ke tahap pengembangan.

3. Pengembangan (*Construction*)

Setelah desain sistem disepakati melalui *workshop* dan *prototype*, tahap pengembangan dimulai. Tahap ini melibatkan pembuatan aplikasi secara iteratif berdasarkan desain yang telah disempurnakan. Sebagaimana disebutkan oleh (Friadi & Gulo, 2020), setelah faktor-faktor bisnis dan teknis disepakati, sistem baru mulai dibuat oleh pengembang sesuai dengan yang disetujui pengguna. Dengan memanfaatkan alat-alat yang ada, pengembang dapat membangun aplikasi secara efisien.

4. Implementasi (*Implementation*)

Tahap implementasi adalah proses peluncuran aplikasi yang telah selesai dikembangkan dan diuji. Sistem yang baru ini akan diuji terlebih dahulu untuk memastikan tidak ada *bug* sebelum diperkenalkan ke organisasi. Pada tahap ini, tim memantau kinerja aplikasi dan pengguna dapat memberikan umpan balik lebih lanjut terhadap sistem yang telah dibuat serta memberikan persetujuan akhir. Tahap ini melengkapi siklus hidup RAD, memastikan aplikasi berkualitas tinggi dapat dihasilkan dalam waktu yang lebih singkat.



Gambar 2.4 Tahapan *Rapid Application Development*.

2.12 *Black-Box Testing*

Pengujian perangkat lunak merupakan tahap krusial dalam siklus pengembangan untuk memastikan kualitas dan kesesuaian fungsionalitas sistem yang dibangun. Salah satu metode pengujian yang umum digunakan adalah *Black Box Testing* (Pengujian Kotak Hitam). Menurut (Sholeh dkk., 2021), *black box testing* adalah pengujian yang dilakukan hanya dengan mengamati hasil eksekusi melalui data uji dan memeriksa fungsionalitas perangkat lunak. Dengan kata lain, dalam pendekatan ini, struktur internal atau cara kerja kode program tidak menjadi fokus pengujian; sistem dianggap sebagai sebuah "kotak hitam" di mana penguji hanya berinteraksi dengan antarmuka sistem, memberikan *input*, dan memvalidasi *output* yang dihasilkan berdasarkan spesifikasi kebutuhan yang telah ditentukan. Tujuan utama dari metode *black box testing* adalah untuk menemukan kesalahan-kesalahan

fungsional, seperti apakah setiap fungsi berjalan sesuai dengan yang diharapkan, apakah ada fitur yang hilang atau tidak berfungsi, serta untuk memastikan bahwa sistem dapat menangani berbagai kondisi *input* dengan benar dan menghasilkan *output* yang akurat. Karena fokusnya pada aspek eksternal dan fungsionalitas dari perspektif pengguna, *black box testing* dapat dilakukan oleh tim penguji yang tidak memiliki pengetahuan mendalam mengenai implementasi teknis sistem.

2.13 Apache JMeter

Apache JMeter adalah perangkat lunak open-source yang berfungsi sebagai alat untuk melakukan *performance testing* pada sebuah aplikasi. JMeter digunakan untuk menganalisis kinerja sebuah sistem informasi dengan cara menyimulasikan beban dari sejumlah pengguna virtual secara bersamaan. Pengujian ini berfokus pada pengukuran metrik-metrik kunci seperti *throughput*, *response time*, dan persentase error untuk mengetahui tingkat keandalan dan stabilitas sistem dalam menangani beban kerja tertentu. (Indrianto, 2023).

III.METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

3.1.1 Waktu Penelitian

Penelitian ini dilaksanakan pada awal semester ganjil tahun akademik 2024/2025, yang dimulai pada bulan September 2024 hingga bulan Maret 2025. Waktu penelitian akan dijelaskan dengan tabel *gantt chart* pada Tabel 3.1.

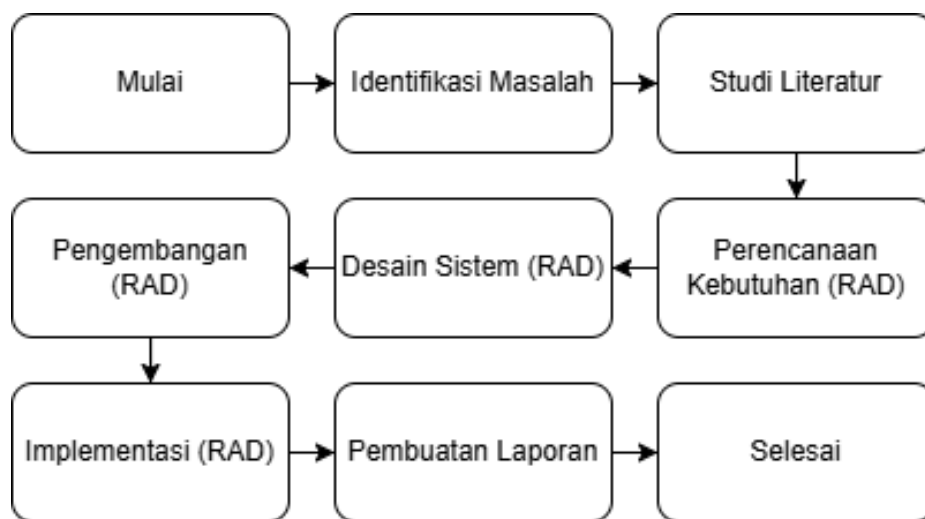
Tabel 3.1 Rencana Penelitian

[illegible]

3.1.2 Tempat Penelitian

Penelitian ini dilaksanakan di Jurusan Ilmu Komputer, Universitas Lampung, yang berlokasi di Jalan Prof. Dr. Ir. Sumantri Brojonegoro No. 1, Gedong Meneng, Kecamatan Rajabasa, Kota Bandar Lampung, Lampung 35141 dan kediaman Kedaton, Bandar Lampung.

3.2 Tahapan Penelitian



Gambar 3.1 Tahapan Penelitian.

Tahapan penelitian ini mengadopsi kerangka kerja Siklus Hidup Pengembangan Perangkat Lunak atau *Software Development Life Cycle* (SDLC) dengan mengintegrasikan model *Rapid Application Development* (RAD) dengan tahapan penyusunan skripsi. Ada tujuh langkah utama yang dilakukan selama penelitian, yaitu identifikasi masalah, studi literatur, perencanaan kebutuhan, desain sistem, pengembangan, implementasi, dan pembuatan laporan.

1. Identifikasi masalah

Pada tahap ini, dilakukan identifikasi terhadap kebutuhan dan permasalahan yang ada dalam sistem penjamin mutu yang sebelumnya. Hasil dari tahap ini digunakan untuk merumuskan kebutuhan sistem yang akan dikembangkan, khususnya untuk modul manajemen *user* dan *file* laporan.

2. Studi Literatur

Studi literatur dilakukan untuk menelusuri referensi-referensi yang terkait dengan sistem penjamin mutu di pendidikan tinggi, manajemen *user* dalam sistem informasi, pengelolaan *file* laporan, serta metode pengembangan sistem informasi menggunakan pendekatan RAD. Hal ini bertujuan untuk membangun dasar teori yang kuat dan memahami perkembangan terkini dalam bidang ini.

3. Perencanaan Kebutuhan (RAD)

Tahap ini merupakan bagian dari metode RAD, di mana kebutuhan fungsional dan non-fungsional sistem direncanakan dengan merujuk pada hasil identifikasi masalah dan masukan dari *stakeholder*. Perencanaan ini akan menjadi panduan dalam desain dan pengembangan sistem.

4. Desain Sistem (RAD)

Pada tahap ini, dilakukan pembuatan desain awal sistem yang akan dikembangkan berdasarkan kebutuhan yang telah ditentukan. Desain sistem dilakukan secara iteratif menggunakan metode *live prototyping*, dengan fokus pada pembuatan antarmuka dan alur sistem yang dapat langsung diuji oleh pengguna..

5. Pengembangan (RAD)

Tahap ini berfokus pada pengembangan sistem secara teknis menggunakan Laravel sebagai *framework* utama. Sistem akan dibangun secara bertahap, dimulai dari pembuatan fitur utama hingga penyempurnaan antarmuka dan fungsionalitas sesuai dengan desain yang telah dibuat.

6. Implementasi (RAD)

Pada tahap implementasi, sistem yang telah dikembangkan akan diuji dan diterapkan dalam lingkungan yang lebih nyata, seperti pada server lokal atau internal jurusan. Pengguna akan mulai menggunakan sistem dan memberikan umpan balik untuk perbaikan lebih lanjut.

7. Penulisan Laporan

Tahap terakhir adalah penyusunan laporan yang menjelaskan seluruh proses penelitian, mulai dari identifikasi masalah, studi literatur, metodologi yang digunakan, hasil pengembangan sistem, analisis evaluasi, hingga kesimpulan dan rekomendasi untuk pengembangan lebih lanjut.

3.3 Identifikasi Masalah

Pada sistem penjamin mutu yang telah dikembangkan sebelumnya, manajemen pengguna masih bersifat umum dan belum mencerminkan struktur organisasi yang lengkap dalam lingkup perguruan tinggi. Sistem hanya mengenal tiga peran utama, yaitu Admin, Penjamin Mutu, dan Dosen, tanpa memperhitungkan jenjang otoritas seperti fakultas atau universitas. Hal ini menjadi kendala ketika sistem ingin digunakan oleh lebih dari satu institusi, karena tidak ada pemisahan akses dan wewenang antar unit organisasi yang berbeda. Untuk menjawab kebutuhan tersebut, dilakukan pengembangan sistem dengan menambahkan struktur peran pengguna secara hierarkis. Pengelolaan hak akses ini diimplementasikan menggunakan pendekatan *Role-Based Access Control* (RBAC) untuk memastikan setiap pengguna memiliki izin yang sesuai dengan perannya. Peran-peran baru yang ditambahkan meliputi Admin, Admin Universitas, Wakil Rektor, Wakil Dekan, Kepala Program Studi, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi, dan Dosen. Pengembangan ini bertujuan agar sistem dapat digunakan secara lebih luas oleh berbagai universitas yang mendaftar, dengan pengaturan hak akses yang sesuai dengan struktur masing-masing institusi.

Selain pengembangan pada sisi manajemen pengguna, sistem sebelumnya juga belum menyediakan fitur untuk mencetak laporan visualisasi CPL dan CPMK dalam bentuk dokumen. Oleh karena itu, pada pengembangan sistem ini ditambahkan fitur cetak *file* laporan dalam format PDF untuk visualisasi CPL dan CPMK. Fitur ini memungkinkan pengguna untuk menghasilkan laporan yang terstruktur, mudah diakses, serta siap untuk digunakan dalam kegiatan *monitoring*, evaluasi, dan pelaporan akademik.

3.4 Studi Literatur

Penjaminan mutu dalam pendidikan tinggi merupakan proses berkelanjutan yang bertujuan untuk memastikan tercapainya standar mutu akademik dan non-akademik. Dalam pelaksanaannya, dibutuhkan sistem yang mampu mengelola data secara terstruktur dan mengakomodasi berbagai peran pengguna sesuai dengan tingkatan organisasi, seperti universitas, fakultas, dan program studi. Sistem dengan manajemen pengguna yang baik memungkinkan pembagian akses dan tanggung

jawab yang jelas, sehingga setiap pengguna dapat menjalankan fungsinya sesuai dengan otoritas yang dimiliki. Hal ini penting untuk mendukung proses evaluasi, pemantauan, serta perbaikan mutu secara menyeluruh dan berkelanjutan.

Selain pengelolaan pengguna, penyediaan fitur pelaporan juga menjadi bagian penting dalam sistem penjamin mutu. Laporan yang disusun berdasarkan data yang divisualisasi memungkinkan pihak terkait untuk memahami dan mengevaluasi capaian pembelajaran dengan lebih efektif. Oleh karena itu, sistem perlu menyediakan mekanisme untuk menghasilkan laporan dalam bentuk dokumen yang dapat dicetak dalam bentuk PDF.

3.5 Perencanaan Kebutuhan (*Rapid Application Development*)

Dalam penelitian ini, metodologi pengembangan sistem yang dipilih adalah model *Rapid Application Development* (RAD), yang merupakan salah satu model dalam kerangka *Software Development Life Cycle* (SDLC) yang menekankan pada siklus pengembangan yang cepat. Perencanaan kebutuhan dalam penelitian ini dilakukan dengan pendekatan RAD yang berfokus pada kecepatan pengembangan dan iterasi berkelanjutan melalui metode *live prototyping*. Dengan menggunakan pendekatan ini, penulis secara aktif membangun *prototype* langsung berdasarkan kebutuhan pengguna, lalu melakukan evaluasi dan perbaikan secara bertahap berdasarkan umpan balik.

3.5.1 Perangkat Penelitian

Penelitian ini menggunakan beberapa alat yang diperlukan untuk menunjang tahapan yang dilakukan sesuai dengan prosedur penelitian. Alat-alat yang digunakan dibagi menjadi dua jenis, yaitu perangkat keras dan perangkat lunak. Daftar alat yang digunakan dalam proses penelitian ini dapat dilihat di bawah berikut:

1. Perangkat Keras

Dalam melaksanakan penelitian ini, penulis menggunakan laptop dengan spesifikasi yang memadai untuk menjalankan proses pengembangan dan pengujian sistem. Perangkat keras yang digunakan memiliki komponen-komponen sebagai berikut:

1. *System Manufacturer* : Lenovo

- 2. *System Model* : Thinkpad P51
- 3. *Processor* : intel® Core™ i7-7820HQ CPU @ 2.90GHz
- 4. *RAM* : 16GB DDR4
- 5. *Penyimpanan* : SSD 256GB SSD 512GB

2. Perangkat Lunak

Dalam pelaksanaan penelitian ini, penulis memanfaatkan berbagai perangkat lunak untuk mendukung proses pengembangan sistem. Setiap perangkat lunak dipilih berdasarkan fungsi spesifik dan kontribusinya terhadap efisiensi pengembangan. Berikut adalah daftar perangkat lunak yang digunakan beserta penjelasan singkatnya:

- 1. Sistem Operasi Windows 11 Pro 64-bit
- 2. Visual Studio Code 1.101.0
- 3. Google Chrome 137.0.7151.120 (Official Build) (64-bit)
- 4. Laragon 6.0
- 5. Laravel Framework 8.83.27
- 6. Bootstrap
- 7. GitHub

3.5.2 Ringkasan Eksekutif

Penelitian (proyek) pengembangan Sistem Penjamin Mutu Jurusan Ilmu Komputer dengan fokus pada Modul Manajemen *User* dan *File* Laporan bertujuan untuk menciptakan sebuah aplikasi yang mampu mengelola pengguna sistem dan *file* laporan terkait penjaminan mutu secara efektif dan efisien. Sistem ini didesain untuk meningkatkan kualitas proses penjaminan mutu di tingkat jurusan Ilmu Komputer.

Pengembangan sistem ini akan berfokus pada dua aspek utama yaitu manajemen *user* dan pengelolaan *file* cetak laporan. Untuk manajemen *user*, akan menambahkan *role* baru pada sistem tersebut. Sementara untuk pengelolaan *file* laporan, sistem akan menyediakan fungsionalitas cetak dalam bentuk pdf.

Metodologi yang akan digunakan dalam pengerjaan proyek ini adalah *Rapid Application Development* (RAD). Proses pengembangan akan dimulai dari tahap

perencanaan kebutuhan, dilanjutkan dengan tahap desain, kemudian tahap pengembangan, dan diakhiri dengan tahap implementasi.

Estimasi siklus pengembangan yang akan dijalankan adalah empat hingga lima iterasi untuk mencapai hasil yang optimal sesuai kebutuhan *stakeholder*. Setiap iterasi akan melibatkan umpan balik dari *stakeholder* di jurusan Ilmu Komputer untuk memastikan sistem yang dikembangkan sudah memenuhi kebutuhan penjaminan mutu jurusan.

Hasil akhir dari proyek ini adalah dapat menghasilkan sistem penjamin mutu adalah mampu meningkatkan efisiensi proses penjaminan mutu di jurusan Ilmu Komputer. Sistem ini akan memudahkan pengelolaan *user* dan dokumen terkait penjaminan mutu, sehingga dapat mendukung peningkatan kualitas pendidikan di jurusan Ilmu Komputer secara berkelanjutan.

3.5.3 Kebutuhan Bisnis

Sistem yang dibutuhkan harus mampu mengelola pengguna untuk mendukung proses penjaminan mutu di Jurusan Ilmu Komputer Universitas Lampung. Fitur utama yang diperlukan adalah kemampuan untuk mengelola berbagai peran pengguna dalam sistem. Hal ini mencakup fleksibilitas untuk menambahkan peran baru sesuai dengan perkembangan kebutuhan organisasi. Dengan adanya manajemen pengguna yang baik, dapat meningkatkan efektivitas sistem penjaminan mutu di jurusan.

Selain manajemen pengguna, sistem juga dituntut untuk dapat mengelola *file* laporan dengan baik. Ini mencakup pengelolaan laporan CPMK (Capaian Pembelajaran Mata Kuliah) dan CPL (Capaian Pembelajaran Lulusan). Sistem harus mampu menyimpan, mengorganisir, dan memvisualisasikan data dari laporan-laporan tersebut. Lebih lanjut, sistem dapat menghasilkan laporan dalam format PDF yang siap cetak. Dengan demikian, proses pelaporan dan dokumentasi untuk keperluan penjaminan mutu dapat dilakukan secara lebih terstruktur.

3.5.4 Kebutuhan Fungsional dan Non-Fungsional

Dalam pengembangan Sistem Penjamin Mutu Jurusan Ilmu Komputer Modul Manajemen *User* dan *File* Laporan terdiri dari kebutuhan fungsional dan kebutuhan

non-fungsional. Berikut adalah kebutuhan fungsional dan kebutuhan non-fungsional yang bisa dilihat pada Tabel 3.2 untuk kebutuhan fungsional dan Tabel 3.3 untuk kebutuhan non-fungsional.

1. Kebutuhan Fungsional

Tabel 3.2 Tabel Kebutuhan Fungsional

No	ID	Kebutuhan Fungsional	Catatan
1	FR-001	Sistem harus menampilkan daftar berita yang dapat diakses oleh tamu.	UC-001
2	FR-002	Sistem harus menyediakan <i>form</i> registrasi universitas untuk tamu.	UC-002
3	FR-003	Sistem harus menyediakan <i>form login</i> dan autentikasi.	UC-003
4	FR-004	Sistem harus menyediakan fitur CRUD berita untuk otoritas Admin	UC-004
5	FR-005	Sistem harus menyediakan manajemen data pendaftaran universitas.	UC-005
6	FR-006	Sistem harus menyediakan manajemen <i>user</i> termasuk penambahan, pengeditan, dan penghapusan.	UC-006
7	FR-007	Sistem harus menyediakan pengaturan antarmuka atau elemen UI tertentu.	UC-007
8	FR-008	Sistem harus memungkinkan <i>user</i> untuk mengelola profil.	UC-008
9	FR-009	Sistem harus menyediakan manajemen prodi termasuk penambahan dan penghapusan.	UC-009
10	FR-010	Sistem harus menyediakan manajemen fakultas termasuk penambahan dan penghapusan.	UC-010
11	FR-011	Sistem harus menyediakan manajemen universitas termasuk penambahan dan penghapusan.	UC-011
12	FR-012	Sistem harus menyediakan fitur Melihat Mahasiswa.	UC-012
13	FR-013	Sistem harus menyediakan fitur Mengelola Mahasiswa.	UC-013
14	FR-014	Sistem harus menampilkan Visualisasi CPL per Mahasiswa.	UC-014
15	FR-015	Sistem harus menghasilkan dan mencetak laporan Visualisasi CPL per mahasiswa dalam format PDF.	UC-015

No	ID	Kebutuhan Fungsional	Catatan
16	FR-016	Sistem harus menampilkan Visualisasi CPMK per Mahasiswa.	UC-016 dan UC-024
17	FR-017	Sistem harus menghasilkan dan mencetak laporan Visualisasi CPMK per mahasiswa dalam format PDF.	UC-017 dan UC-025
18	FR-018	Sistem harus menampilkan Visualisasi CPL per Angkatan.	UC-018
19	FR-019	Sistem harus menghasilkan dan mencetak laporan Visualisasi CPL per Angkatan dalam format PDF.	UC-019
20	FR-020	Sistem harus menampilkan Visualisasi CPMK per Angkatan.	UC-020
21	FR-021	Sistem harus menghasilkan dan mencetak laporan Visualisasi CPMK per Angkatan dalam format PDF.	UC-021
22	FR-022	Sistem harus menampilkan Visualisasi CPMK per Mata Kuliah.	UC-022
23	FR-023	Sistem harus menghasilkan dan mencetak laporan Visualisasi CPMK per Mata Kuliah dalam format PDF.	UC-023

2. Kebutuhan Non-Fungsional

Tabel 3.3 Tabel Kebutuhan Non Fungsional

No	ID	Kebutuhan Non-Fungsional
1	NFR-001	Sistem harus memberikan umpan balik (misalnya, notifikasi sukses, pesan kesalahan yang informatif) yang jelas terhadap setiap aksi pengguna, sehingga memandu pengguna dalam berinteraksi dengan sistem.
2	NFR-002	Sistem harus menerapkan <i>password</i> yang kuat untuk akun pengguna, yang mencakup persyaratan kompleksitas minimal (panjang minimal 8 karakter, serta kombinasi huruf besar, huruf kecil, angka, dan simbol) dan memastikan penyimpanan <i>password</i> di dalam basis data dilakukan secara aman menggunakan teknik <i>hashing</i> .
3	NFR-003	Sistem harus menyediakan jalur akses yang terstruktur dan mekanisme navigasi yang konsisten bagi pengguna untuk menjangkau semua fitur yang sesuai dengan hak aksesnya.

No	ID	Kebutuhan Non-Fungsional
4	NFR-004	Sistem harus mampu menjaga stabilitas dan kinerja di bawah beban pengguna simultan (<i>load testing</i>), serta memberikan waktu respons (<i>response time</i>) yang cepat dan wajar untuk setiap interaksi pengguna.

Untuk memenuhi kebutuhan keamanan (NFR-002), sistem dirancang dengan mengimplementasikan mekanisme keamanan standar yang disediakan oleh *framework* Laravel. Kata sandi pengguna diamankan menggunakan *hashing* Bcrypt secara otomatis. Untuk mencegah serangan *SQL Injection*, seluruh interaksi dengan basis data akan memanfaatkan Eloquent ORM (Object-Relational Mapping) Laravel yang secara inheren menggunakan parameter *binding*. Kontrol akses berbasis peran (RBAC) akan diimplementasikan menggunakan kombinasi *middleware* dan *gates* untuk memastikan setiap pengguna hanya dapat mengakses fungsi yang sesuai dengan otoritasnya.

3.5.5 Nilai Bisnis

Nilai bisnis yang dapat dihasilkan dari sistem ini dapat dibagi menjadi dua kategori, yaitu *tangible* dan *intangible*. Berikut adalah penjelasannya:

1. *Tangible*

1. Efisiensi operasional

Dengan adanya modul manajemen *user* dan *file* laporan, jurusan dapat mengoptimalkan pengelolaan akses pengguna. Hal ini menghemat waktu dan sumber daya dalam mengelola hak akses, menyimpan, dan mencetak laporan-laporan visualisasi CPL dan CPMK.

2. Peningkatan keamanan data

Sistem ini memungkinkan pengelolaan *user* yang lebih terstruktur, mengurangi risiko akses tidak sah, dan meningkatkan integritas data. Pengelolaan *file* laporan visualisasi yang sistematis juga mengurangi kemungkinan kehilangan atau kesalahan dalam penanganan laporan penting.

3. Optimalisasi penggunaan sumber daya

Dengan modul *file* laporan yang, jurusan dapat mengurangi penggunaan kertas dan ruang penyimpanan fisik, sekaligus meningkatkan aksesibilitas terhadap informasi penting.

2. *Intangible*

1. Peningkatan transparansi dan akuntabilitas

Sistem manajemen *user* yang baik memastikan bahwa hanya personel yang berwenang yang dapat mengakses informasi sensitif. Pengelolaan *file* laporan yang terstruktur juga meningkatkan transparansi dalam proses penjaminan mutu, memudahkan pelaporan kepada pemangku kepentingan.

2. Peningkatan kolaborasi dan komunikasi:

Dengan sistem manajemen *user* yang terintegrasi, staf dan dosen dapat berkolaborasi lebih efektif dalam proses penjaminan mutu. Akses yang terorganisir ke *file* laporan juga memfasilitasi berbagi informasi yang lebih baik antar tim dan departemen.

3. Peningkatan kualitas pengambilan keputusan:

Akses cepat dan mudah ke laporan-laporan penjaminan mutu memungkinkan pimpinan jurusan untuk membuat keputusan berbasis data dengan lebih efektif. Ini dapat mendukung peningkatan berkelanjutan dalam kualitas pendidikan dan operasional jurusan.

3.5.6 Definisi dan Strategi Analisis Kebutuhan

Untuk mendapatkan informasi mengenai kebutuhan sistem pada penelitian ini dilakukan dengan cara melalui wawancara kepada calon pengguna dimana hasil wawancara dituangkan pada tabel Matris Kebutuhan Fungsional dan tabel Matriks Kebutuhan Non-Fungsional proses *As-Is* dan proses *To-Be* yang ditunjukkan pada Tabel 3.3 dan Tabel 3.4.

Tabel 3.4 Matriks Kebutuhan Fungsional *As-Is* dan *To-Be*

Kebutuhan Fungsional		
No	Proses Berjalan (<i>As-Is</i>)	Sistem yang akan dibuat (<i>To-Be</i>)
1	Manajemen <i>user</i> masih terbatas, yang mengakibatkan kesulitan dalam mengatur	Sistem memiliki otoritas baru berupa Admin Universitas, Wakil Rektor, Wakil

Kebutuhan Fungsional		
No	Proses Berjalan (<i>As-Is</i>)	Sistem yang akan dibuat (<i>To-Be</i>)
	akses dan peran berbagai <i>stakeholder</i> . Manajemen pengguna hanya ada Admin, Dosen, dan Penjamin Mutu.	Dekan, Kepala Program Studi Penjamin Mutu Universitas, Penjamin Mutu Fakultas, dan Penjamin Mutu Program Studi.
2	Sistem belum memiliki fitur untuk menghasilkan laporan visualisasi yang bisa dicetak atau diunduh. Laporan yang tersedia di sistem saat ini hanya berupa data visualisasi yang hanya bisa ditampilkan pada sistem saja, sehingga tidak dapat dicetak.	Sistem memiliki fitur untuk menghasilkan laporan visualisasi dalam bentuk PDF, yang disesuaikan dengan kebutuhan pemangku kepentingan. Setiap <i>user</i> dengan otoritas berbeda (Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen) akan mendapatkan laporan yang sesuai dengan kebutuhan.

Tabel 3.5 Matriks Kebutuhan Non-Fungsional *As-Is* dan *To-Be*

Kebutuhan Non-Fungsional		
No	Proses Berjalan (<i>As-Is</i>)	Sistem yang akan dibuat (<i>To-Be</i>)
1	<i>Operational Requirements</i>	
	- Struktur pengelolaan akses <i>user</i> masih sederhana dan belum mencakup semua jenjang otoritas yang dibutuhkan. - Proses pencetakan laporan visualisasi CPL dan CPMK belum tersedia dalam sistem.	- Sistem menyediakan mekanisme pengelolaan <i>user</i> yang mendukung berbagai peran dan jenjang otoritas (Admin Universitas, Wakil Rektor, Wakil Dekan, Kepala Program Studi, Penjamin Mutu Universitas, Fakultas, dan Program Studi). - Sistem menyediakan fungsionalitas untuk menghasilkan dan mencetak <i>file</i> laporan visualisasi CPL dan CPMK dalam format PDF.
2	<i>Performance Requirements</i>	
	Pengelolaan peran dan hak akses pengguna dilakukan dengan cara yang mungkin memerlukan intervensi manual bertingkat.	Sistem dirancang untuk merespons permintaan pengelolaan <i>user</i> dan pembuatan laporan PDF sesuai dengan alur kerja yang telah ditentukan.

Kebutuhan Non-Fungsional		
No	Proses Berjalan (<i>As-Is</i>)	Sistem yang akan dibuat (<i>To-Be</i>)
	– Pembuatan <i>file</i> laporan dari data visualisasi dilakukan secara manual di luar sistem. – Kinerja dan stabilitas sistem di bawah beban pengguna simultan belum terukur dan tidak terjamin.	– Sistem memiliki fitur terintegrasi untuk menghasilkan <i>file</i> laporan visualisasi dalam format PDF langsung dari data yang ada di sistem. – Sistem dirancang dan diuji untuk mampu menjaga stabilitas dan kinerja yang responsif ketika diakses oleh banyak pengguna secara bersamaan (NFR-004).
3	<i>Reliability Requirements</i>	
	– Potensi kesalahan dalam pengelolaan <i>user</i> dan pembuatan laporan lebih besar karena proses manual.	– Sistem dirancang untuk bekerja sesuai dengan fungsionalitas yang telah diuji (<i>black-box testing</i>). <i>File</i> laporan yang dihasilkan dalam format PDF dapat dicetak dan diarsipkan secara konsisten.
4	<i>Safety/Security Requirements</i>	
	– Perlindungan data <i>user</i> dan <i>file</i> laporan dari perubahan yang tidak diinginkan bergantung pada prosedur manual. – Kontrol akses terhadap data dan fitur belum sepenuhnya mengakomodasi semua jenjang peran.	– Sistem menerapkan kebijakan kata sandi yang kuat (panjang minimal 8 karakter, kombinasi huruf besar, kecil, angka, simbol) dan penyimpanan kata sandi menggunakan teknik hashing (NFR-002). – Sistem menerapkan kontrol akses berbasis peran yang terdefinisi, di mana setiap peran hanya dapat mengakses fitur dan data yang sesuai dengan otoritasnya.
5	<i>Security Requirements</i>	
	– Pengguna belum dapat mencetak laporan visualisasi CPL dan CPMK langsung dari sistem dalam format PDF.	– Pengguna dapat mengakses dan mencetak laporan visualisasi CPL dan CPMK dalam format PDF langsung dari sistem.

Kebutuhan Non-Fungsional	
No	Proses Berjalan (<i>As-Is</i>)
	Sistem yang akan dibuat (<i>To-Be</i>)
	– Sistem memberikan umpan balik yang jelas terhadap aksi pengguna dan memiliki mekanisme navigasi yang konsisten untuk menjangkau fitur (NFR-001, NFR-003).

3.5.7 Product Backlog

Product backlog merupakan daftar menyeluruh dari semua fitur dan tugas yang direncanakan pada suatu produk. Tabel 3.6 berikut ini menyajikan daftar fitur yang akan dikembangkan dalam proyek atau penelitian ini.

Tabel 3.6 *Product Backlog*

No	Item	Deskripsi	Estimasi	PIC	Capaian
1	Membuat <i>Use Case Diagram, Activity Diagram</i>	Membuat UML agar memahami fungsionalitas sistem.	15 hari	Dhiaurrahman Raziq Ramadhan dan Naufal Hilal	Mendokumentasikan pengembangan sistem
2	Membuat ERD Sistem	Membuat ERD untuk memahami hubungan antar sistem.	5 hari	Dhiaurrahman Raziq Ramadhan dan Naufal Hilal	Mendokumentasikan pengembangan sistem
3	Mendesain halaman Berita	Pengembangan tampilan halaman dilakukan langsung di sistem menggunakan <i>live prototyping</i> .	5 hari	Dhiaurrahman Raziq Ramadhan	Melihat Berita (UC-001) dan Mengelola Berita (UC-004)

4	Mendesain halaman <i>login</i>	Pengembangan tampilan halaman dilakukan langsung di sistem menggunakan <i>live prototyping</i> .	5 hari	Dhiaurrahman Raziq Ramadhan	Melakukan <i>login</i> (UC-003)
5	Mendesain halaman Registrasi Universitas	Pengembangan tampilan halaman dilakukan langsung di sistem menggunakan <i>live prototyping</i> .	5 hari	Dhiaurrahman Raziq Ramadhan	Menambahkan Registrasi Universitas (UC-002) dan Mengelola Registrasi Universitas (UC-005)
6	Mendesain halaman <i>User</i>	Pengembangan tampilan halaman dilakukan langsung di sistem menggunakan <i>live prototyping</i> .	5 hari	Dhiaurrahman Raziq Ramadhan	Mengelola <i>User</i> (UC-006)
7	Mendesain halaman Tema Universitas	Pengembangan tampilan halaman dilakukan langsung di sistem menggunakan <i>live prototyping</i> .	5 hari	Dhiaurrahman Raziq Ramadhan	Mengelola Tema Universitas (UC-005)

8	Mendesain halaman Profil	Pengembangan tampilan halaman dilakukan langsung di sistem menggunakan <i>live prototyping</i> .	5 hari	Dhiaurrahman Raziq Ramadhan	Mengelola Profil (UC-008)
9	Mendesain halaman Prodi	Pengembangan tampilan halaman dilakukan langsung di sistem menggunakan <i>live prototyping</i> .	5 hari	Dhiaurrahman Raziq Ramadhan	Mengelola Prodi (UC-009)
10	Mendesain halaman Fakultas	Pengembangan tampilan halaman dilakukan langsung di sistem menggunakan <i>live prototyping</i> .	5 hari	Dhiaurrahman Raziq Ramadhan	Mengelola Fakultas (UC-010)
11	Mendesain halaman Universitas	Pengembangan tampilan halaman dilakukan langsung di sistem menggunakan <i>live prototyping</i> .	5 hari	Dhiaurrahman Raziq Ramadhan	Mengelola Universitas (UC-011)

12	Mendesain halaman Mahasiswa	Pengembangan tampilan halaman dilakukan langsung di sistem menggunakan live prototyping.	5 hari	Dhiaurrahman Raziq Ramadhan	Melihat Mahasiswa (UC-012), Mengelola Mahasiswa (UC-013)
13	Mendesain halaman Visualisasi CPL per Mahasiswa	Pengembangan tampilan halaman dilakukan langsung di sistem menggunakan live prototyping.	5 hari	Dhiaurrahman Raziq Ramadhan	Melihat Visualisasi CPL per Mahasiswa (UC-014) dan Mencetak Laporan Visualisasi CPL per Mahasiswa (UC-015)
10	Mendesain halaman Visualisasi CPMK per Mahasiswa	Pengembangan tampilan halaman dilakukan langsung di sistem menggunakan live prototyping.	5 hari	Dhiaurrahman Raziq Ramadhan	Melihat Visualisasi CPMK per Mahasiswa (UC-016), Mencetak laporan Visualisasi CPMK per Mahasiswa (UC-017), Melihat Visualisasi CPMK per Mahasiswa (UC-024) dan Mencetak laporan Visualisasi CPMK per Mahasiswa (UC-025)
12	Mendesain halaman Visualisasi CPL per Angkatan	Pengembangan tampilan halaman dilakukan langsung di sistem menggunakan	5 hari	Dhiaurrahman Raziq Ramadhan	Melihat Visualisasi CPL per Angkatan (UC-018), Mencetak laporan Visualisasi CPL per Angkatan (UC-019)

		<i>live prototyping.</i>			
13	Mendesain halaman CPMK per Angkatan	Pengembangan tampilan halaman dilakukan langsung di sistem menggunakan <i>live prototyping.</i>	5 hari	Dhiaurrahman Raziq Ramadhan	Melihat Visualisasi CPMK per Angkatan (UC-020), Mencetak laporan Visualisasi CPMK per Angkatan (UC-021)
14	Mendesain halaman CPMK per Mata Kuliah	Pengembangan tampilan halaman dilakukan langsung di sistem menggunakan <i>live prototyping.</i>	5 hari	Dhiaurrahman Raziq Ramadhan	Melihat Visualisasi CPMK per Mata Kuliah (UC-022), Mencetak laporan Visualisasi CPMK per Mata Kuliah (UC-023)
15	Pengembangan halaman Berita	Implementasi frontend dan backend untuk fitur berita		Dhiaurrahman Raziq Ramadhan	Melihat Berita (UC-001) dan Mengelola Berita (UC-004)
16	Pengembangan halaman <i>login</i>	Implementasi sistem login dengan autentikasi <i>user</i>		Dhiaurrahman Raziq Ramadhan	Melakukan <i>login</i> (UC-003)
17	Pengembangan halaman Registrasi Universitas	Implementasi fitur registrasi universitas		Dhiaurrahman Raziq Ramadhan	Menambahkan Registrasi Universitas (UC-002) dan Mengelola Registrasi Universitas (UC-005)

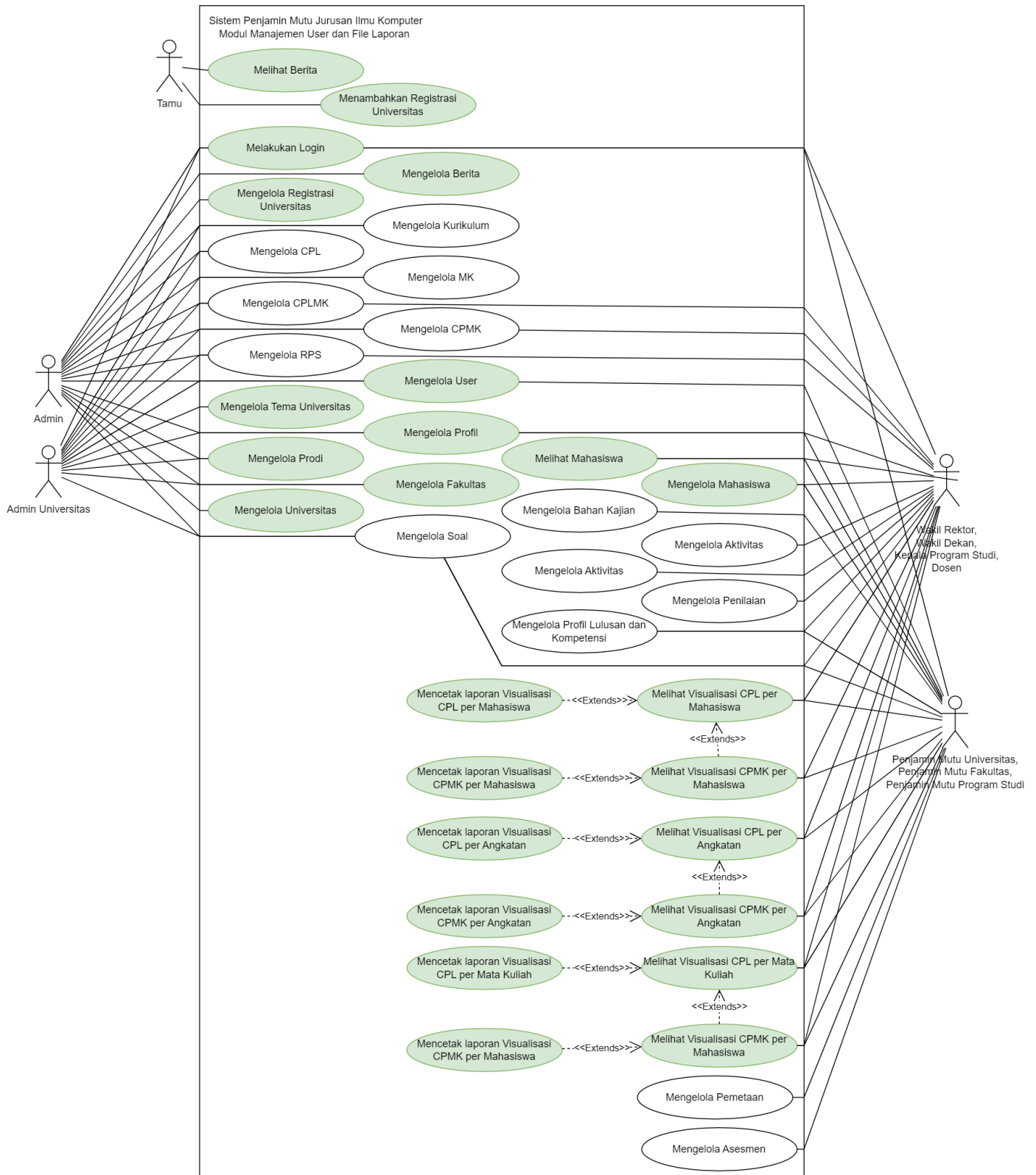
18	Pengembangan halaman <i>User</i>	Implementasi sistem manajemen <i>user</i> dan multi otoritas	Dhiaurrahman Raziq Ramadhan	Mengelola <i>User</i> (UC-006)
19	Pengembangan halaman Tema Universitas	Implementasi fitur pengelolaan tema universitas	Dhiaurrahman Raziq Ramadhan	Mengelola Tema Universitas (UC-005)
20	Pengembangan halaman Profil	Implementasi pengelolaan profil <i>user</i>	Dhiaurrahman Raziq Ramadhan	Mengelola Profil (UC-008)
21	Pengembangan halaman Prodi	Implementasi pengelolaan data prodi	Dhiaurrahman Raziq Ramadhan	Mengelola Prodi (UC-009)
22	Pengembangan halaman Fakultas	Implementasi pengelolaan data fakultas	Dhiaurrahman Raziq Ramadhan	Mengelola Fakultas (UC-010)
23	Pengembangan halaman Universitas	Implementasi pengelolaan data universitas	Dhiaurrahman Raziq Ramadhan	Mengelola Universitas (UC-011)
24	Pengembangan halaman Mahasiswa	Implementasi pengelolaan data mahasiswa	Dhiaurrahman Raziq Ramadhan	Melihat Mahasiswa (UC-012), Mengelola Mahasiswa (UC-013)
25	Pengembangan halaman Visualisasi	Implementasi visualisasi dan pelaporan CPL per mahasiswa	Dhiaurrahman Raziq Ramadhan	Melihat Visualisasi CPL per Mahasiswa (UC-014) dan Mencetak Laporan Visualisasi CPL per Mahasiswa (UC-015)

	CPL per Mahasiswa			
26	Pengembangan halaman Visualisasi CPMK per Mahasiswa	Implementasi visualisasi dan pelaporan CPMK per mahasiswa	Dhiaurrahman Raziq Ramadhan	Melihat Visualisasi CPMK per Mahasiswa (UC-016), Mencetak laporan Visualisasi CPMK per Mahasiswa (UC-017), Melihat Visualisasi CPMK per Mahasiswa (UC-024) dan Mencetak laporan Visualisasi CPMK per Mahasiswa (UC-025)
27	Pengembangan halaman Visualisasi CPL per Angkatan	Implementasi visualisasi dan pelaporan CPL per angkatan	Dhiaurrahman Raziq Ramadhan	Melihat Visualisasi CPL per Angkatan (UC-018), Mencetak laporan Visualisasi CPL per Angkatan (UC-019)
28	Pengembangan halaman CPMK per Angkatan	Implementasi visualisasi dan pelaporan CPMK per angkatan	Dhiaurrahman Raziq Ramadhan	Melihat Visualisasi CPMK per Angkatan (UC-020), Mencetak laporan Visualisasi CPMK per Angkatan (UC-021)
29	Pengembangan halaman CPMK per Mata Kuliah	Implementasi visualisasi dan pelaporan CPMK per mata kuliah	Dhiaurrahman Raziq Ramadhan	Melihat Visualisasi CPMK per Mata Kuliah (UC-022), Mencetak laporan Visualisasi CPMK per Mata Kuliah (UC-023)

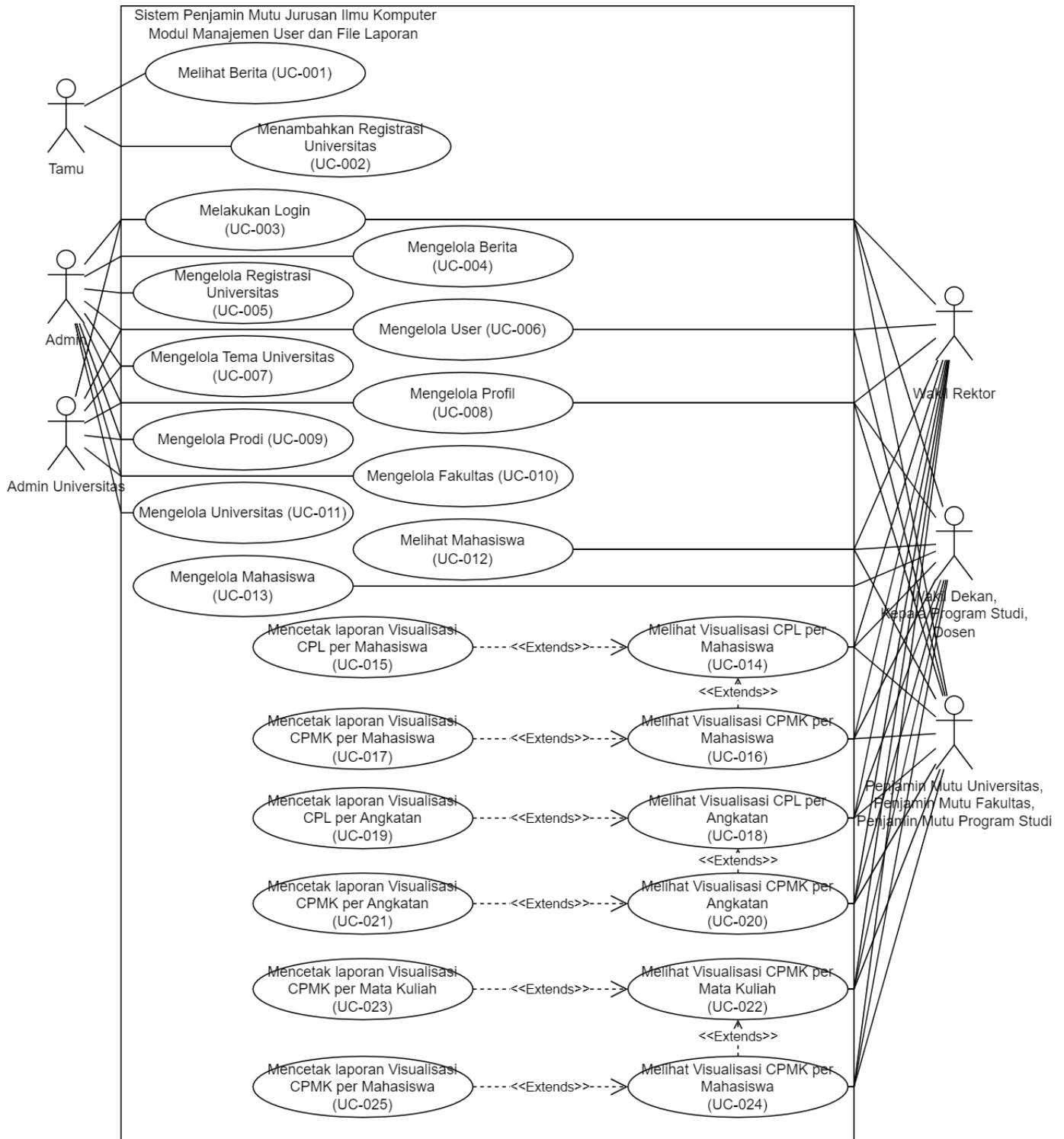
3.5.8 Use Case Diagram

Use case diagram merupakan salah satu diagram dalam *Unified Modelling Language* (UML) yang merefleksikan kebutuhan fungsional dari sebuah sistem perangkat lunak dari sudut pandang pengguna. Diagram ini terdiri dari tiga komponen utama: aktor yang merepresentasikan pengguna atau sistem lain yang berinteraksi, *use case* yang menggambarkan fungsionalitas yang disediakan sistem, dan relasi yang menunjukkan hubungan antara elemen-elemen tersebut. Dalam penelitian ini, *use case* diagram yang ditampilkan pada Gambar 3.3 memberikan gambaran umum mengenai seluruh fungsi utama dari Sistem Penjamin Mutu Jurusan Ilmu Komputer, khususnya pada modul Manajemen Pengguna dan *File* Laporan. Diagram ini secara visual memetakan interaksi antara berbagai aktor dengan fitur-fitur yang tersedia.

Aktor-aktor yang terlibat dalam sistem ini mencerminkan struktur hierarki yang ada di lingkungan perguruan tinggi, dimulai dari Tamu yang memiliki akses publik untuk melihat berita dan melakukan registrasi universitas baru. Selanjutnya, terdapat Admin dengan kewenangan global untuk mengelola data master seperti universitas, registrasi, dan berita. Di tingkat institusi, terdapat Admin Universitas yang bertugas mengelola pengguna, program studi, fakultas, dan kustomisasi tema untuk universitasnya masing-masing. Sementara itu, peran-peran seperti Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, serta tim Penjamin Mutu (di tingkat Universitas, Fakultas, dan Program Studi) merupakan konsumen utama dari data visualisasi dan laporan capaian pembelajaran.



Gambar 3.2 Use Case Diagram Global Sistem Penjamin Mutu Jurusan Ilmu Komputer.



Gambar 3.3 Use Case Diagram Sistem Penjamin Mutu Jurusan Ilmu Komputer.

Use case pada Gambar merupakan fitur yang dikembangkan pada penelitian ini. Untuk deskripsi dari *use case* akan dijelaskan beberapa tabel berikut ini:

1. Melihat berita (UC-001)

Tabel 3.7 Deskripsi *Use Case* Melihat Berita

Nama <i>Use Case</i>: Melihat berita	ID: UC-001
Aktor: Tamu	
Deskripsi: <i>Use case</i> ini menggambarkan proses pengguna (Tamu) dalam melihat berita yang tersedia di sistem. Pengguna dapat memilih menu "Berita" untuk menampilkan daftar berita yang telah dibuat oleh Admin.	
Pemicu (<i>Trigger</i>): Pengguna mengklik menu "Berita" di navbar.	
Kondisi Awal (<i>Preconditions</i>): Sistem memiliki setidaknya satu berita yang dapat ditampilkan.	
Alur Normal (<i>Normal Course</i>): 1. Pengguna mengakses sistem melalui browser. 2. Sistem menampilkan halaman yang sedang diakses oleh pengguna. 3. Pengguna mengklik menu "Berita" di navbar. 4. Sistem menampilkan daftar berita kepada pengguna. 5. Pengguna memilih salah satu berita yang tersedia. 6. Sistem menampilkan isi berita yang dipilih. 7. Pengguna melihat berita yang telah dipilih.	
Kondisi Akhir (<i>Postconditions</i>): 1. Sistem menampilkan halaman berita yang telah dipilih.	

2. Menambahkan Registrasi Universitas (UC-002)

Tabel 3.8 Deskripsi *Use Case* Menambahkan Registrasi Universitas

Nama <i>Use Case</i>: Menambahkan registrasi universitas	ID: UC-002
Aktor: Tamu	
Deskripsi: <i>Use case</i> ini menggambarkan proses registrasi universitas baru yang dilakukan oleh pengguna (tamu) yang ingin bergabung dan menggunakan sistem. Setelah mengisi form registrasi, data universitas akan disimpan dalam sistem dan menunggu proses verifikasi dari admin sebelum dapat digunakan.	

Pemicu (Trigger): Pengguna mengklik menu "Bergabung" di navbar.
Kondisi Awal (Preconditions): Pengguna telah mengakses sistem melalui browser.
Alur Normal (Normal Course): 1. Pengguna mengakses sistem melalui browser 2. Sistem menampilkan halaman yang sedang diakses oleh pengguna. 3. Pengguna mengklik menu "Bergabung" di navbar. 4. Sistem menampilkan halaman registrasi universitas. 5. Pengguna mengisi form registrasi universitas dan mengklik tombol Submit untuk mengirimkan form. 6. Sistem menampilkan notifikasi bahwa registrasi berhasil dan sedang menunggu verifikasi melalui email.
Kondisi Akhir (Postconditions): 1. Sistem menambahkan registrasi universitas di <i>database</i>. 2. Sistem menampilkan notifikasi bahwa registrasi universitas berhasil. 3. Pengguna harus menunggu email konfirmasi setelah proses verifikasi admin.

3. Melakukan *Login* (UC-003)

Tabel 3.9 Deskripsi *Use Case* Melakukan *Login*

Nama Use Case: Mengelola <i>Login</i>	ID: UC-003
Aktor: Admin, Admin Universitas, Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi.	
Deskripsi: <i>Use case</i> ini menggambarkan proses melakukan <i>Login</i> .	
Pemicu (Trigger): Pengguna menuju halaman <i>Login</i> .	
Kondisi Awal (Preconditions): Pengguna telah mengakses sistem melalui browser.	
Alur Normal (Normal Course): 1. Pengguna mengakses sistem melalui browser. 2. Sistem menampilkan halaman yang sedang diakses oleh pengguna. 3. Pengguna mengklik halaman menu "Login" di navbar. 4. Sistem menampilkan halaman <i>login</i> 5. Pengguna mengisi form login dan mengklik tombol "Log in" untuk mengirimkan form 6. Sistem menampilkan halaman dashboard Sistem Penjamin Mutu.	
Kondisi Akhir (Postconditions):	

1. Sistem menampilkan halaman dashboard Sistem Penjamin Mutu.

4. Mengelola Berita (UC-004)

Tabel 3.10 Deskripsi *Use Case* Mengelola Berita

Nama Use Case: Mengelola Berita	ID: UC-004
Aktor: Admin	
Deskripsi: <i>Use case</i> ini menggambarkan proses pengelolaan berita yang dilakukan oleh pengguna (Admin), termasuk menambahkan, melihat, mengubah, dan menghapus berita yang ditampilkan pada sistem.	
Pemicu (Trigger): Pengguna memilih menu "Berita" di sidebar.	
Kondisi Awal (Preconditions): Pengguna telah masuk ke dalam sistem dengan otoritas Admin	
Alur Normal (Melihat Berita): 1. Pengguna membuka <i>sidebar</i> "Berita" dan memilih submenu "Daftar Berita". 2. Sistem menampilkan daftar berita yang tersedia.	
Alur Normal (Menambahkan Berita): 1. Pengguna mengakses <i>sidebar</i> "Berita" dan memilih submenu "Tambah Berita". 2. Sistem menampilkan <i>form</i> tambah berita. 3. Pengguna mengisi <i>form</i> berita dan mengklik Simpan. 4. Sistem menampilkan notifikasi bahwa berita berhasil ditambahkan.	
Alur Alternatif 1 (Mengubah Berita) 1. Pengguna mengakses <i>sidebar</i> "Berita" dan memilih submenu "Daftar Berita". 2. Sistem menampilkan daftar berita yang tersedia. 3. Pengguna memilih berita yang ingin diperbarui dengan mengklik tombol "Edit". 4. Sistem menampilkan <i>form</i> edit berita dengan data yang sudah ada. 5. Pengguna mengubah data berita dan mengklik Simpan. 6. Sistem mengubah berita dan menampilkan notifikasi bahwa berita berhasil diperbarui.	
Alur Alternatif 2 (Menghapus Berita) 1. Pengguna mengakses <i>sidebar</i> "Berita" dan memilih submenu "Daftar Berita". 2. Sistem menampilkan daftar berita yang tersedia. 3. Pengguna memilih berita yang ingin dihapus dengan mengklik tombol "Delete". 4. Sistem menampilkan konfirmasi penghapusan. 5. Pengguna mengonfirmasi penghapusan.	

6. Sistem menghapus berita dan menampilkan notifikasi bahwa berita telah dihapus.
Kondisi Akhir (<i>Postconditions</i>):
1. Jika Pengguna menambahkan berita, berita baru tersimpan dan ditampilkan di daftar berita.
2. Jika Pengguna mengubah berita, berita yang diedit akan diperbarui di sistem.
3. Jika Pengguna menghapus berita, berita tersebut akan dihapus dari sistem.
4. Sistem menampilkan notifikasi sukses setiap kali tindakan berhasil dilakukan.

5. Mengelola Registrasi Universitas (UC-005)

Tabel 3.11 Deskripsi *Use Case* Mengelola Registrasi Universitas

Nama <i>Use Case</i>: Mengelola Registrasi Universitas	ID: UC-005
Aktor: Admin	
Deskripsi: <i>Use case</i> ini menggambarkan proses yang dilakukan oleh pengguna (Admin) dalam mengelola registrasi universitas yang ingin bergabung ke dalam sistem. Pengguna (Admin) dapat melihat daftar universitas yang telah mendaftar, menyetujui registrasi yang memenuhi persyaratan, atau menolak registrasi dengan memberikan alasan penolakan.	
Pemicu (<i>Trigger</i>): Pengguna memilih menu "Registrasi Universitas" di <i>sidebar</i> .	
Kondisi Awal (<i>Preconditions</i>): Pengguna telah masuk ke dalam sistem dengan otoritas Admin	
Alur Normal (Melihat Daftar Registrasi Universitas): 1. Pengguna memilih menu "Registrasi Universitas". 2. Sistem menampilkan daftar universitas yang telah mendaftar.	
Alur Normal (Menyetujui Registrasi Universitas): 1. Pengguna memilih menu "Registrasi Universitas". 2. Sistem menampilkan daftar universitas yang telah mendaftar. 3. Pengguna memilih registrasi universitas yang ingin ditinjau dengan mengklik tombol "Detail". 4. Sistem menampilkan halaman detail registrasi universitas. 5. Pengguna mengklik tombol "Approve" 6. Sistem menampilkan dialog konfirmasi penyetujuan. 7. Pengguna mengonfirmasi penyetujuan 8. Sistem menyetujui registrasi universitas dan menampilkan notifikasi bahwa pendaftaran universitas telah disetujui.	
Alur Alternatif (Menolak Registrasi Universitas): 1. Pengguna memilih menu "Registrasi Universitas".	

2. Sistem menampilkan daftar universitas yang telah mendaftar. 3. Pengguna memilih registrasi universitas yang ingin ditinjau dengan mengklik tombol "Detail". 4. Sistem menampilkan halaman detail registrasi universitas. 5. Pengguna mengklik tombol "Reject". 6. Sistem menampilkan dialog form alasan penolakan. 7. Pengguna mengisi form alasan penolakan dan mengklik tombol "Tolak". 8. Sistem menolak registrasi universitas dan menampilkan notifikasi bahwa pendaftaran universitas ditolak.
Kondisi Akhir (Postconditions): 1. Jika Pengguna menyetujui registrasi universitas, universitas tersebut resmi terdaftar dalam sistem. 2. Jika Pengguna menolak registrasi universitas, universitas tersebut tidak ditambahkan ke sistem, dan sistem memberikan notifikasi kepada pengguna. 3. Sistem menampilkan notifikasi sukses setelah tindakan dilakukan.

6. Mengelola User (UC-006)

Tabel 3.12 Deskripsi Use Case Mengelola User

Nama Use Case: Mengelola User	ID: UC-006
Aktor: Admin, Admin Universitas	
Deskripsi: Use case ini menggambarkan proses pengelolaan user yang dilakukan oleh pengguna (Admin dan Admin Universitas), termasuk menambahkan, melihat, mengubah, dan menghapus user yang ditampilkan pada sistem.	
Pemicu (Trigger): Pengguna memilih menu "User" di sidebar.	
Kondisi Awal (Preconditions): Pengguna telah masuk ke dalam sistem dengan otoritas Admin atau Admin Universitas.	
Alur Normal (Melihat User): 1. Pengguna memilih menu "User" dan memilih submenu "Daftar User". 2. Sistem menampilkan daftar User yang ada.	
Alur Normal (Menambah User) 1. Pengguna memilih menu "User" dan memilih submenu "Tambah User". 2. Sistem menampilkan form tambah user. 3. Pengguna mengisi form user dan mengklik Simpan.	

4. Sistem menampilkan notifikasi bahwa <i>user</i> berhasil ditambahkan.
Alur Alternatif 1 (Mengubah <i>User</i>) 1. Pengguna mengakses sidebar “<i>User</i>” dan memilih submenu “Daftar <i>User</i>”. 2. Sistem menampilkan daftar <i>user</i> yang ada. 3. Pengguna memilih <i>user</i> yang ingin diperbarui dengan mengklik tombol “Edit”. 4. Sistem menampilkan <i>form</i> edit <i>user</i> dengan data yang sudah ada. 5. Pengguna mengubah data <i>user</i> dan mengklik Simpan. 6. Sistem mengubah <i>user</i> dan menampilkan notifikasi bahwa <i>user</i> berhasil diperbarui.
Alur Alternatif 2 (Menghapus <i>User</i>) 1. Pengguna mengakses <i>sidebar</i> “<i>User</i>” dan memilih submenu “Daftar <i>User</i>”. 2. Sistem menampilkan daftar <i>user</i> yang tersedia. 3. Pengguna memilih <i>user</i> yang ingin dihapus dengan mengklik tombol “Delete”. 4. Sistem menampilkan konfirmasi penghapusan. 5. Pengguna mengonfirmasi penghapusan. 6. Sistem menghapus <i>user</i> dan menampilkan notifikasi bahwa <i>user</i> telah dihapus.
Kondisi Akhir (<i>Postconditions</i>): 1. Jika Pengguna menambahkan <i>user</i>, <i>user</i> baru tersimpan dan ditampilkan di daftar <i>user</i>. 2. Jika Pengguna mengubah <i>user</i>, <i>user</i> yang diedit akan diperbarui di sistem. 3. Jika Pengguna menghapus <i>user</i>, <i>user</i> tersebut akan dihapus dari sistem. 4. Sistem menampilkan notifikasi sukses setiap kali tindakan berhasil dilakukan.

7. Mengelola Tema Universitas (UC-007)

Tabel 3.13 Deskripsi *Use Case* Mengelola Tema Universitas

Nama <i>Use Case</i>: Mengelola Tema Universitas	ID: UC-007
Aktor: Admin Universitas	
Deskripsi: <i>Use case</i> ini menggambarkan proses pengelolaan tampilan oleh Admin Universitas, yang mencakup perubahan warna <i>header</i> , <i>footer</i> , dan pengaturan logo universitas agar sesuai dengan identitas masing-masing universitas.	
Pemicu (<i>Trigger</i>): Pengguna memilih ikon gir pada Navbar untuk masuk ke halaman pengaturan tampilan.	
Kondisi Awal (<i>Preconditions</i>): Pengguna telah masuk ke dalam sistem dengan otoritas Admin Universitas	
Alur Normal (<i>Normal Course</i>):	

1. Pengguna memilih ikon gir pada Navbar untuk masuk ke halaman pengaturan tampilan. 2. Sistem menampilkan halaman Kustomisasi Tampilan. 3. Pengguna memilih warna tema dan logo universitas, kemudian mengklik tombol "Simpan". 4. Sistem mengubah tampilan dengan tema dan logo yang baru serta menampilkan notifikasi bahwa perubahan tampilan berhasil disimpan.
Kondisi Akhir (Postconditions): 1. Pengguna mengubah tampilan sistem, tema dan logo universitas akan diperbarui sesuai dengan pengaturan yang dipilih. 2. Sistem akan menampilkan tampilan baru dengan warna tema dan logo universitas yang telah diperbarui.

8. Mengelola Profil (UC-008)

Tabel 3.14 Deskripsi *Use Case* Mengelola Profil

Nama Use Case: Mengelola Profil	ID: UC-008
Aktor: Admin, Admin Universitas, Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi.	
Deskripsi: <i>Use case</i> ini menggambarkan proses yang dilakukan oleh pengguna dalam mengelola profil. Pengguna dapat melihat profil, mengubah profil, mengganti <i>password</i> , dan mengganti otoritas jika memiliki lebih dari satu otoritas.	
Pemicu (Trigger): Pengguna memilih foto profil pada Navbar lalu mengklik "My Profile".	
Kondisi Awal (Preconditions): Pengguna telah masuk ke dalam sistem penjamin mutu	
Alur Normal (Melihat Profil): 1. Pengguna memilih foto profil pada Navbar lalu mengklik "My Profile". 2. Sistem menampilkan halaman profil.	
Alur Alternatif 1 (Mengubah nama Profil): 1. Pengguna memilih foto profil pada Navbar lalu mengklik "My Profile". 2. Sistem menampilkan halaman profil. 3. Pengguna mengubah nama profil dan menekan tombol <i>Submit</i>. 4. Sistem menampilkan halaman profil dengan nama profil yang baru.	
Alur Alternatif 2 (Mengganti Password): 1. Pengguna memilih foto profil pada Navbar lalu mengklik "My Profile". 2. Sistem menampilkan halaman profil. 3. Pengguna mengisi <i>form</i> ganti <i>password</i> dan menekan tombol <i>Submit</i>. 4. Sistem menampilkan halaman profil dengan notifikasi <i>password</i> berhasil diganti.	

Alur Alternatif 3 (Mengganti otoritas): 1. Pengguna memilih foto profil pada Navbar lalu mengklik “My Profile”. 2. Sistem menampilkan halaman profil. 3. Pengguna mengganti otoritas yang dimiliki. 4. Sistem menampilkan halaman profil dengan notifikasi otoritas berhasil diubah.
Kondisi Akhir (Postconditions): 1. Jika pengguna hanya melihat profil, tidak ada perubahan data yang terjadi. 2. Jika pengguna mengubah profil, data profil pengguna diperbarui di sistem. 3. Jika pengguna mengganti <i>password</i>, <i>password</i> baru langsung berlaku, dan pengguna perlu menggunakannya untuk <i>login</i> berikutnya. 4. Jika pengguna mengganti otoritas, sistem akan menerapkan otoritas baru pada sesi berikutnya tanpa mengubah data otoritas pengguna lain.

9. Mengelola Prodi (UC-009)

Tabel 3.15 Deskripsi *Use Case* Mengelola Prodi

Nama Use Case: Mengelola Prodi	ID: UC-009
Aktor: Admin, Admin Universitas	
Deskripsi: <i>Use case</i> ini menggambarkan proses pengelolaan prodi yang dilakukan oleh pengguna (Admin dan Admin Universitas), termasuk menambahkan, melihat, dan menghapus prodi yang ditampilkan pada sistem.	
Pemicu (Trigger): Pengguna memilih menu "Prodi" di <i>sidebar</i> .	
Kondisi Awal (Preconditions): Pengguna telah masuk ke dalam sistem dengan otoritas Admin atau Admin Universitas.	
Alur Normal (Melihat Prodi): 1. Pengguna memilih menu "Prodi" dan memilih submenu "Daftar Prodi". 2. Sistem menampilkan daftar prodi yang ada.	
Alur Normal (Menambah Prodi) 1. Pengguna memilih menu "Prodi" dan memilih submenu "Tambah Prodi". 2. Sistem menampilkan form tambah prodi. 3. Pengguna mengisi form prodi dan mengklik Simpan. 4. Sistem menampilkan notifikasi bahwa prodi berhasil ditambahkan.	
Alur Alternatif 1 (Menghapus Prodi) 1. Pengguna mengakses sidebar “Prodi” dan memilih submenu “Daftar Prodi”. 2. Sistem menampilkan daftar prodi yang tersedia.	

3. Pengguna memilih prodi yang ingin dihapus dengan mengklik tombol “Delete”. 4. Sistem menampilkan konfirmasi penghapusan. 5. Pengguna mengonfirmasi penghapusan. 6. Sistem menghapus prodi dan menampilkan notifikasi bahwa prodi telah dihapus..
Kondisi Akhir (<i>Postconditions</i>): 1. Jika Pengguna menambahkan prodi, prodi baru tersimpan dan ditampilkan di daftar prodi. 2. Jika Pengguna menghapus prodi, prodi tersebut akan dihapus dari sistem. 3. Sistem menampilkan notifikasi sukses setiap kali tindakan berhasil dilakukan.

10. Mengelola Fakultas (UC-010)

Tabel 3.16 Deskripsi *Use Case* Mengelola Fakultas

Nama <i>Use Case</i>: Mengelola Fakultas	ID: UC-010
Aktor: Admin, Admin Universitas	
Deskripsi: <i>Use case</i> ini menggambarkan proses pengelolaan fakultas yang dilakukan oleh pengguna (Admin dan Admin Universitas), termasuk menambahkan, melihat, dan menghapus fakultas yang ditampilkan pada sistem.	
Pemicu (<i>Trigger</i>): Pengguna memilih menu "Fakultas" di <i>sidebar</i> .	
Kondisi Awal (<i>Preconditions</i>): Pengguna telah masuk ke dalam sistem dengan otoritas Admin atau Admin Universitas.	
Alur Normal (Melihat Fakultas): 1. Pengguna memilih menu "Fakultas" dan memilih submenu "Daftar Fakultas". 2. Sistem menampilkan daftar fakultas yang ada.	
Alur Normal (Menambah Fakultas) 1. Pengguna memilih menu "Fakultas" dan memilih submenu "Tambah Fakultas". 2. Sistem menampilkan <i>form</i> tambah fakultas. 3. Pengguna mengisi <i>form</i> fakultas dan mengklik Simpan. 4. Sistem menampilkan notifikasi bahwa fakultas berhasil ditambahkan.	
Alur Alternatif 1 (Menghapus Fakultas) 7. Pengguna mengakses <i>sidebar</i> “Prodi” dan memilih submenu “Daftar Prodi”. 8. Sistem menampilkan daftar prodi yang tersedia. 9. Pengguna memilih prodi yang ingin dihapus dengan mengklik tombol “Delete”. 10. Sistem menampilkan konfirmasi penghapusan. 11. Pengguna mengonfirmasi penghapusan.	

12. Sistem menghapus prodi dan menampilkan notifikasi bahwa prodi telah dihapus..
Kondisi Akhir (<i>Postconditions</i>): 1. Jika Pengguna menambahkan fakultas, fakultas baru tersimpan dan ditampilkan di daftar fakultas. 2. Jika Pengguna menghapus fakultas, fakultas tersebut akan dihapus dari sistem. 3. Sistem menampilkan notifikasi sukses setiap kali tindakan berhasil dilakukan.

11. Mengelola Universitas (UC-011)

Tabel 3.17 Deskripsi *Use Case* Mengelola Universitas

Nama <i>Use Case</i>: Mengelola Universitas	ID: UC-011
Aktor: Admin	
Deskripsi: <i>Use case</i> ini menggambarkan proses pengelolaan universitas yang dilakukan oleh pengguna (Admin), termasuk menambahkan, melihat, dan menghapus universitas yang ditampilkan pada sistem.	
Pemicu (<i>Trigger</i>): Pengguna memilih menu "Universitas" di <i>sidebar</i> .	
Kondisi Awal (<i>Preconditions</i>): Pengguna telah masuk ke dalam sistem dengan otoritas Admin.	
Alur Normal (Melihat Universitas): 1. Pengguna memilih menu "Universitas" dan memilih submenu "Daftar Universitas". 2. Sistem menampilkan daftar universitas yang ada.	
Alur Normal (Menambah Universitas) 1. Pengguna memilih menu "Universitas" dan memilih submenu "Tambah Universitas. 2. Sistem menampilkan <i>form</i> tambah universitas. 3. Pengguna mengisi <i>form</i> universitas dan mengklik Simpan. 4. Sistem menampilkan notifikasi bahwa universitas berhasil ditambahkan.	
Alur Alternatif 1 (Menghapus Universitas) 1. Pengguna mengakses <i>sidebar</i> "Universitas" dan memilih submenu "Daftar Universitas". 2. Sistem menampilkan daftar universitas yang tersedia. 3. Pengguna memilih universitas yang ingin dihapus dengan mengklik tombol "Delete". 4. Sistem menampilkan konfirmasi penghapusan. 5. Pengguna mengonfirmasi penghapusan. 6. Sistem menghapus universitas dan menampilkan notifikasi bahwa universitas telah dihapus..	

Kondisi Akhir (Postconditions):

1. Jika Pengguna menambahkan universitas, universitas baru tersimpan dan ditampilkan di daftar universitas.
2. Jika Pengguna menghapus universitas, universitas tersebut akan dihapus dari sistem.
3. Sistem menampilkan notifikasi sukses setiap kali tindakan berhasil dilakukan.

12. Melihat Mahasiswa (UC-012)

Tabel 3.18 Deskripsi Use Case Melihat Mahasiswa

Nama Use Case: Melihat Mahasiswa	ID: UC-012
Aktor: Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi.	
Deskripsi: <i>Use case</i> ini menggambarkan proses pengelolaan mahasiswa yang dilakukan oleh pengguna untuk melihat mahasiswa yang ditampilkan pada sistem.	
Pemicu (Trigger): Pengguna memilih menu "Mahasiswa" di <i>sidebar</i> .	
Kondisi Awal (Preconditions): Pengguna telah masuk ke dalam sistem dengan otoritas Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi.	
Alur Normal (Normal Course): 1. Pengguna memilih menu "Mahasiswa" dan memilih submenu "Daftar Mahasiswa". 2. Sistem menampilkan daftar mahasiswa yang ada.	
Kondisi Akhir (Postconditions): 1. Sistem menampilkan data Mahasiswa.	

13. Mengelola Mahasiswa (UC-013)

Tabel 3.19 Deskripsi Use Case Mengelola Mahasiswa

Nama Use Case: Mengelola Mahasiswa	ID: UC-013
Aktor: Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi.	
Deskripsi: <i>Use case</i> ini menggambarkan proses pengelolaan mahasiswa yang dilakukan oleh pengguna, termasuk menambahkan, mengubah, dan menghapus mahasiswa yang ditampilkan pada sistem.	
Pemicu (Trigger): Pengguna memilih menu "Mahasiswa" di <i>sidebar</i> .	

Kondisi Awal (<i>Preconditions</i>): Pengguna telah masuk ke dalam sistem dengan otoritas Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi.
Alur Normal (Menambah Mahasiswa) 1. Pengguna memilih menu "Mahasiswa" dan memilih submenu "Tambah Mahasiswa. 2. Sistem menampilkan <i>form</i> tambah mahasiswa. 3. Pengguna mengisi <i>form</i> mahasiswa dan mengklik Simpan. 4. Sistem menampilkan notifikasi bahwa mahasiswa berhasil ditambahkan.
Alur Alternatif 1 (Mengubah Mahasiswa) 1. Pengguna mengakses <i>sidebar</i> "Mahasiswa" dan memilih submenu "Daftar Mahasiswa". 2. Sistem menampilkan daftar mahasiswa yang ada. 3. Pengguna memilih mahasiswa yang ingin diperbarui dengan mengklik tombol "Edit". 4. Sistem menampilkan <i>form</i> edit mahasiswa dengan data yang sudah ada. 5. Pengguna mengubah data mahasiswa dan mengklik Simpan. 6. Sistem mengubah mahasiswa dan menampilkan notifikasi bahwa <i>user</i> berhasil diperbarui.
Alur Alternatif 2 (Menghapus Mahasiswa) 1. Pengguna mengakses <i>sidebar</i> "Mahasiswa" dan memilih submenu "Daftar Mahasiswa". 2. Sistem menampilkan daftar mahasiswa yang tersedia. 3. Pengguna memilih mahasiswa yang ingin dihapus dengan mengklik tombol "Delete". 4. Sistem menampilkan konfirmasi penghapusan. 5. Pengguna mengonfirmasi penghapusan. 6. Sistem menghapus mahasiswa dan menampilkan notifikasi bahwa mahasiswa telah dihapus.
Kondisi Akhir (<i>Postconditions</i>): 1. Jika Pengguna menambahkan mahasiswa, mahasiswa baru tersimpan dan ditampilkan di daftar mahasiswa. 2. Jika Pengguna mengubah mahasiswa, mahasiswa tersebut kan diperbarui dari sistem. 3. Jika Pengguna menghapus mahasiswa, mahasiswa tersebut akan dihapus dari sistem. 4. Sistem menampilkan notifikasi sukses setiap kali tindakan berhasil dilakukan.

14. Melihat Visualisasi CPL per Mahasiswa (UC-014)

Tabel 3.20 Deskripsi *Use Case* Melihat Visualisasi CPL per Mahasiswa

Nama Use Case: Melihat Visualisasi CPL per Mahasiswa	ID: UC-014
---	-------------------

Aktor: Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi.
Deskripsi: <i>Use case</i> ini menggambarkan melihat Visualisasi CPL per Mahasiswa.
Pemicu (Trigger): Pengguna memilih menu “Visualisasi” lalu memilih submenu “Per Mahasiswa”.
Kondisi Awal (Preconditions): Pengguna telah masuk ke dalam sistem dengan otoritas Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi.
Alur Normal (Normal Course): 1. Pengguna memilih menu “Visualisasi” dan memilih submenu “Per Mahasiswa”. 2. Sistem menampilkan <i>form</i> Visualisasi CPL per Mahasiswa. 3. Pengguna mengisi <i>form</i> Visualisasi CPL per Mahasiswa sesuai dengan data yang ingin ditampilkan. 4. Sistem menampilkan tombol Print PDF, visualisasi CPL per Mahasiswa, serta <i>form</i> untuk melihat CPMK mahasiswa.
Kondisi Akhir (Postconditions): 1. Sistem menampilkan tombol Print PDF. 2. Sistem menampilkan visualisasi CPL per mahasiswa.

15. Mencetak Laporan Visualisasi CPL per Mahasiswa (UC-015)

Tabel 3.21 Deskripsi *Use Case* Mencetak Laporan Visualisasi CPL per Mahasiswa

Nama Use Case: Mencetak Laporan Visualisasi CPL per Mahasiswa	ID: UC-015
Aktor: Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi, Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen.	
Deskripsi: <i>Use case</i> ini menggambarkan proses pencetakan laporan visualisasi mahasiswa yang dipilih. Laporan ini berbentuk <i>file</i> PDF yang dapat dicetak oleh pengguna dengan otoritas yang telah ditentukan.	
Pemicu (Trigger): Pengguna memilih menu “Visualisasi” lalu memilih submenu “Per Mahasiswa”. Dan telah mengisi <i>form</i> visualisasi CPL per Mahasiswa	
Kondisi Awal (Preconditions): Pengguna telah masuk ke dalam sistem dengan otoritas Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi, Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen.	

Alur Normal (<i>Normal Course</i>): 1. Pengguna memilih menu “Visualisasi” dan memilih submenu “Per Mahasiswa”. 2. Sistem menampilkan <i>form</i> Visualisasi CPL per Mahasiswa. 3. Pengguna mengisi <i>form</i> Visualisasi CPL per Mahasiswa sesuai dengan data yang ingin ditampilkan. 4. Sistem menampilkan tombol Print PDF, visualisasi CPL per Mahasiswa, serta <i>form</i> untuk melihat CPMK mahasiswa. 5. Pengguna memilih opsi "Print PDF." 6. Sistem menghasilkan laporan dalam format PDF yang siap dicetak.
Kondisi Akhir (<i>Postconditions</i>): 1. Pengguna mendapatkan laporan cetak Visualisasi CPL per Mahasiswa dalam format PDF.

16. Melihat Visualisasi CPMK per Mahasiswa (UC-016)

Tabel 3.22 Deskripsi *Use Case* Melihat Visualisasi CPMK per Mahasiswa

Nama <i>Use Case</i>: Melihat Visualisasi CPMK per Mahasiswa	ID: UC-016
Aktor: Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi, Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen.	
Deskripsi: <i>Use case</i> ini menggambarkan melihat Visualisasi CPMK per Mahasiswa. <i>Use Case</i> ini dapat berjalan ketika <i>Use Case</i> melihat visualisasi CPL per Mahasiswa dilakukan/ <i>Extends</i> .	
Pemicu (<i>Trigger</i>): Pengguna mengisi form lihat CPMK pada halaman Visualisasi CPL per Mahasiswa.	
Kondisi Awal (<i>Preconditions</i>): Pengguna telah masuk ke dalam sistem dengan otoritas Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi, Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen.	
Alur Normal (<i>Normal Course</i>): 1. Pengguna memilih menu “Visualisasi” dan memilih submenu “Per Mahasiswa”. 2. Sistem menampilkan <i>form</i> Visualisasi CPL per Mahasiswa. 3. Pengguna mengisi <i>form</i> Visualisasi CPL per Mahasiswa sesuai dengan data yang ingin ditampilkan dan mengklik tombol Submit. 4. Sistem menampilkan tombol Print PDF, visualisasi CPL per Mahasiswa, serta <i>form</i> untuk melihat CPMK mahasiswa. 5. Pengguna mengisi <i>form</i> CPMK Mahasiswa dan mengklik tombol <i>Submit</i>. 6. Sistem menampilkan tombol Print PDF dan visualisasi CPMK per Mahasiswa.	
Kondisi Akhir (<i>Postconditions</i>):	

1. Sistem menampilkan tombol Print PDF.
2. Sistem menampilkan visualisasi CPMK per mahasiswa.

17. Mencetak Laporan Visualisasi CPMK per Mahasiswa (UC-017)

Tabel 3.23 Deskripsi *Use Case* Mencetak Laporan Visualisasi CPMK per Mahasiswa

Nama <i>Use Case</i>: Mencetak Laporan Visualisasi CPMK per Mahasiswa	ID: UC-017
Aktor: Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi, Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen.	
Deskripsi: <i>Use case</i> ini menggambarkan proses pencetakan laporan visualisasi CPMK mahasiswa yang dipilih. <i>Use Case</i> ini dapat berjalan ketika <i>Use Case</i> melihat visualisasi CPMK per Mahasiswa dilakukan/ <i>Extends</i> . Laporan ini berbentuk <i>file</i> PDF yang dapat dicetak oleh pengguna dengan otoritas yang telah ditentukan.	
Pemicu (<i>Trigger</i>): Pengguna mengisi <i>form</i> lihat CPMK pada halaman Visualisasi CPL per Mahasiswa.	
Kondisi Awal (<i>Preconditions</i>): Pengguna telah masuk ke dalam sistem dengan otoritas Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi, Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, dan sudah mengisi <i>form</i> lihat CPMK	
Alur Normal (<i>Normal Course</i>): 1. Pengguna memilih menu “Visualisasi” dan memilih submenu “Per Mahasiswa”. 2. Sistem menampilkan <i>form</i> Visualisasi CPL per Mahasiswa. 3. Pengguna mengisi <i>form</i> Visualisasi CPL per Mahasiswa sesuai dengan data yang ingin ditampilkan dan mengklik tombol Submit. 4. Sistem menampilkan tombol Print PDF, visualisasi CPL per Mahasiswa, serta <i>form</i> untuk melihat CPMK mahasiswa. 5. Pengguna mengisi <i>form</i> CPMK Mahasiswa dan mengklik tombol Submit. 6. Sistem menampilkan tombol Print PDF dan visualisasi CPMK per Mahasiswa. 7. Pengguna memilih opsi "Print PDF." 8. Sistem menghasilkan laporan dalam format PDF yang siap dicetak.	
Kondisi Akhir (<i>Postconditions</i>): 1. Pengguna mendapatkan laporan cetak Visualisasi CPMK per Mahasiswa dalam format PDF.	

18. Melihat Visualisasi CPL per Angkatan (UC-018)

Tabel 3.24 Deskripsi *Use Case* Melihat Visualisasi CPL per Angkatan

Nama Use Case: Melihat Visualisasi CPL per Angkatan	ID: UC-018
Aktor: Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi, Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen.	
Deskripsi: <i>Use case</i> ini menggambarkan melihat Visualisasi CPL per Angkatan.	
Pemicu (Trigger): Pengguna memilih menu “Visualisasi” lalu memilih submenu “Per Angkatan”.	
Kondisi Awal (Preconditions): Pengguna telah masuk ke dalam sistem dengan otoritas Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, atau Penjamin Mutu	
Alur Normal (Normal Course): 1. Pengguna memilih menu “Visualisasi” dan memilih submenu “Per Angkatan”. 2. Sistem menampilkan <i>form</i> Visualisasi CPL per Angkatan. 3. Pengguna mengisi <i>form</i> Visualisasi CPL per Angkatan sesuai dengan data yang ingin ditampilkan. 4. Sistem menampilkan tombol Print PDF, visualisasi CPL per Angkatan, serta <i>form</i> untuk melihat CPMK Angkatan.	
Kondisi Akhir (Postconditions): 1. Sistem menampilkan tombol Print PDF. 2. Sistem menampilkan visualisasi CPL per Angkatan.	

19. Mencetak Laporan Visualisasi CPL per Angkatan (UC-019)

Tabel 3.25 Deskripsi *Use Case* Mencetak Laporan Visualisasi CPL per Angkatan

Nama Use Case: Mencetak Laporan Visualisasi CPL per Angkatan	ID: UC-019
Aktor: Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi.	
Deskripsi: <i>Use case</i> ini menggambarkan proses pencetakan laporan visualisasi angkatan yang dipilih. Laporan ini berbentuk <i>file</i> PDF yang dapat dicetak oleh pengguna dengan otoritas yang telah ditentukan.	
Pemicu (Trigger): Pengguna memilih menu “Visualisasi” lalu memilih submenu “Per Angkatan”. Dan telah mengisi <i>form</i> visualisasi CPL per Angkatan	

Kondisi Awal (<i>Preconditions</i>): Pengguna telah masuk ke dalam sistem dengan otoritas Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi..
Alur Normal (<i>Normal Course</i>): 1. Pengguna memilih menu “Visualisasi” dan memilih submenu “Per Angkatan”. 2. Sistem menampilkan <i>form</i> Visualisasi CPL per Angkatan. 3. Pengguna mengisi <i>form</i> Visualisasi CPL per Angkatan sesuai dengan data yang ingin ditampilkan. 4. Sistem menampilkan tombol Print PDF, visualisasi CPL per Angkatan, serta <i>form</i> untuk melihat CPMK Angkatan. 5. Pengguna memilih opsi "Print PDF." 6. Sistem menghasilkan laporan dalam format PDF yang siap dicetak.
Kondisi Akhir (<i>Postconditions</i>): 1. Pengguna mendapatkan laporan cetak Visualisasi CPL per Angkatan dalam format PDF.

20. Melihat Visualisasi CPMK per Angkatan (UC-020)

Tabel 3.26 Deskripsi *Use Case* Melihat Visualisasi CPMK per Angkatan

Nama <i>Use Case</i>: Melihat Visualisasi CPMK per Angkatan	ID: UC-020
Aktor: Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi.	
Deskripsi: <i>Use case</i> ini menggambarkan melihat Visualisasi CPMK per Angkatan. <i>Use Case</i> ini dapat berjalan ketika <i>Use Case</i> melihat visualisasi CPL per Angkatan dilakukan/ <i>Extends</i> .	
Pemicu (<i>Trigger</i>): Pengguna mengisi <i>form</i> lihat CPMK pada halaman Visualisasi CPL per Angkatan.	
Kondisi Awal (<i>Preconditions</i>): Pengguna telah masuk ke dalam sistem dengan otoritas Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi.	
Alur Normal (<i>Normal Course</i>): 1. Pengguna memilih menu “Visualisasi” dan memilih submenu “Per Angkatan”. 2. Sistem menampilkan <i>form</i> Visualisasi CPL per Angkatan. 3. Pengguna mengisi <i>form</i> Visualisasi CPL per Angkatan sesuai dengan data yang ingin ditampilkan dan mengklik tombol Submit. 4. Sistem menampilkan tombol Print PDF, visualisasi CPL per Angkatan, serta <i>form</i> untuk melihat CPMK Angkatan.	

5. Pengguna mengisi <i>form</i> CPMK Angkatan dan mengklik tombol <i>Submit</i> .
6. Sistem menampilkan tombol Print PDF dan visualisasi CPMK per Angkatan.
Kondisi Akhir (<i>Postconditions</i>):
1. Sistem menampilkan tombol Print PDF.
2. Sistem menampilkan visualisasi CPMK per Angkatan.

21. Mencetak Laporan Visualisasi CPMK per Angkatan (UC-021)

Tabel 3.27 Deskripsi *Use Case* Mencetak Laporan Visualisasi CPMK per Angkatan

Nama <i>Use Case</i>: Mencetak Laporan Visualisasi CPMK per Angkatan	ID: UC-021
Aktor: Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi.	
Deskripsi: <i>Use case</i> ini menggambarkan proses pencetakan laporan visualisasi CPMK angkatan yang dipilih. <i>Use Case</i> ini dapat berjalan ketika <i>Use Case</i> melihat visualisasi CPMK per Angkatan dilakukan/ <i>Extends</i> . Laporan ini berbentuk <i>file</i> PDF yang dapat dicetak oleh pengguna dengan otoritas yang telah ditentukan.	
Pemicu (<i>Trigger</i>): Pengguna mengisi <i>form</i> lihat CPMK pada halaman Visualisasi CPL per Angkatan.	
Kondisi Awal (<i>Preconditions</i>): Pengguna telah masuk ke dalam sistem dengan otoritas Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi, dan sudah mengisi <i>form</i> lihat CPMK.	
Alur Normal (<i>Normal Course</i>): 1. Pengguna memilih menu “Visualisasi” dan memilih submenu “Per Angkatan”. 2. Sistem menampilkan <i>form</i> Visualisasi CPL per Angkatan. 3. Pengguna mengisi <i>form</i> Visualisasi CPL per Angkatan sesuai dengan data yang ingin ditampilkan dan mengklik tombol <i>Submit</i>. 4. Sistem menampilkan tombol Print PDF, visualisasi CPL per Angkatan, serta <i>form</i> untuk melihat CPMK Angkatan. 5. Pengguna mengisi <i>form</i> CPMK Angkatan dan mengklik tombol <i>Submit</i>. 6. Sistem menampilkan tombol Print PDF dan visualisasi CPMK per Angkatan. 7. Pengguna memilih opsi "Print PDF." 8. Sistem menghasilkan laporan dalam format PDF yang siap dicetak.	
Kondisi Akhir (<i>Postconditions</i>):	

1. Pengguna mendapatkan laporan cetak Visualisasi CPMK per Mahasiswa dalam format PDF.

22. Melihat Visualisasi CPMK per Mata Kuliah (UC-022)

Tabel 3.28 Deskripsi *Use Case* Melihat Visualisasi CPMK per Mata Kuliah

Nama Use Case: Melihat Visualisasi CPMK per Mata Kuliah	ID: UC-022
Aktor: Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi, Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen.	
Deskripsi: <i>Use case</i> ini menggambarkan melihat Visualisasi CPMK per Mata Kuliah.	
Pemicu (Trigger): Pengguna memilih menu “Visualisasi” lalu memilih submenu “Per Mata Kuliah”.	
Kondisi Awal (Preconditions): Pengguna telah masuk ke dalam sistem dengan otoritas Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, atau Penjamin Mutu.	
Alur Normal (Normal Course): 1. Pengguna memilih menu “Visualisasi” dan memilih submenu “Per Mata Kuliah”. 2. Sistem menampilkan <i>form</i> Visualisasi CPMK per Mata Kuliah. 3. Pengguna mengisi <i>form</i> Visualisasi CPMK per Mata Kuliah sesuai dengan data yang ingin ditampilkan. 4. Sistem menampilkan tombol Print PDF, visualisasi CPMK per Mata Kuliah, serta <i>form</i> untuk melihat CPMK Mahasiswa.	
Kondisi Akhir (Postconditions): 1. Sistem menampilkan tombol Print PDF. 2. Sistem menampilkan visualisasi CPMK per Mata Kuliah.	

23. Mencetak Laporan Visualisasi CPMK per Mata Kuliah (UC-023)

Tabel 3.29 Deskripsi *Use Case* Mencetak Laporan Visualisasi CPMK per Mata Kuliah

Nama Use Case: Mencetak Laporan Visualisasi CPMK per Mata Kuliah	ID: UC-023
Aktor: Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi.	
Deskripsi: <i>Use case</i> ini menggambarkan proses pencetakan laporan visualisasi mata kuliah yang dipilih. Laporan ini berbentuk <i>file</i> PDF yang dapat dicetak oleh pengguna dengan otoritas yang telah ditentukan.	

Pemicu (Trigger): Pengguna memilih menu “Visualisasi” lalu memilih submenu “Per Mata Kuliah”. Dan telah mengisi form visualisasi CPMK per Mata Kuliah.
Kondisi Awal (Preconditions): Pengguna telah masuk ke dalam sistem dengan otoritas Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi.
Alur Normal (Normal Course): 1. Pengguna memilih menu “Visualisasi” dan memilih submenu “Per Angkatan”. 2. Sistem menampilkan form Visualisasi CPL per Angkatan. 3. Pengguna mengisi form Visualisasi CPL per Angkatan sesuai dengan data yang ingin ditampilkan. 4. Sistem menampilkan tombol Print PDF, visualisasi CPL per Angkatan, serta <i>form</i> untuk melihat CPMK Angkatan. 5. Pengguna memilih opsi "Print PDF." 6. Sistem menghasilkan laporan dalam format PDF yang siap dicetak.
Kondisi Akhir (Postconditions): 1. Pengguna mendapatkan laporan cetak Visualisasi CPMK per Mata Kuliah dalam format PDF.

24. Melihat Visualisasi CPMK per Mahasiswa (UC-024)

Tabel 3.30 Deskripsi *Use Case* Melihat Visualisasi CPMK per Mahasiswa

Nama Use Case: Melihat Visualisasi CPMK per Mahasiswa	ID: UC-024
Aktor: Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi.	
Deskripsi: <i>Use case</i> ini menggambarkan melihat Visualisasi CPMK per Mahasiswa. <i>Use Case</i> ini dapat berjalan ketika <i>Use Case</i> melihat visualisasi CPMK per Mata Kuliah dilakukan/ <i>Extends</i> .	
Pemicu (Trigger): Pengguna mengisi form lihat CPMK pada halaman Visualisasi CPMK per Mata Kuliah.	
Kondisi Awal (Preconditions): Pengguna telah masuk ke dalam sistem dengan otoritas Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi.	
Alur Normal (Normal Course): 1. Pengguna memilih menu “Visualisasi” dan memilih submenu “Per Mata Kuliah”. 2. Sistem menampilkan <i>form</i> Visualisasi CPMK per Mata Kuliah.	

3. Pengguna mengisi <i>form</i> Visualisasi CPMK per Mata Kuliah sesuai dengan data yang ingin ditampilkan dan mengklik tombol <i>Submit</i>. 4. Sistem menampilkan tombol Print PDF, visualisasi CPMK per Mata Kuliah, serta <i>form</i> untuk melihat CPMK mahasiswa. 5. Pengguna mengisi <i>form</i> CPMK Mahasiswa dan mengklik tombol <i>Submit</i>. 6. Sistem menampilkan tombol Print PDF dan visualisasi CPMK per Mahasiswa.
Kondisi Akhir (<i>Postconditions</i>): 1. Sistem menampilkan tombol Print PDF. 2. Sistem menampilkan visualisasi CPMK per mahasiswa.

25. Mencetak Laporan Visualisasi CPMK per Mahasiswa (UC-025)

Tabel 3.31 Deskripsi *Use Case* Mencetak Laporan Visualisasi CPMK per Mahasiswa

Nama <i>Use Case</i>: Mencetak Laporan Visualisasi CPMK per Mahasiswa	ID: UC-025
Aktor: Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi.	
Deskripsi: <i>Use case</i> ini menggambarkan proses pencetakan laporan visualisasi CPMK mahasiswa yang dipilih. <i>Use Case</i> ini dapat berjalan ketika <i>Use Case</i> melihat visualisasi CPMK per Mata Kuliah dilakukan/ <i>Extends</i> . Laporan ini berbentuk <i>file</i> PDF yang dapat dicetak oleh pengguna dengan otoritas yang telah ditentukan.	
Pemicu (<i>Trigger</i>): Pengguna mengisi <i>form</i> lihat CPMK pada halaman Visualisasi CPMK per Mata Kuliah.	
Kondisi Awal (<i>Preconditions</i>): Pengguna telah masuk ke dalam sistem dengan otoritas Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi, dan sudah mengisi <i>form</i> lihat CPMK	
Alur Normal (<i>Normal Course</i>): 1. Pengguna memilih menu “Visualisasi” dan memilih submenu “Per Mata Kuliah”. 2. Sistem menampilkan <i>form</i> Visualisasi CPMK per Mata Kuliah. 3. Pengguna mengisi <i>form</i> Visualisasi CPMK per Mata Kuliah sesuai dengan data yang ingin ditampilkan dan mengklik tombol <i>Submit</i>. 4. Sistem menampilkan tombol Print PDF, visualisasi CPMK per Mata Kuliah, serta <i>form</i> untuk melihat CPMK mahasiswa. 5. Pengguna mengisi <i>form</i> CPMK Mahasiswa dan mengklik tombol <i>Submit</i>. 6. Sistem menampilkan tombol Print PDF dan visualisasi CPMK per Mahasiswa. 7. Pengguna memilih opsi "Print PDF."	

8. Sistem menghasilkan laporan dalam format PDF yang siap dicetak.
Kondisi Akhir (<i>Postconditions</i>):
1. Pengguna mendapatkan laporan cetak Visualisasi CPMK per Mahasiswa dalam format PDF.

3.5.9 Spesifikasi Perangkat Lunak

Spesifikasi perangkat keras dan perangkat lunak merupakan sebuah dokumentasi yang menggambarkan spesifikasi dari sebuah perangkat yang dibutuhkan untuk memanjang jalannya aplikasi. Berikut spesifikasi perangkat keras dan perangkat lunak yang dapat dilihat pada Tabel 3.32.

Tabel 3.32 Spesifikasi Perangkat Keras dan Perangkat Lunak

No	Spesifikasi	Server	Client
1	Sistem Operasi	Linux (Kernel Version 4.18.0-553.27.1.lve.el8.x86_64)	Windows 10/11 / macOS / Linux
2	Perangkat Lunak	- Database Server (MySQL 8.0.41) - Framework (Laravel 8.83) - Hosting (cPanel 126.0)	Web browser (Google Chrome, Mozilla Firefox, Safari, Microsoft Edge, dll)

3.5.10 Skenario Pengujian Fungsional

Penelitian ini akan menerapkan pengujian sistem menggunakan metode *black-box*. Tabel berikut menyajikan skenario pengujian sistem, dimulai dari skenario normal yang tertera pada Tabel 3.33.

Tabel 3.33 Skenario Pengujian Sistem

No	Testing ID	Objective/ Output	Kasus Uji	Tipe Uji	Role	Hasil yang Diharapkan	Hasil Sebenarnya	Status Pengujian
1	BBT-001/UC-001/FR-001	Sistem harus menampilkan daftar berita yang dapat diakses oleh tamu.	Mengakses halaman utama dan melihat daftar berita	Positif	Tamu	Sistem menampilkan halaman utama (<i>landing page</i>) dengan daftar berita yang berstatus <i>published</i> . Setiap berita menampilkan judul, gambar, dan tanggal publikasi.		
2	BBT-002/UC-	Sistem harus	mengisi <i>form</i>	Positif	Tamu	Sistem menerima data input dari form		

No	Testing ID	Objective/ Output	Kasus Uji	Tipe Uji	Role	Hasil yang Diharapkan	Hasil Sebenarnya	Status Pengujian
	002/FR-002	menyediakan <i>form</i> registrasi universitas untuk tamu.	registrasi dan menekan tombol <i>submit</i>			registrasi, memvalidasi, dan menyimpan data. Sistem menampilkan notifikasi bahwa registrasi berhasil dikirim dan menunggu verifikasi.		
3a	BBT-003a/U C-003/FR-003	Sistem harus menyediakan <i>form login</i> dan autentikasi.	<i>User</i> memasukkan email dan <i>password</i> lalu berhasil diarahkan ke <i>dashboard</i> sesuai perannya	Positif	Admin, Admin Universitas, Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi	Pengguna melakukan login melalui dua skenario: (1) memasukkan email dan <i>password</i> yang valid, dan (2) menggunakan opsi 'Masuk dengan Google' dengan akun yang telah terdaftar di sistem.		
3b	BBT-003b/U C-003/FR-003	Sistem harus menyediakan <i>form login</i> dan autentikasi.	Pengguna mencoba <i>login</i> dengan <i>password</i> yang salah.	Negatif	Admin, Admin Universitas, Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi	Sistem menolak autentikasi dan menampilkan pesan <i>error</i> yang informatif (misal: "Email atau <i>password</i> salah").		

No	Testing ID	Objective/ Output	Kasus Uji	Tipe Uji	Role	Hasil yang Diharapkan	Hasil Sebenarnya	Status Pengujian
3c	BBT-003c/U C-003/FR-003	Sistem harus menyediakan <i>form login</i> dan autentikasi.	Pengguna mencoba <i>login</i> dengan email yang tidak terdaftar di sistem.	Negatif	Admin, Admin Universitas, Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi	Sistem menolak autentikasi dan menampilkan pesan <i>error</i> yang sama seperti <i>password</i> salah untuk alasan keamanan.		
4a	BBT-004a/U C-004/FR-004	User dapat menampilkan seluruh daftar berita	User menampilkan seluruh daftar berita	Positif	Admin	Sistem menampilkan tabel berisi daftar semua berita (baik <i>published</i> maupun <i>draft</i>) yang ada di <i>database</i> , beserta informasi judul, status, tanggal, dan tombol aksi (<i>Edit</i> , <i>Delete</i>).		
4b	BBT-004b/U C-004/FR-004	User dapat menambahkan berita	User menambahkan berita	Positif	Admin	Sistem menerima data berita baru, memvalidasi, dan menyimpan ke <i>database</i> . Berita baru muncul di daftar berita. Sistem menampilkan notifikasi sukses.		
4c	BBT-004c/U C-004/FR-004	User dapat mengubah berita	User mengubah berita	Positif	Admin	Sistem menampilkan data berita yang dipilih pada <i>form</i> edit. Setelah diubah dan disimpan, data berita di <i>database</i> terbaru.		

No	Testing ID	Objective/ Output	Kasus Uji	Tipe Uji	Role	Hasil yang Diharapkan	Hasil Sebenarnya	Status Pengujian
						Sistem menampilkan notifikasi sukses.		
4d	BBT-004d/U C-004/FR-004	User dapat menghapus berita	User menghapus berita	Positif	Admin	Setelah konfirmasi, sistem menghapus data berita dari database. Berita tidak lagi muncul di daftar. Sistem menampilkan notifikasi sukses.		
5a	BBT-005a/U C-005/FR-005	User dapat melihat daftar registrasi universitas	User melihat daftar registrasi universitas	Positif	Admin	Sistem menampilkan daftar semua universitas yang melakukan registrasi beserta statusnya (pending, approved, rejected).		
5b	BBT-005b/U C-005/FR-005	User dapat Menyetujui registrasi universitas	User Menyetujui registrasi universitas	Positif	Admin	Status registrasi universitas berubah menjadi "approved". Akun Admin Universitas untuk universitas tersebut dibuat. Notifikasi email dikirim. Sistem menampilkan pesan sukses.		
5c	BBT-005c/U C-005/FR-005	User dapat Menolak registrasi universitas	User Menolak registrasi universitas	Positif	Admin	Status registrasi universitas berubah menjadi "rejected". Alasan penolakan tersimpan. Notifikasi email dikirim. Sistem menampilkan pesan sukses.		
6a	BBT-006a/U C-006/FR-006	User dapat menampilkan daftar user	User melihat daftar user yang telah terdaftar	Positif	Admin, Admin Universitas	Sistem menampilkan daftar semua pengguna yang terdaftar beserta informasi relevan seperti nama, email, otoritas, program studi, dan fakultas. Tombol aksi (<i>Edit</i> ,		

No	Testing ID	Objective/ Output	Kasus Uji	Tipe Uji	Role	Hasil yang Diharapkan	Hasil Sebenarnya	Status Pengujian
						<i>Delete, Reset Password</i>) tersedia.		
6b	BBT-006b/UC-006/FR-006	User dapat menambahkan user	User menambahkan data user	Positif	Admin, Admin Universitas	Sistem menerima data user baru, memvalidasi, dan menyimpan ke database. User baru muncul di daftar. Sistem menampilkan notifikasi sukses.		
6c	BBT-006c/UC-006/FR-006	User dapat mengubah user	Admin mengubah data user	Positif	Admin, Admin Universitas	Sistem menampilkan data user yang dipilih pada form edit. Setelah diubah (termasuk otoritas) dan disimpan, data user di database terbaru. Sistem menampilkan notifikasi sukses.		
6d	BBT-006d/UC-006/FR-006	User dapat menghapus user	User menghapus user	Positif	Admin, Admin Universitas	Setelah konfirmasi, sistem menghapus data user dari database. User tidak lagi muncul di daftar. Sistem menampilkan notifikasi sukses.		
7	BBT-007/UC-007/FR-007	User dapat mengatur tampilan antarmuka sistem	User melakukan pengaturan UI dan perubahan tampilan berhasil disimpan	Positif	Admin Universitas	Sistem menyimpan pengaturan kustomisasi tema (warna header/footer, logo universitas). Tampilan sistem (frontend) berubah sesuai dengan pengaturan baru setelah disimpan.		
8	BBT-008/UC-008/FR-008	User dapat mengelola profil akun	User mengubah data profil dan	Positif	Admin, Admin Universitas, Wakil Rektor, Wakil Dekan,	Sistem berhasil menyimpan perubahan data profil pengguna (nama, password baru yang ter-hash, otoritas aktif		

No	Testing ID	Objective/ Output	Kasus Uji	Tipe Uji	Role	Hasil yang Diharapkan	Hasil Sebenarnya	Status Pengujian
			menyimpannya		Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi	jika ada >1). Notifikasi sukses ditampilkan untuk setiap perubahan.		
9a	BBT-009/UC-009/FR-009	User dapat menampilkan daftar prodi	Melihat daftar program studi	Positif	Admin, Admin Universitas	Sistem menampilkan tabel berisi daftar semua program studi yang terdaftar, beserta fakultas dan universitas terkait, dan tombol aksi (Delete).		
9b	BBT-009b/UC-009/FR-009	User dapat menambahkan data prodi	Menambahkan program studi	Positif	Admin, Admin Universitas	Sistem menerima data prodi baru, memvalidasi (termasuk keterkaitan dengan fakultas dan universitas), dan menyimpan ke database. Prodi baru muncul di daftar. Notifikasi sukses.		
9c	BBT-009c/UC-009/FR-009	User dapat menghapus data prodi	Menghapus program studi	Positif	Admin, Admin Universitas	Setelah konfirmasi, sistem menghapus data prodi dari database. Prodi tidak lagi muncul di daftar. Notifikasi sukses.		
10a	BBT-010a/UC-010/FR-010	User dapat menampilkan daftar fakultas	Melihat daftar fakultas	Positif	Admin, Admin Universitas	Sistem menampilkan tabel berisi daftar semua fakultas yang terdaftar, beserta universitas terkait, dan tombol aksi (Delete).		

No	Testing ID	Objective/ Output	Kasus Uji	Tipe Uji	Role	Hasil yang Diharapkan	Hasil Sebenarnya	Status Pengujian
10b	BBT-010b/U C-010/FR-010	User dapat menambahkan data fakultas	Menambahkan fakultas	Positif	Admin, Admin Universitas	Sistem menerima data fakultas baru, memvalidasi (termasuk keterkaitan dengan universitas), dan menyimpan ke database. Fakultas baru muncul di daftar. Notifikasi sukses.		
10c	BBT-010c/U C-010/FR-010	User dapat menghapus data fakultas	Menghapus fakultas	Positif	Admin, Admin Universitas	Setelah konfirmasi, sistem menghapus data fakultas dari database. Fakultas tidak lagi muncul di daftar. Notifikasi sukses.		
11a	BBT-011a/U C-011/FR-011	User dapat menambahkan dan menghapus data universitas	Melihat daftar universitas	Positif	Admin	Sistem menampilkan tabel berisi daftar semua universitas yang terdaftar, beserta logo (jika ada) dan tombol aksi (Delete).		
11b	BBT-011b/U C-011/FR-011	User dapat menambahkan data universitas	Menambahkan universitas	Positif	Admin	Sistem menerima data universitas baru, memvalidasi, mengunggah logo, dan menyimpan ke database. Universitas baru muncul di daftar. Notifikasi sukses.		
11c	BBT-011c/U C-011/FR-011	User dapat menghapus data universitas	Menghapus universitas	Positif	Admin	Setelah konfirmasi, sistem menghapus data universitas dari database. Universitas tidak lagi muncul di daftar. Notifikasi sukses.		
12	BBT-012/UC-012/FR-012	User dapat menampilkan daftar mahasiswa	Melihat data mahasiswa	Positif	Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin	Sistem menampilkan tabel berisi daftar mahasiswa yang relevan dengan hak akses pengguna (misalnya, berdasarkan prodi),		

No	Testing ID	Objective/ Output	Kasus Uji	Tipe Uji	Role	Hasil yang Diharapkan	Hasil Sebenarnya	Status Pengujian
					Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi	beserta NPM, nama, angkatan, dan tombol aksi.		
13a	BBT-013a/UC-013/FR-013	User dapat menambahkan mahasiswa		Positif	Dosen, Penjamin Mutu Program Studi	Sistem menerima data mahasiswa baru (NPM, nama, angkatan), memvalidasi, dan menyimpan ke database. Mahasiswa baru muncul di daftar. Notifikasi sukses.		
13b	BBT-013b/UC-013/FR-013	User dapat mengubah data mahasiswa	Mengedit data mahasiswa	Positif	Dosen, Penjamin Mutu Program Studi	Sistem menampilkan data mahasiswa yang dipilih pada form edit. Setelah diubah dan disimpan, data mahasiswa di database terbaru. Notifikasi sukses.		
13c	BBT-013c/UC-013/FR-013	User dapat menghapus data mahasiswa	Menghapus data mahasiswa	Positif	Dosen, Penjamin Mutu Program Studi	Setelah konfirmasi, sistem menghapus data mahasiswa dari database. Mahasiswa tidak lagi muncul di daftar. Notifikasi sukses.		
14	BBT-014/UC-014/FR-014	User dapat menampilkan visualisasi CPL per mahasiswa.	User memilih mahasiswa dan melihat grafik visualisasi CPL	Positif	Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas,	Sistem menampilkan visualisasi CPL per mahasiswa dalam bentuk grafik (misalnya radar chart), tabel ketercapaian, pemetaan profil lulusan, dan soal terendah CPL.		

No	Testing ID	Objective/ Output	Kasus Uji	Tipe Uji	Role	Hasil yang Diharapkan	Hasil Sebenarnya	Status Pengujian
					Penjamin Mutu Program Studi			
15	BBT-015/UC-015/FR-015	User dapat mencetak laporan visualisasi CPL per mahasiswa dalam PDF.	User menekan tombol cetak pada visualisasi CPL dan hasil dalam PDF sesuai tampilan	Positif	Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi	Sistem menghasilkan <i>file</i> PDF dari visualisasi CPL per Mahasiswa. <i>File</i> PDF berisi data dan grafik yang sesuai dengan tampilan di halaman web, serta memiliki tata letak yang rapi.		
16	BBT-016/UC-016 dan UC-024/FR-016	User dapat menampilkan visualisasi CPMK per mahasiswa dari dua jalur (CPL dan Mata Kuliah)	User membuka visualisasi CPMK dari jalur CPL atau Mata Kuliah dan melihat data mahasiswa	Positif	Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi	Sistem menampilkan visualisasi CPMK per mahasiswa (grafik radar, tabel ketercapaian, soal terendah CPMK) berdasarkan mata kuliah yang relevan, baik diakses dari CPL maupun Mata Kuliah.		
17	BBT-017/UC-017 dan UC-025/FR-025	User dapat mencetak visualisasi CPMK per Mahasiswa dalam PDF dari dua jalur	User mencetak visualisasi CPMK dari dua jalur dan memasti	Positif	Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas,	Sistem menghasilkan <i>file</i> PDF dari visualisasi CPMK per Mahasiswa. <i>File</i> PDF berisi data dan grafik yang sesuai dengan tampilan di halaman		

No	Testing ID	Objective/ Output	Kasus Uji	Tipe Uji	Role	Hasil yang Diharapkan	Hasil Sebenarnya	Status Pengujian
			kan hasil PDF sesuai visualisasi		Penjamin Mutu Fakultas, Penjamin Mutu Program Studi	web, serta memiliki tata letak yang rapi.		
18	BBT-018/UC-018/FR-018	User dapat menampilkan visualisasi CPL per angkatan	User memilih tahun angkatan dan melihat grafik visualisasi CPL	Positif	Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi	Sistem menampilkan visualisasi CPL per angkatan (grafik radar rata-rata, min, maks), tabel soal CPL terendah, dan daftar mahasiswa beserta capaian CPL masing-masing.		
19	BBT-019/UC-019/FR-019	User dapat mencetak visualisasi CPL per angkatan dalam PDF	User mencetak grafik visualisasi CPL per angkatan dan hasil sesuai tampilan	Positif	Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi	Sistem menghasilkan <i>file</i> PDF dari visualisasi CPL per Angkatan. <i>File</i> PDF berisi data dan grafik yang sesuai dengan tampilan di halaman web, serta memiliki tata letak yang rapi.		
20	BBT-020/UC-020/FR-020	User dapat menampilkan visualisasi CPMK per angkatan	User memilih tahun angkatan dan melihat grafik	Positif	Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin	Sistem menampilkan visualisasi CPMK per angkatan (grafik radar rata-rata, min, maks CPMK untuk mata		

No	Testing ID	Objective/ Output	Kasus Uji	Tipe Uji	Role	Hasil yang Diharapkan	Hasil Sebenarnya	Status Pengujian
			visualisasi CPMK		Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi	kuliah tertentu), tabel soal CPMK terendah.		
21	BBT-021/UC-021/FR-021	User dapat mencetak visualisasi CPMK per Angkatan dalam PDF.	User mencetak grafik visualisasi CPMK per Angkatan dan hasil sesuai tampilan	Positif	Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi	Sistem menghasilkan <i>file</i> PDF dari visualisasi CPMK per Angkatan. <i>File</i> PDF berisi data dan grafik yang sesuai dengan tampilan di halaman web, serta memiliki tata letak yang rapi.		
22	BBT-022/UC-022/FR-022	User dapat menampilkan visualisasi CPMK per mata kuliah	User memilih mata kuliah tertentu dan melihat visualisasi CPMK dari sana	Positif	Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi	Sistem menampilkan visualisasi CPMK per mata kuliah untuk angkatan tertentu (grafik radar rata-rata, min, maks CPMK), tabel soal CPMK terendah, dan form untuk lihat CPMK individu.		
23	BBT-023/UC-023/FR-023	User dapat mencetak visualisasi CPMK per mata	User mencetak visualisasi	Positif	Wakil Rektor, Wakil Dekan, Kepala Program	Sistem menghasilkan <i>file</i> PDF dari visualisasi CPMK per mata kuliah. <i>File</i> PDF berisi data dan grafik		

No	Testing ID	Objective/ Output	Kasus Uji	Tipe Uji	Role	Hasil yang Diharapkan	Hasil Sebenarnya	Status Pengujian
		kuliah dalam PDF	CPMK dari mata kuliah dan hasil dalam bentuk PDF)		Studi, Dosen, Penjamin Mutu Universitas, Penjamin Mutu Fakultas, Penjamin Mutu Program Studi	yang sesuai dengan tampilan di halaman web, serta memiliki tata letak yang rapi.		

3.5.11 Skenario Pengujian Non-Fungsional

Pengujian non-fungsional dilakukan untuk memverifikasi kualitas sistem dari aspek-aspek yang tidak terkait langsung dengan fungsi bisnis, seperti usabilitas, keamanan, dan kinerja. Pengujian ini bertujuan untuk memastikan sistem tidak hanya berjalan sesuai fitur, tetapi juga andal, aman, dan nyaman digunakan. Berikut adalah skenario pengujian non-fungsional yang dirancang berdasarkan kebutuhan yang telah diidentifikasi.

Tabel 3.34 Skenario Pengujian Non-Fungsional

No	ID Kebutuhan	Aspek yang Diuji	Skenario Pengujian	Alat Bantu	Kriteria Kelulusan
1	NFR-001	<i>Usability</i> (Umpan Balik)	Verifikasi dilakukan melalui observasi langsung selama menjalankan seluruh skenario pengujian fungsional (Tabel 4.1). Setiap hasil dari aksi Simpan, Ubah, Hapus, atau saat terjadi <i>error</i> validasi akan diperiksa untuk memastikan notifikasi atau pesan kesalahan yang sesuai muncul.	Observasi Manual	Setiap aksi pengguna menghasilkan umpan balik visual (notifikasi sukses, pesan <i>error</i>) yang jelas dan informatif.
2	NFR-002	<i>Security</i> (Keamanan Akun)	1. Mencoba membuat <i>user</i> baru dengan	Observasi Manual, Inspeksi	1. Sistem menolak <i>password</i> yang

			<i>password</i> yang lemah (misal: "12345") 2. Memeriksa tabel <i>users</i> di <i>database</i> untuk memastikan kolom <i>password</i> berisi <i>string</i> acak (hasil <i>hashing</i>), bukan <i>password</i> asli.	<i>Database</i> (phpMyAdmin)	tidak memenuhi kriteria kompleksitas 2. Data <i>password</i> di <i>database</i> tersimpan dalam format <i>hash</i> dan tidak dapat dibaca secara langsung.
3	NFR-003	Usability (Navigasi)	Melakukan <i>login</i> dengan berbagai peran (misal: Admin Universitas, Dosen, Penjamin Mutu Fakultas) dan menavigasi menu yang tersedia. Verifikasi bahwa struktur menu logis, konsisten, dan hanya menampilkan fitur yang sesuai dengan hak akses peran tersebut.	Observasi Manual	Navigasi konsisten di seluruh halaman dan sesuai dengan hak akses setiap peran pengguna.
4	NFR-004	Performance (Stabilitas & Uji Beban)	Mensimulasikan beban dari 50 pengguna yang aktif bertahap dalam 25 detik (<i>ramp-up</i>) untuk menjalankan alur kerja sistem, mulai dari <i>login</i> hingga <i>generate</i> PDF.	Apache JMeter	Tingkat Error harus 1%. Waktu respons rata-rata untuk setiap langkah tetap wajar dan tidak mengalami degradasi parah. <i>Throughput</i> menunjukkan sistem mampu menangani beban.

3.5.12 Validasi *Prototype*

Validasi *prototype* pada penelitian ini dilakukan melalui pendekatan diskusi langsung bersama *stakeholder*. Setiap hasil pengembangan *prototype* diuji, kemudian dikonsultasikan dalam sesi bimbingan untuk mendapatkan masukan secara iteratif. Validasi dianggap berhasil apabila masukan telah diterapkan dan

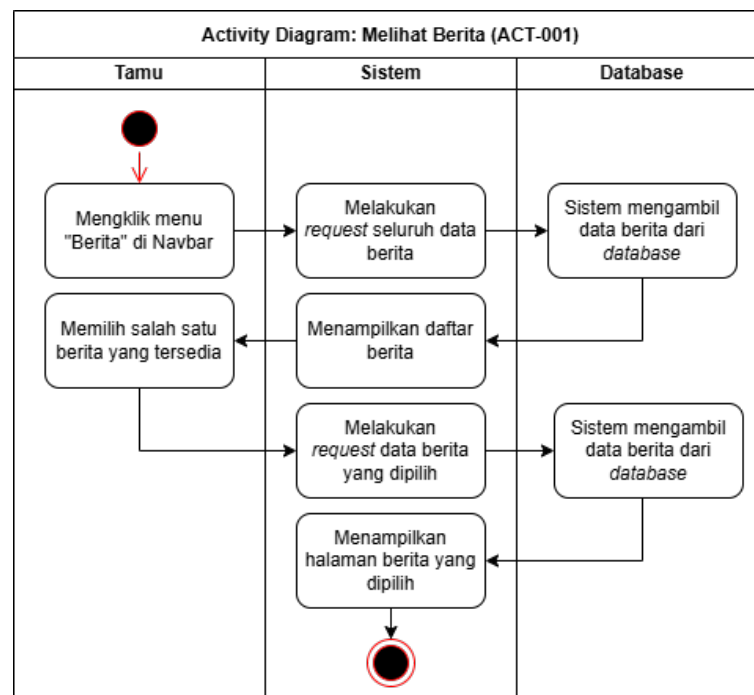
tidak terdapat koreksi lanjutan. Pendekatan ini dipilih karena proses pengembangan dilakukan secara langsung menggunakan metode *live prototyping*.

3.5.13 Activity Diagram

Activity Diagram berfungsi sebagai model fungsional yang menyediakan penjelasan lebih mendetail mengenai *use case diagram*. Beberapa *Activity Diagram* yang relevan dengan sistem ini akan dipaparkan sebagai berikut:

1. Melihat Berita (ACT-001)

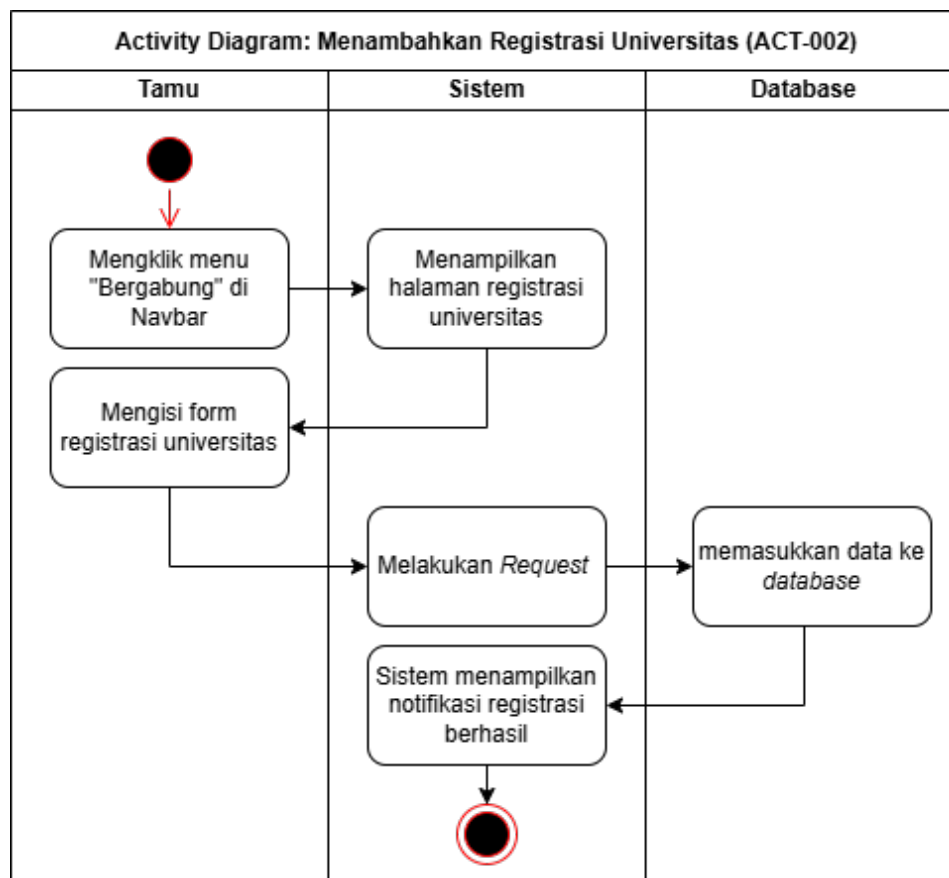
Activity Diagram pada Gambar 3.4 menjelaskan alur untuk melihat berita oleh pengguna tamu. Proses dimulai ketika tamu mengklik menu “Berita” pada navbar, lalu sistem mengirimkan *request* untuk mengambil seluruh data berita dari *database*. Setelah itu, sistem menampilkan daftar berita yang tersedia. Pengguna kemudian memilih salah satu berita, dan sistem kembali melakukan *request* untuk mengambil data berita yang dipilih dari *database*. Terakhir, sistem menampilkan halaman berisi detail berita yang dipilih oleh pengguna..



Gambar 3.4 *Activity Diagram* Melihat Berita.

2. Menambahkan Registrasi Universitas (ACT-002)

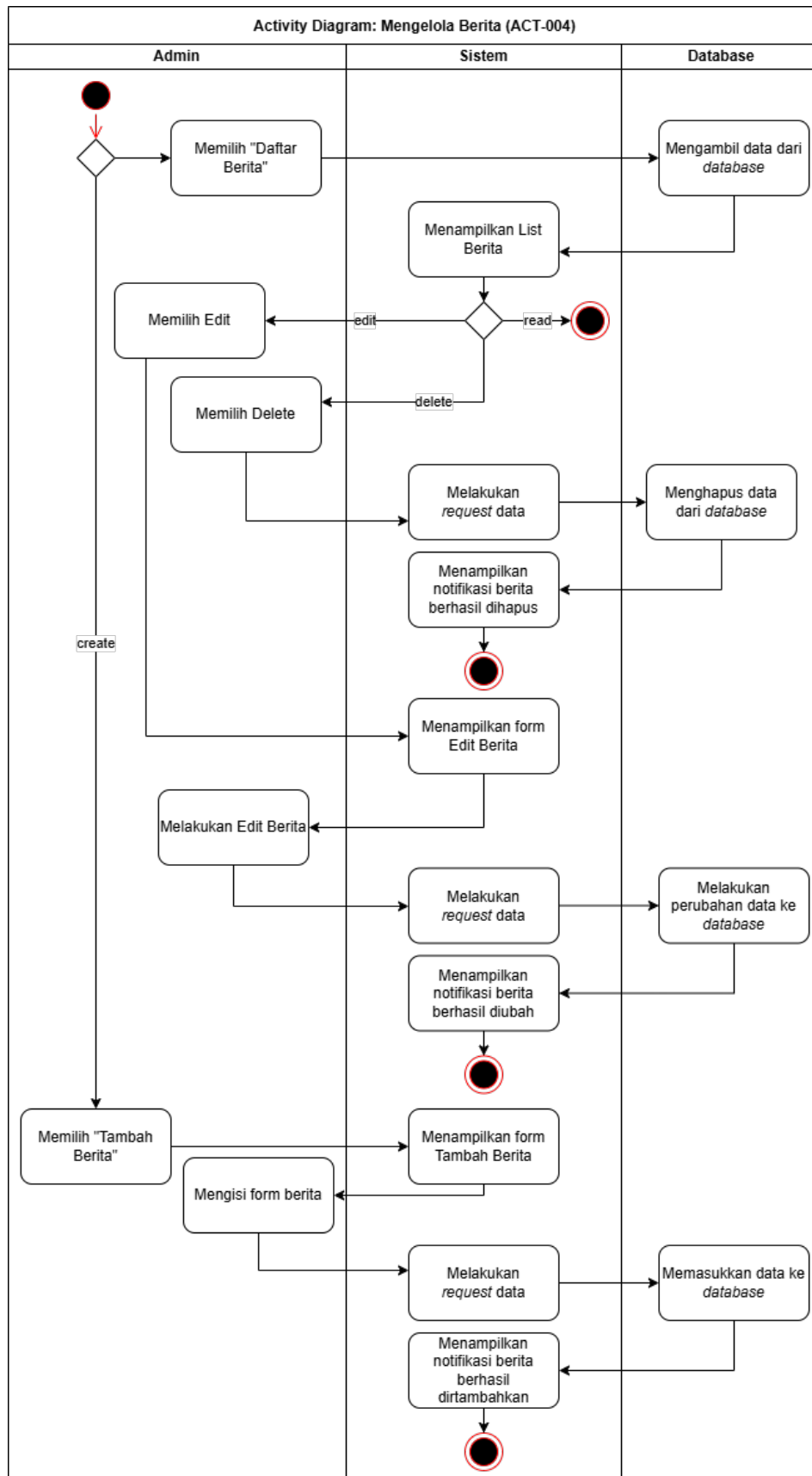
Activity Diagram pada Gambar 3.5 menjelaskan alur untuk menambahkan Registrasi Universitas oleh pengguna tamu. Proses dimulai saat tamu mengklik menu "Bergabung" pada navbar, lalu sistem menampilkan halaman *form* registrasi universitas. Pengguna kemudian mengisi form tersebut dan mengirimkan data melalui tombol *submit*. Setelah itu, sistem memproses *request* dan menyimpan data ke dalam *database*. Jika proses penyimpanan berhasil, sistem akan menampilkan notifikasi bahwa registrasi universitas telah berhasil dilakukan.



Gambar 3.5 *Activity Diagram* Menambahkan Registrasi Universitas

3. Melakukan *Login* (ACT-003)

Activity Diagram pada Gambar 3.6 menjelaskan alur aktivitas pengguna saat melakukan proses *login* ke dalam sistem. Proses diawali dengan sistem menampilkan halaman *login* yang menyediakan dua pilihan: mengisi form secara manual dengan email dan *password*, atau menggunakan autentikasi



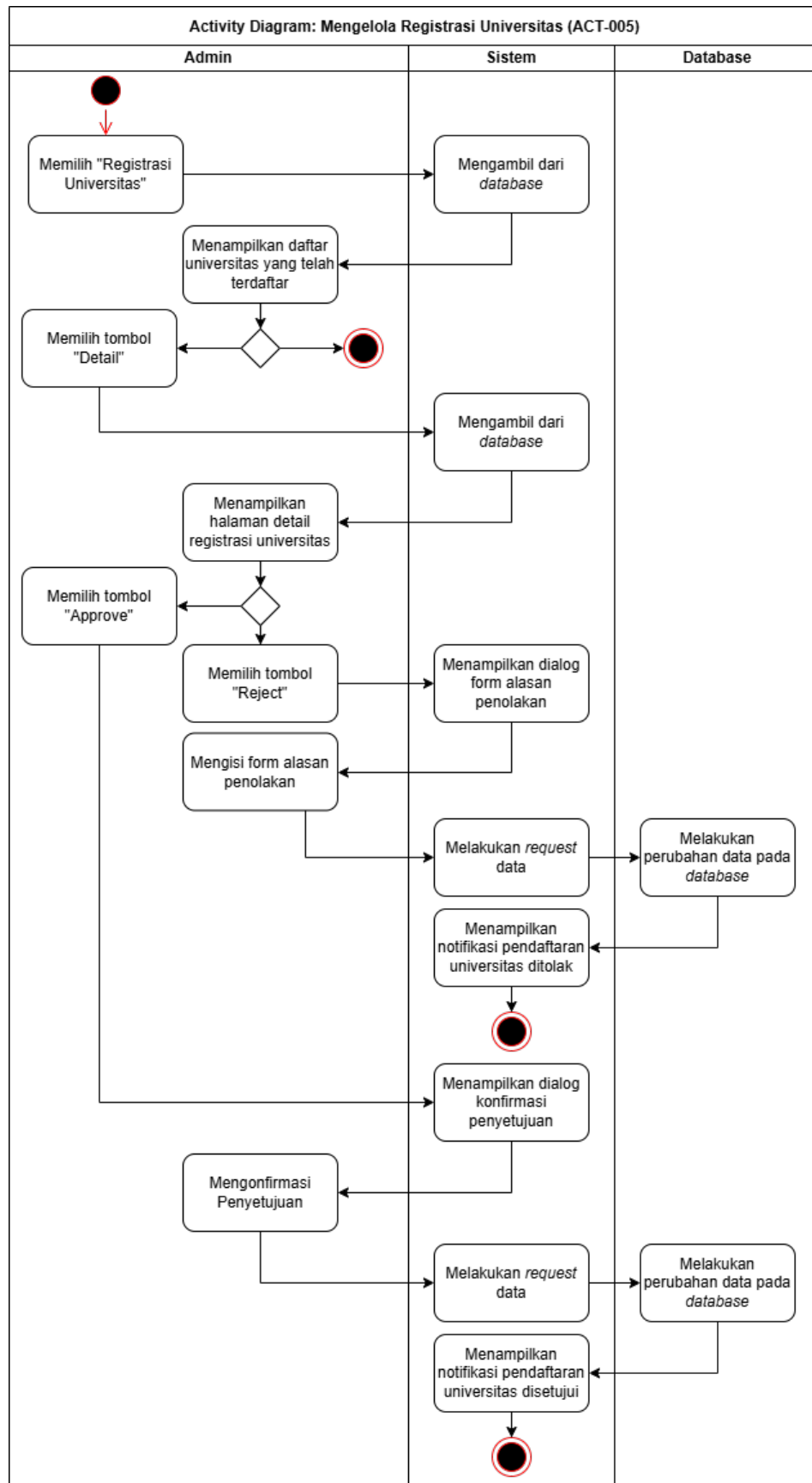
Gambar 3.7 Activity Diagram mengelola Berita

5. Mengelola Registrasi Universitas (ACT-005)

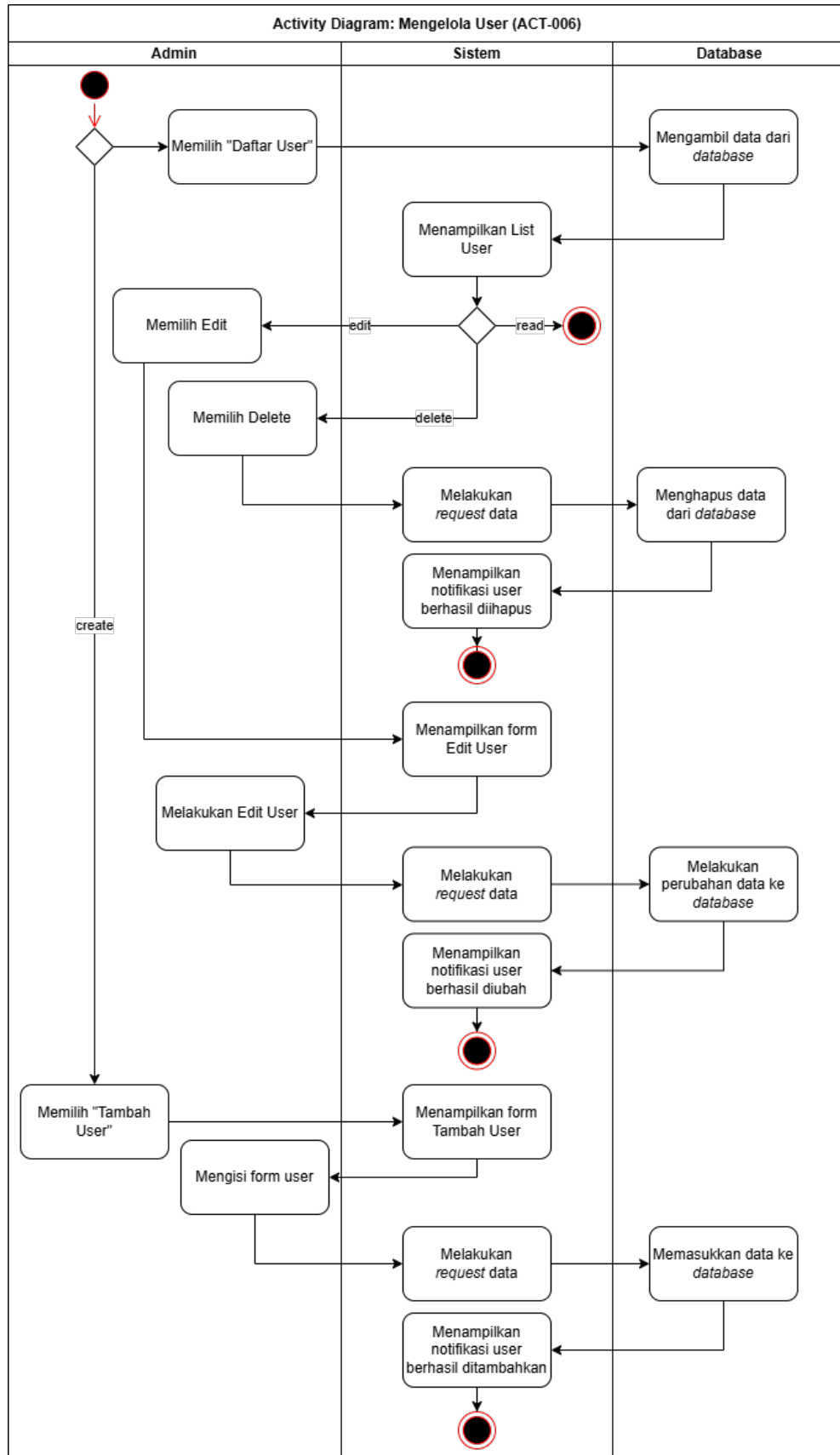
Activity Diagram pada Gambar 3.8 menjelaskan alur pengelolaan registrasi universitas dalam sistem oleh admin. Proses dimulai ketika admin memilih menu "Registrasi Universitas," kemudian sistem mengambil data dari *database* dan menampilkan daftar universitas yang telah terdaftar. Admin dapat memilih tombol "Detail" untuk melihat informasi lebih lanjut, lalu diberikan opsi untuk menyetujui atau menolak registrasi. Jika admin memilih "Reject," sistem menampilkan *form* alasan penolakan yang harus diisi sebelum mengirim permintaan perubahan data ke *database* dan menampilkan notifikasi penolakan. Jika admin memilih "Approve," sistem menampilkan dialog konfirmasi penyetujuan sebelum mengirim permintaan perubahan data ke *database* dan menampilkan notifikasi persetujuan.

6. Mengelola User (ACT-005)

Activity Diagram pada Gambar 3.9 menjelaskan alur pengelolaan pengguna dalam sistem oleh admin. Proses dimulai ketika admin memilih menu "Daftar User," kemudian sistem mengambil data pengguna dari *database* dan menampilkan daftar *user* yang telah terdaftar. Admin dapat memilih salah satu tindakan, yaitu "Edit," "Delete," atau "Tambah User." Jika admin memilih "Edit," sistem menampilkan *form* Edit *User*, lalu admin mengubah data dan mengirim permintaan perubahan ke *database*. Setelah perubahan berhasil, sistem menampilkan notifikasi bahwa data pengguna telah diperbarui. Jika admin memilih "Delete," sistem mengirim permintaan penghapusan ke *database*, lalu setelah data berhasil dihapus, sistem menampilkan notifikasi bahwa pengguna telah dihapus. Jika admin memilih "Tambah User," sistem menampilkan *form* Tambah *User*, kemudian admin mengisi data pengguna baru dan mengirim permintaan penyimpanan ke *database*. Setelah data berhasil disimpan, sistem menampilkan notifikasi bahwa pengguna baru telah ditambahkan.



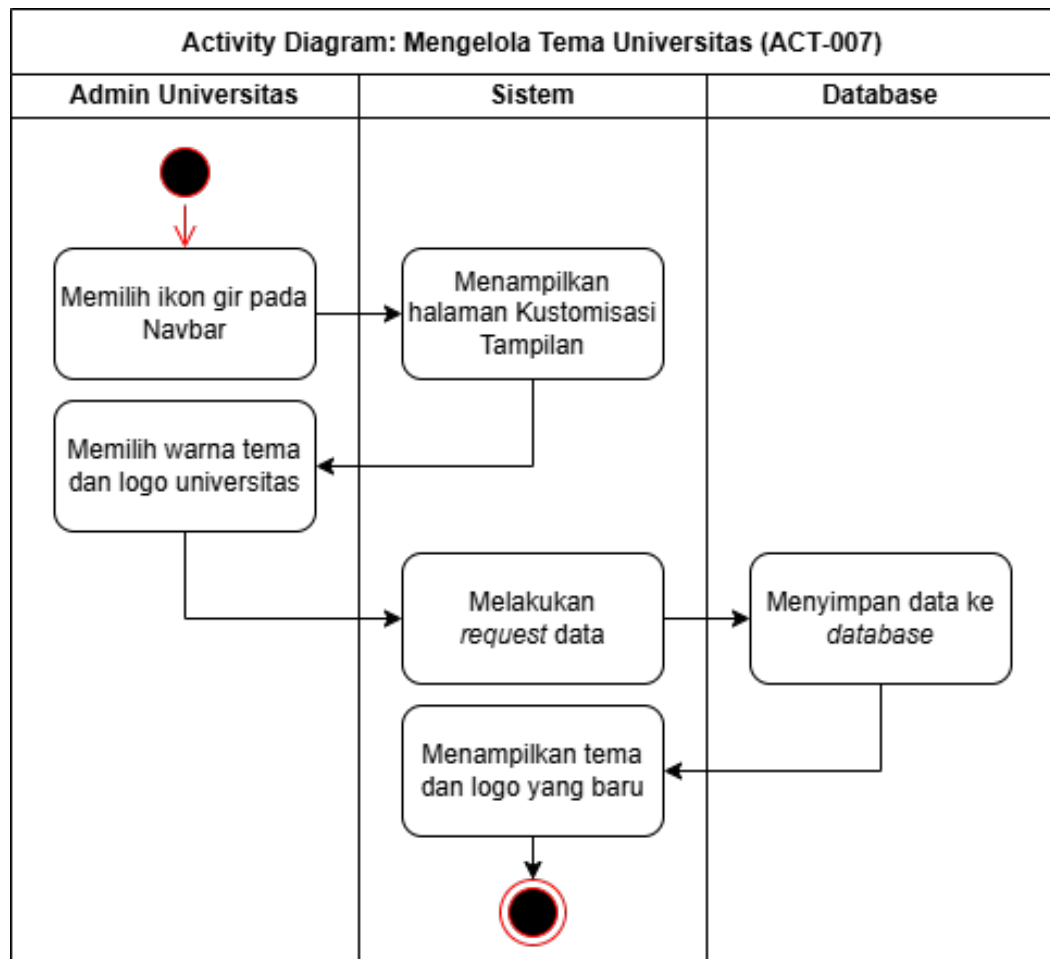
Gambar 3.8 Activity Diagram Mengelola Registrasi Universitas.



Gambar 3.9 *Activity Diagram* Mengelola User.

7. Mengelola Tema Universitas (ACT-006)

Activity Diagram pada Gambar 3.10 menjelaskan alur pengelolaan tema universitas oleh Admin Universitas dalam sistem. Proses dimulai ketika Admin Universitas memilih ikon gir pada Navbar, kemudian sistem menampilkan halaman Kustomisasi Tampilan. Admin Universitas dapat memilih warna tema dan logo universitas sesuai kebijakan institusi. Setelah melakukan pemilihan, sistem mengirimkan *request* data untuk menyimpan perubahan ke *database*. Setelah data berhasil disimpan, sistem menampilkan tema dan logo yang baru sesuai dengan perubahan yang telah dilakukan oleh Admin Universitas.



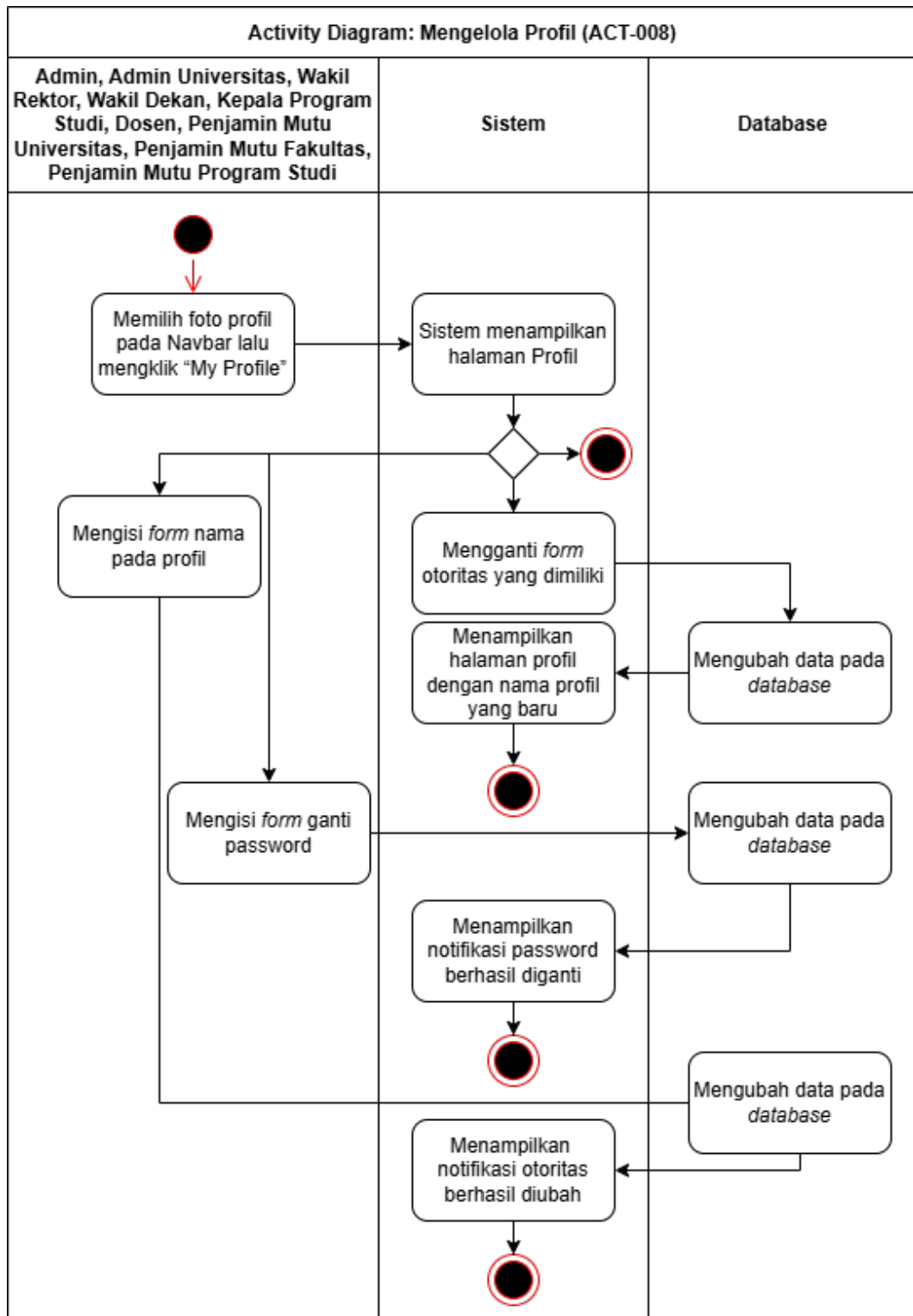
Gambar 3.10 *Activity Diagram* Mengelola Tema Universitas.

8. Mengelola Profil (ACT-008)

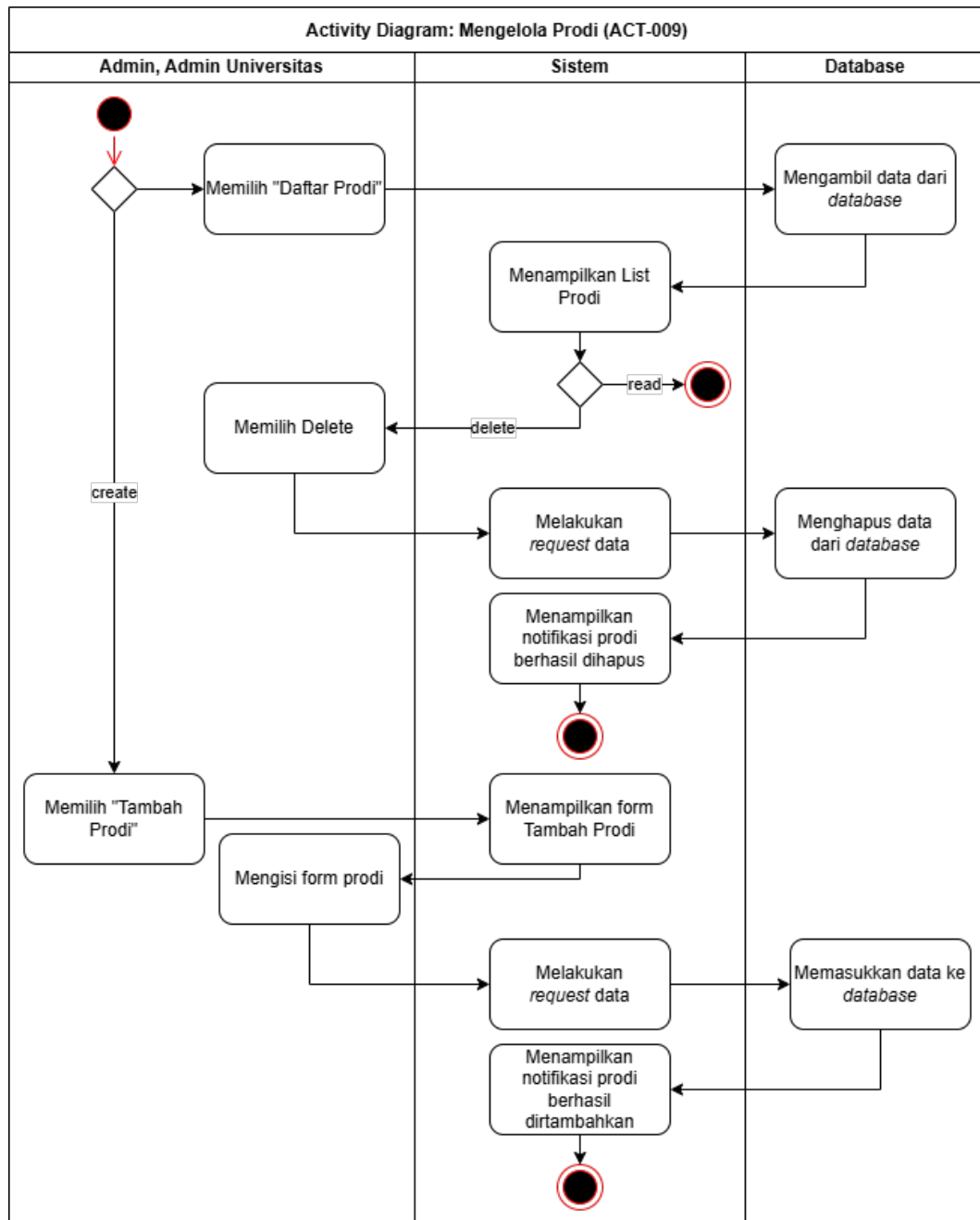
Activity Diagram pada Gambar 3.11 menjelaskan proses mengelola profil oleh pengguna dalam sistem. Aktivitas dimulai ketika pengguna memilih foto profil pada navbar dan mengklik “My Profile”, kemudian sistem menampilkan halaman profil. Di halaman ini, pengguna dapat melakukan berbagai pengelolaan profil seperti mengganti nama profil dengan mengisi *form* yang disediakan, mengganti *password*, serta memilih otoritas aktif apabila memiliki lebih dari satu. Setiap perubahan yang dilakukan akan disimpan ke dalam *database* dan sistem akan memberikan notifikasi bahwa perubahan berhasil dilakukan, seperti notifikasi penggantian nama, *password*, maupun otoritas. Diagram ini mencerminkan alur lengkap yang mencakup aktivitas melihat, mengubah, dan menyimpan data profil pengguna.

9. Mengelola Prodi (ACT-009)

Activity Diagram pada Gambar 3.12 menjelaskan proses mengelola prodi dalam sistem, yang mencakup aktivitas melihat, menambah, dan menghapus data prodi. Proses dimulai ketika pengguna memilih menu "Daftar Prodi", kemudian sistem menampilkan daftar prodi dengan mengambil data dari *database*. Pengguna dapat memilih untuk menghapus salah satu prodi, yang akan memicu penghapusan data dari *database* dan sistem menampilkan notifikasi bahwa prodi berhasil dihapus. Selain itu, pengguna juga dapat memilih untuk menambahkan prodi dengan mengklik "Tambah Prodi", lalu mengisi *form* yang disediakan. Sistem kemudian memproses permintaan tersebut dan memasukkan data ke dalam *database*, serta menampilkan notifikasi bahwa prodi berhasil ditambahkan.



Gambar 3.11 Activity Diagram Mengelola Profil.

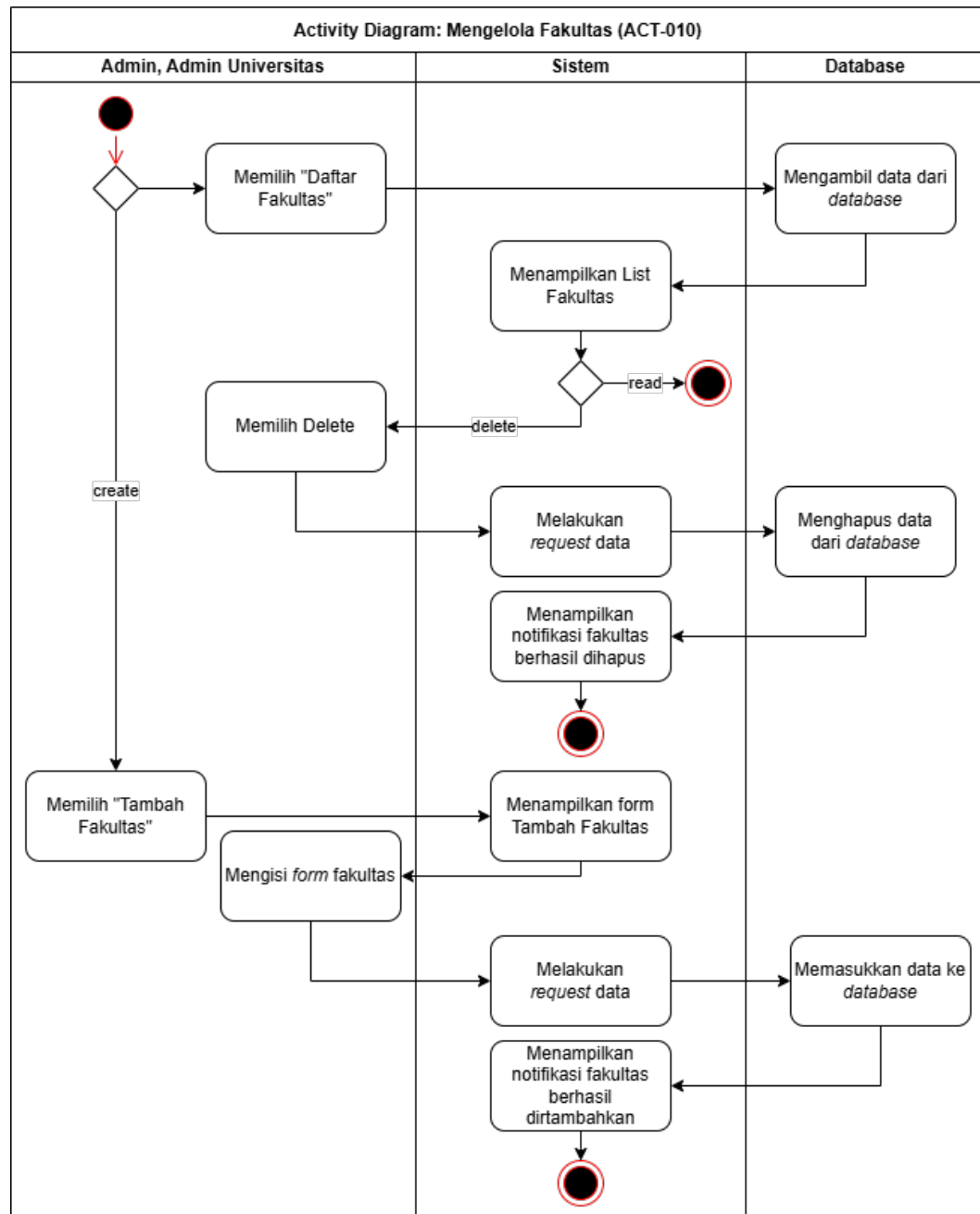


Gambar 3.12 Activity Diagram Mengelola Prodi.

10. Mengelola Fakultas (ACT-010)

Activity Diagram pada Gambar 3.13 menjelaskan proses mengelola Fakultas dalam sistem, yang mencakup aktivitas melihat, menambah, dan menghapus data fakultas. Proses dimulai ketika pengguna memilih menu "Daftar Fakultas", kemudian sistem menampilkan daftar fakultas dengan mengambil data dari *database*. Pengguna dapat memilih untuk menghapus salah satu fakultas, yang

akan memicu penghapusan data dari *database* dan sistem menampilkan notifikasi bahwa fakultas berhasil dihapus. Selain itu, pengguna juga dapat memilih untuk menambahkan fakultas dengan mengklik "Tambah Fakultas", lalu mengisi *form* yang disediakan. Sistem kemudian memproses permintaan tersebut dan memasukkan data ke dalam *database*, serta menampilkan notifikasi bahwa fakultas berhasil ditambahkan.



Gambar 3.13 Activity Diagram Mengelola Fakultas.

11. Mengelola Universitas (ACT-011)

Activity Diagram pada Gambar 3.14 menjelaskan proses mengelola Universitas dalam sistem, yang mencakup aktivitas melihat, menambah, dan menghapus data universitas. Proses dimulai ketika pengguna memilih menu "Daftar Universitas", kemudian sistem menampilkan daftar universitas dengan mengambil data dari *database*. Pengguna dapat memilih untuk menghapus salah satu universitas, yang akan memicu penghapusan data dari *database* dan sistem menampilkan notifikasi bahwa universitas berhasil dihapus. Selain itu, pengguna juga dapat memilih untuk menambahkan universitas dengan mengklik "Tambah Universitas", lalu mengisi *form* yang disediakan. Sistem kemudian memproses permintaan tersebut dan memasukkan data ke dalam *database*, serta menampilkan notifikasi bahwa universitas berhasil ditambahkan.

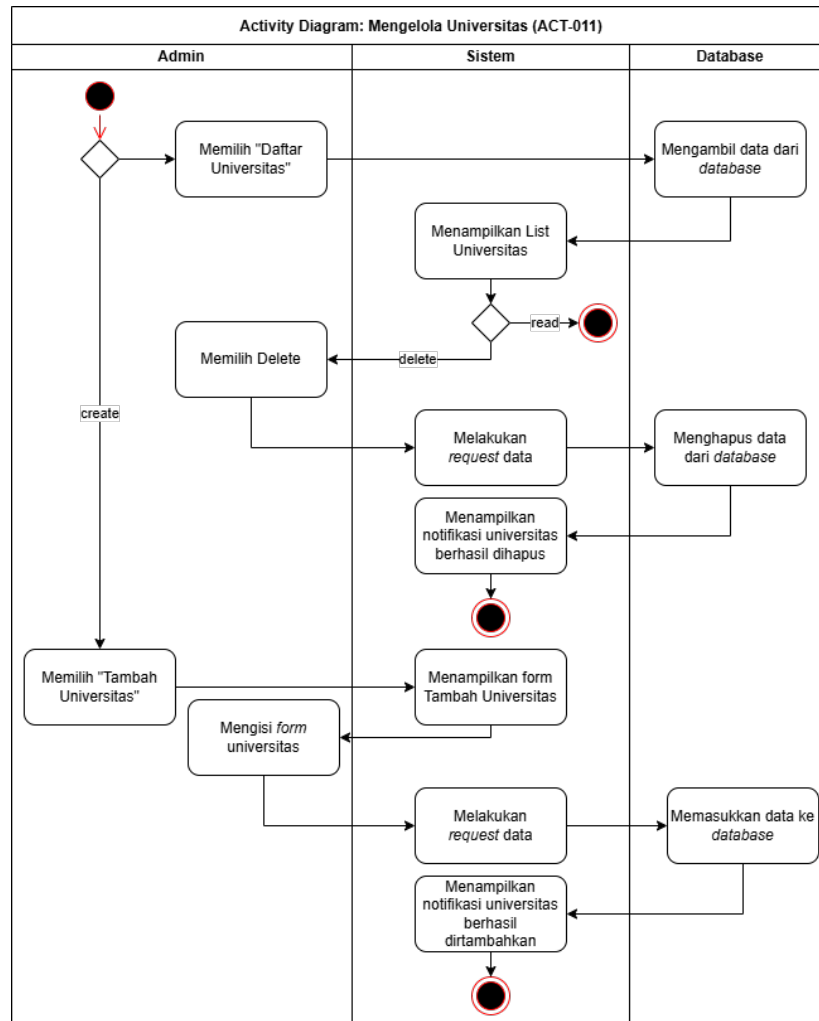
12. Mengelola Mahasiswa (ACT-012)

Activity Diagram pada Gambar 3.15 menjelaskan alur melihat Mahasiswa dalam sistem. Proses dimulai ketika *user* memilih menu "Daftar Mahasiswa," kemudian sistem mengambil data mahasiswa dari *database* dan menampilkan daftar mahasiswa yang telah terdaftar.

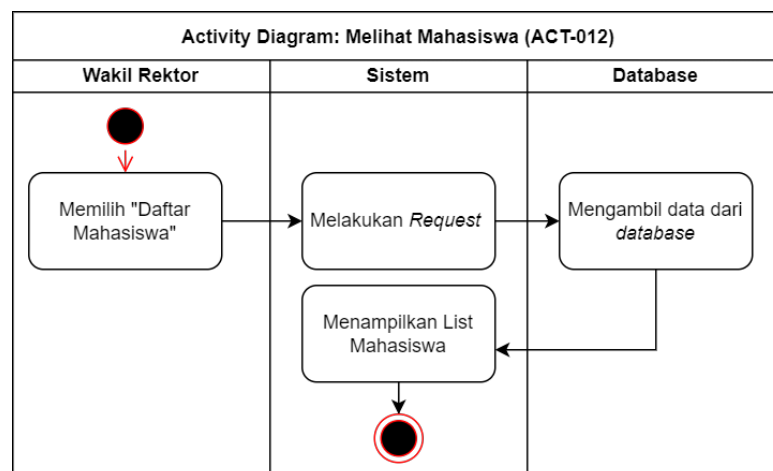
13. Mengelola Mahasiswa (ACT-013)

Activity Diagram pada Gambar 3.16 menjelaskan alur pengelolaan Mahasiswa dalam sistem. Pengguna dapat memilih salah satu tindakan, yaitu "Edit," "Delete," atau "Tambah Mahasiswa." Jika pengguna memilih "Edit," sistem menampilkan *form* Edit Mahasiswa, lalu *user* mengubah data dan mengirim permintaan perubahan ke *database*. Setelah perubahan berhasil, sistem menampilkan notifikasi bahwa data pengguna telah diperbarui. Jika pengguna memilih "Delete," sistem mengirim permintaan penghapusan ke *database*, lalu setelah data berhasil dihapus, sistem menampilkan notifikasi bahwa pengguna telah dihapus. Jika *user* memilih "Tambah Mahasiswa," sistem menampilkan *form* Tambah Mahasiswa, kemudian *user* mengisi data pengguna baru dan mengirim permintaan penyimpanan ke *database*. Setelah data berhasil

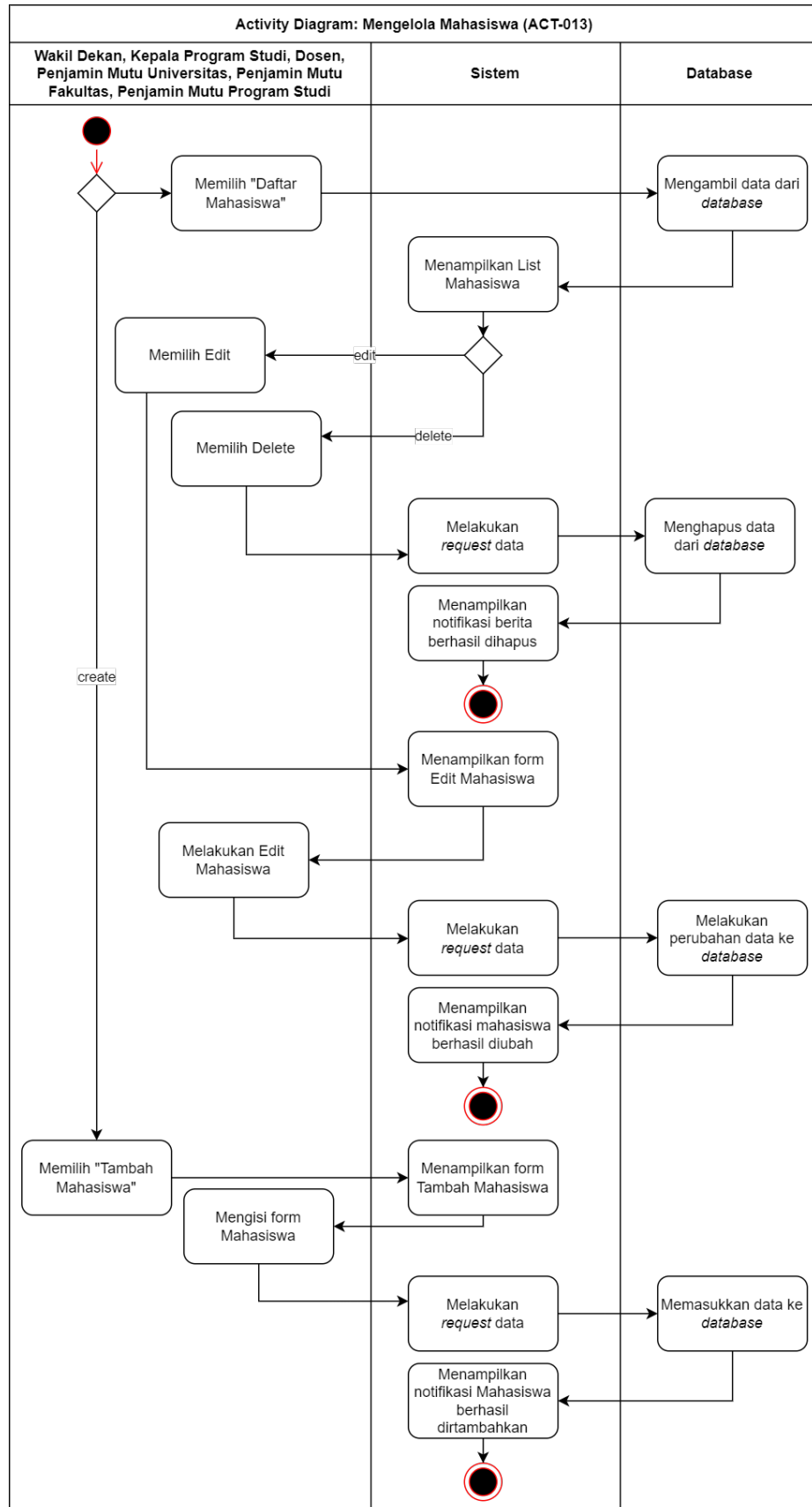
disimpan, sistem menampilkan notifikasi bahwa mahasiswa baru telah ditambahkan.



Gambar 3.14 Activity Diagram Mengelola Universitas.



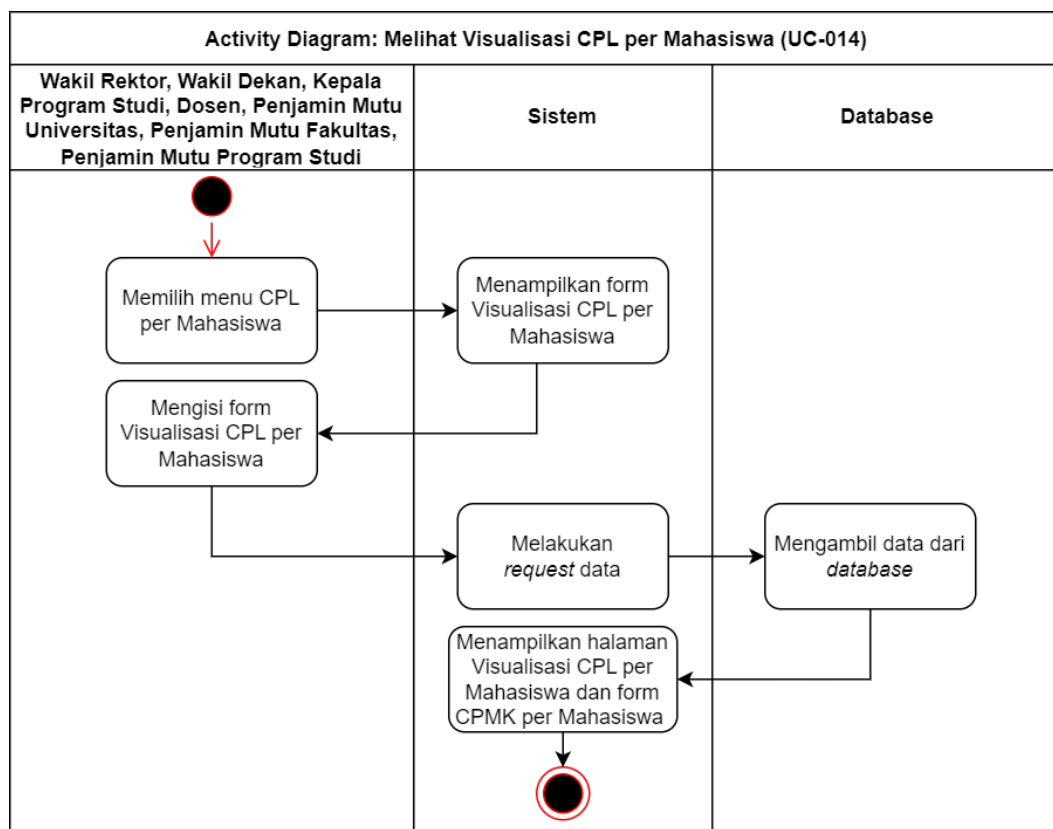
Gambar 3.15 Activity Diagram Melihat Mahasiswa.



Gambar 3.16 Activity Diagram Mengelola Mahasiswa.

14. Melihat Visualisasi CPL per Mahasiswa (ACT-014)

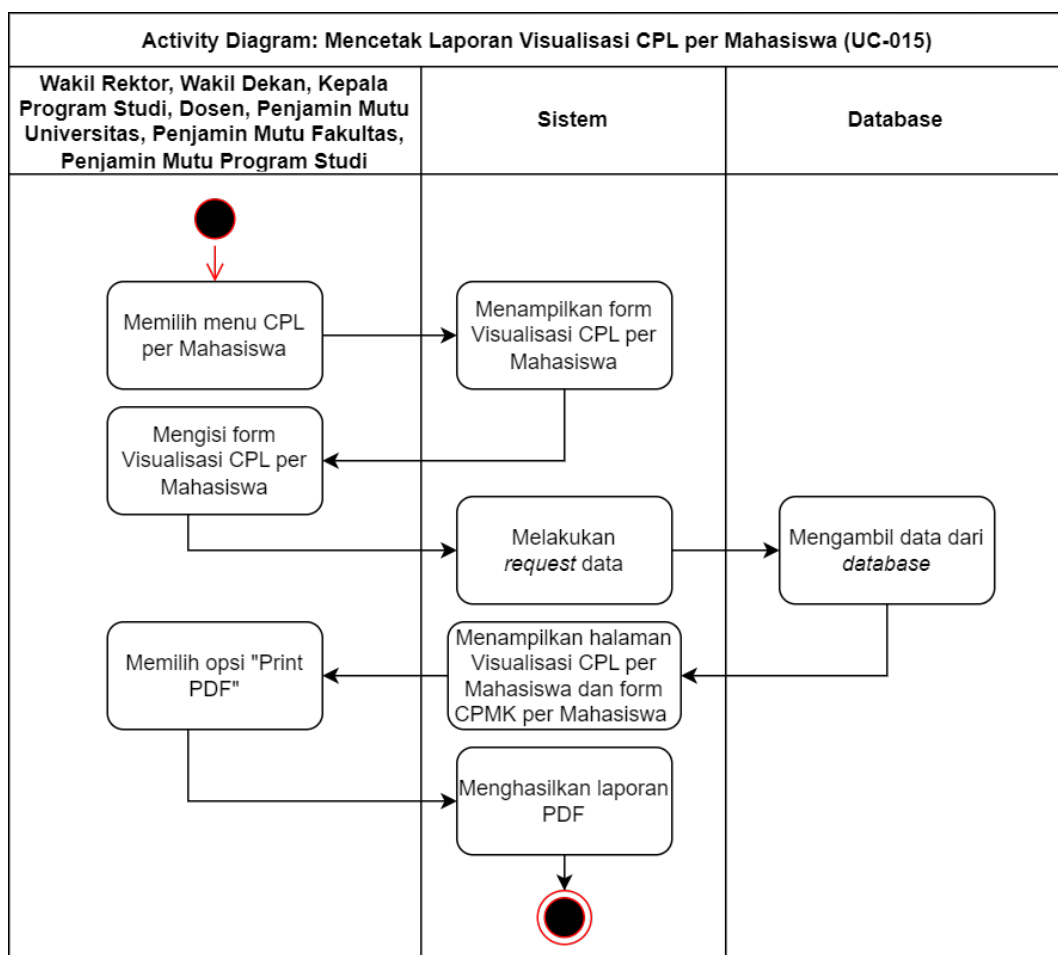
Activity Diagram pada Gambar 3.17 menjelaskan alur proses melihat visualisasi Capaian Pembelajaran Lulusan (CPL) per mahasiswa oleh Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, dan Penjamin Mutu dalam sistem. Proses dimulai ketika pengguna memilih menu CPL per Mahasiswa, kemudian sistem menampilkan *form* Visualisasi CPL per Mahasiswa. Pengguna mengisi *form* tersebut dengan data yang diperlukan, lalu sistem melakukan *request* data ke *database*. *Database* mengambil data yang relevan dan mengirimkannya kembali ke sistem. Setelah data diterima, sistem menampilkan halaman Visualisasi CPL per Mahasiswa beserta *form* Capaian Pembelajaran Mata Kuliah (CPMK) per Mahasiswa.



Gambar 3.17 *Activity Diagram* Melihat Visualisasi CPL per Mahasiswa.

15. Mencetak Laporan Visualisasi CPL per Mahasiswa (ACT-015)

Activity Diagram pada Gambar 3.18 menjelaskan alur proses pencetakan laporan visualisasi Capaian Pembelajaran Lulusan (CPL) per mahasiswa oleh Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, dan Penjamin Mutu dalam sistem. Proses dimulai ketika pengguna memilih menu CPL per Mahasiswa, kemudian sistem menampilkan *form* Visualisasi CPL per Mahasiswa. Pengguna mengisi *form* tersebut dengan data yang diperlukan, lalu sistem melakukan *request* data ke *database*. *Database* mengambil data yang relevan dan mengirimkannya kembali ke sistem. Setelah data diterima, sistem menampilkan halaman Visualisasi CPL per Mahasiswa beserta *form* Capaian Pembelajaran Mata Kuliah (CPMK) per Mahasiswa. Pengguna kemudian memilih opsi "Print PDF," yang menyebabkan sistem menghasilkan laporan dalam format PDF.



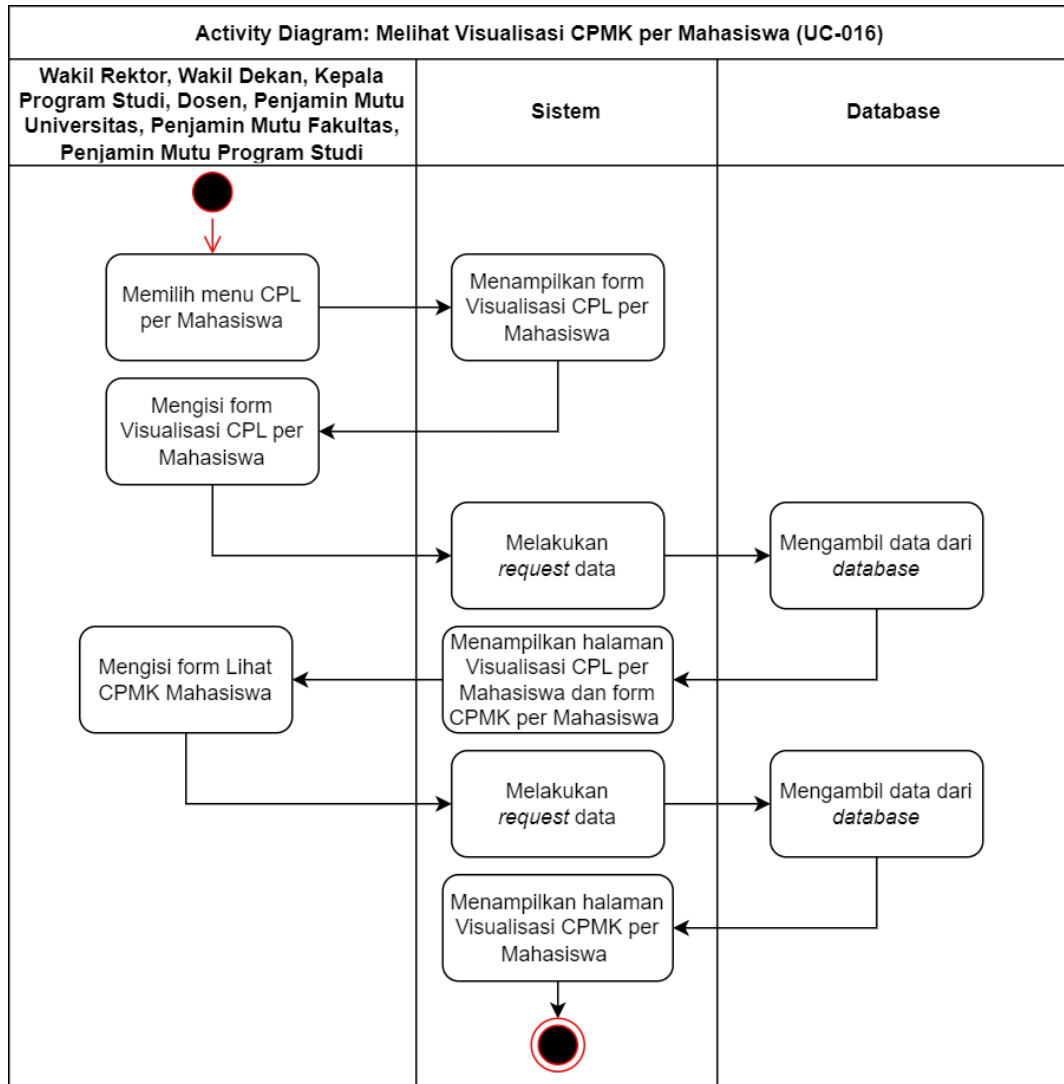
Gambar 3.18 *Activity Diagram* Mencetak Laporan Visualisasi CPL Mahasiswa per Mahasiswa.

16. Melihat Visualisasi CPMK per Mahasiswa (ACT-016)

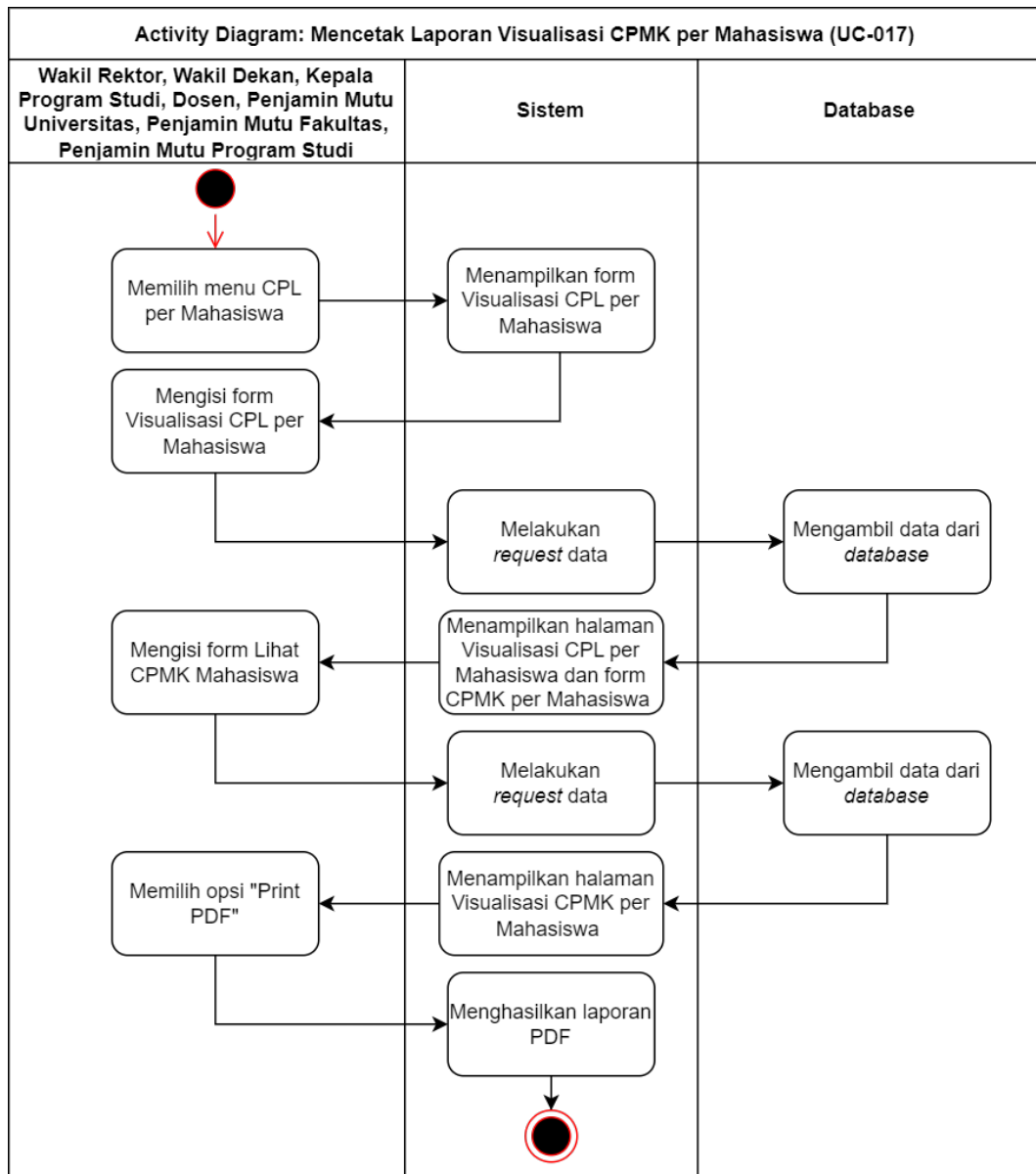
Activity Diagram pada Gambar 3.19 menjelaskan alur proses melihat visualisasi Capaian Pembelajaran Mata Kuliah (CPMK) per mahasiswa oleh Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, dan Penjamin Mutu dalam sistem. Proses dimulai ketika pengguna memilih menu CPL per Mahasiswa, kemudian sistem menampilkan *form* Visualisasi CPL per Mahasiswa. Pengguna mengisi *form* tersebut dengan data yang diperlukan, lalu sistem melakukan *request* data ke *database*. *Database* mengambil data yang relevan dan mengirimkannya kembali ke sistem. Setelah data diterima, sistem menampilkan halaman Visualisasi CPL per Mahasiswa beserta *form* Lihat CPMK Mahasiswa. Pengguna kemudian mengisi *form* Lihat CPMK Mahasiswa, lalu sistem kembali melakukan *request* data ke *database*. Setelah *database* mengambil dan mengirimkan data CPMK yang relevan, sistem menampilkan halaman Visualisasi CPMK per Mahasiswa.

17. Mencetak Laporan Visualisasi CPMK per Mahasiswa (ACT-017)

Activity Diagram pada Gambar 3.20 menjelaskan alur proses pencetakan laporan visualisasi Capaian Pembelajaran Mata Kuliah (CPMK) per mahasiswa oleh Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, dan Penjamin Mutu dalam sistem. Proses dimulai ketika pengguna memilih menu CPL per Mahasiswa, kemudian sistem menampilkan *form* Visualisasi CPL per Mahasiswa. Pengguna mengisi *form* tersebut dengan data yang diperlukan, lalu sistem melakukan *request* data ke *database*. *Database* mengambil data yang relevan dan mengirimkannya kembali ke sistem. Setelah data diterima, sistem menampilkan halaman Visualisasi CPL per Mahasiswa beserta *form* Lihat CPMK Mahasiswa. Pengguna kemudian mengisi *form* Lihat CPMK Mahasiswa, lalu sistem kembali melakukan *request* data ke *database*. Setelah *database* mengambil dan mengirimkan data CPMK yang relevan, sistem menampilkan halaman Visualisasi CPMK per Mahasiswa. Pengguna kemudian memilih opsi "Print PDF," yang menyebabkan sistem menghasilkan laporan dalam format PDF.



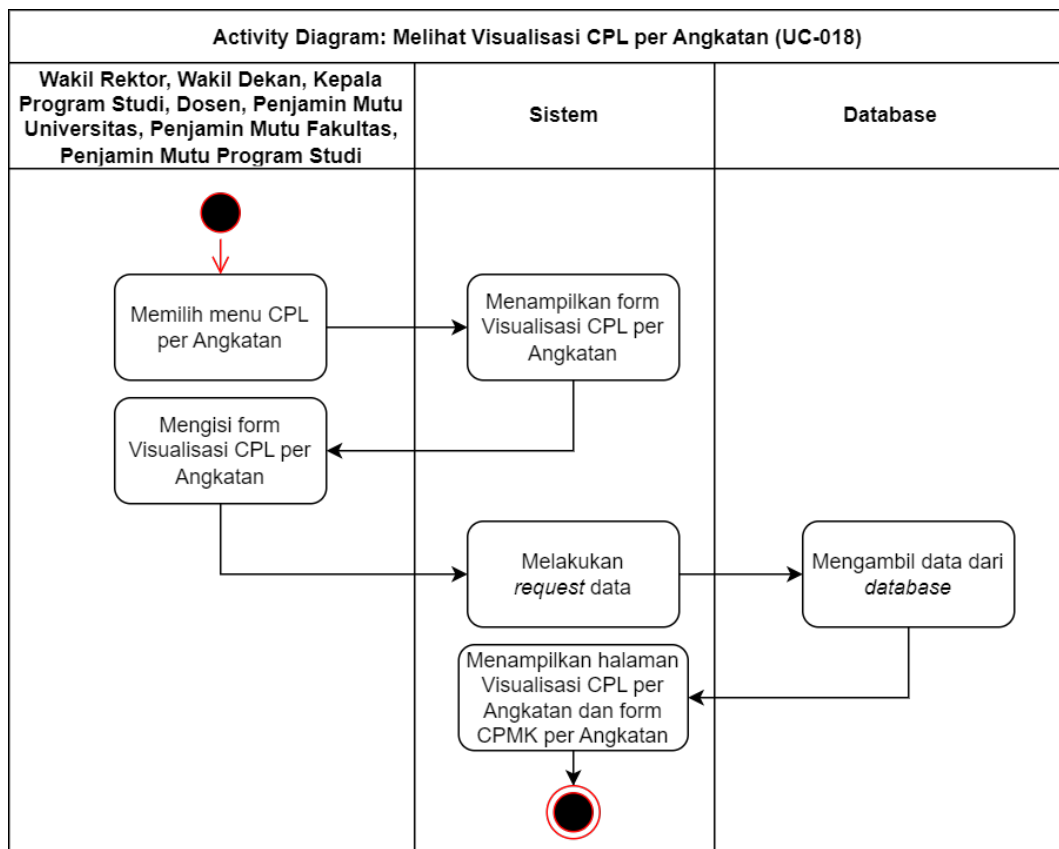
Gambar 3.19 Activity Diagram Melihat Visualisasi CPMK per Mahasiswa.



Gambar 3.20 Activity Diagram Mencetak Laporan Visualisasi CPMK per Mahasiswa.

18. Melihat Visualisasi CPL per Angkatan (ACT-018)

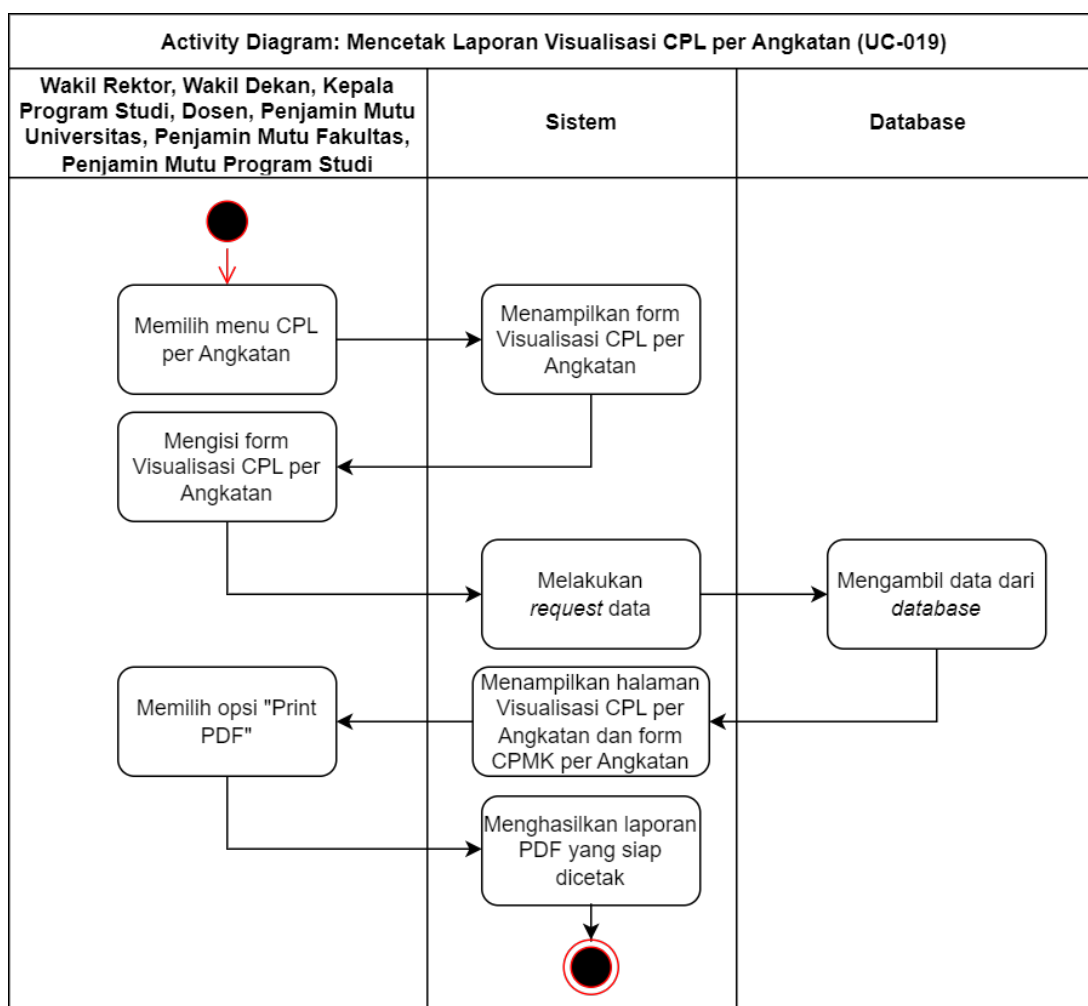
Activity Diagram pada Gambar 3.21 menjelaskan alur proses melihat visualisasi Capaian Pembelajaran Lulusan (CPL) per angkatan oleh Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, dan Penjamin Mutu dalam sistem. Proses dimulai ketika pengguna memilih menu CPL per Angkatan, kemudian sistem menampilkan *form* Visualisasi CPL per Angkatan. Pengguna mengisi *form* tersebut dengan data yang diperlukan, lalu sistem melakukan *request* data ke *database*. *Database* mengambil data yang relevan dan mengirimkannya kembali ke sistem. Setelah data diterima, sistem menampilkan halaman Visualisasi CPL per Angkatan beserta *form* Capaian Pembelajaran Mata Kuliah (CPMK) per Angkatan.



Gambar 3.21 *Activity Diagram* Melihat Visualisasi CPL per Angkatan

19. Mencetak Laporan Visualisasi CPL per Angkatan (UC-019)

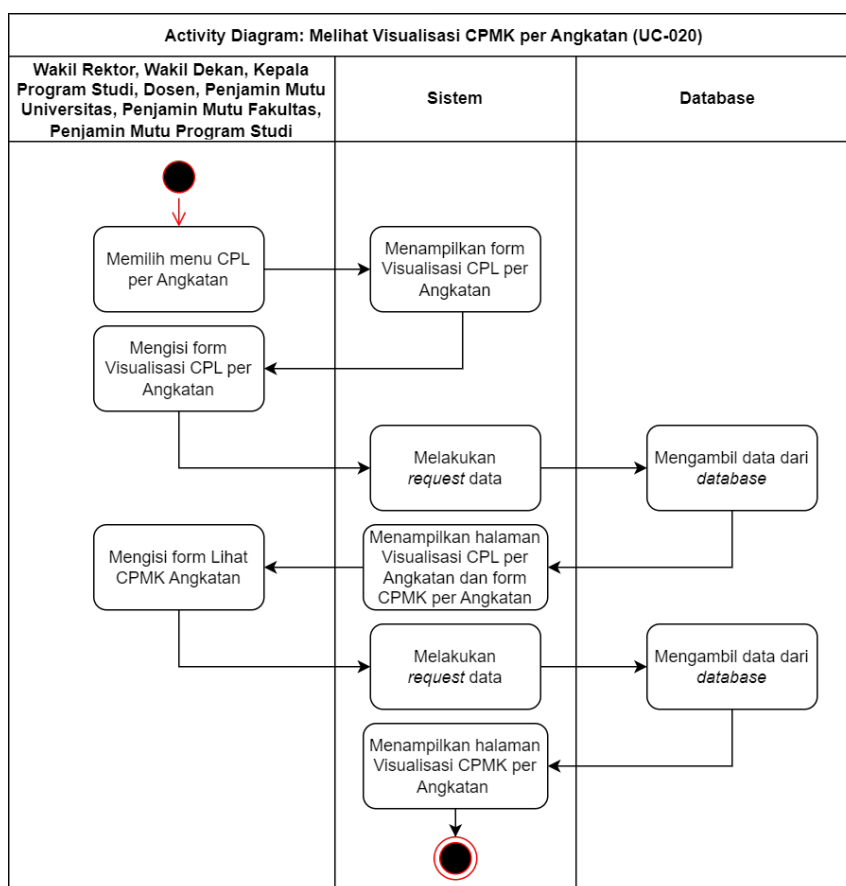
Activity Diagram pada Gambar 3.22 menjelaskan alur proses pencetakan laporan visualisasi Capaian Pembelajaran Lulusan (CPL) per angkatan oleh Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, dan Penjamin Mutu dalam sistem. Proses dimulai ketika pengguna memilih menu CPL per Angkatan, kemudian sistem menampilkan form Visualisasi CPL per Angkatan. Pengguna mengisi form tersebut dengan data yang diperlukan, lalu sistem melakukan *request* data ke *database*. *Database* mengambil data yang relevan dan mengirimkannya kembali ke sistem. Setelah data diterima, sistem menampilkan halaman Visualisasi CPL per Angkatan beserta form Capaian Pembelajaran Mata Kuliah (CPMK) per Angkatan. Pengguna kemudian memilih opsi "Print PDF," yang menyebabkan sistem menghasilkan laporan dalam format PDF.



Gambar 3.22 *Activity Diagram* mencetak laporan CPL per Mahasiswa.

20. Melihat Visualisasi CPMK per Angkatan (ACT-020)

Activity Diagram pada Gambar 3.23 menjelaskan alur proses melihat visualisasi Capaian Pembelajaran Mata Kuliah (CPMK) per mahasiswa oleh Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, dan Penjamin Mutu dalam sistem. Proses dimulai ketika pengguna memilih menu CPL per Mahasiswa, kemudian sistem menampilkan *form* Visualisasi CPL per Mahasiswa. Pengguna mengisi *form* tersebut dengan data yang diperlukan, lalu sistem melakukan *request* data ke *database*. *Database* mengambil data yang relevan dan mengirimkannya kembali ke sistem. Setelah data diterima, sistem menampilkan halaman Visualisasi CPL per Mahasiswa beserta *form* Lihat CPMK Mahasiswa. Pengguna kemudian mengisi *form* Lihat CPMK Mahasiswa, lalu sistem kembali melakukan *request* data ke *database*. Setelah *database* mengambil dan mengirimkan data CPMK yang relevan, sistem menampilkan halaman Visualisasi CPMK per Mahasiswa.



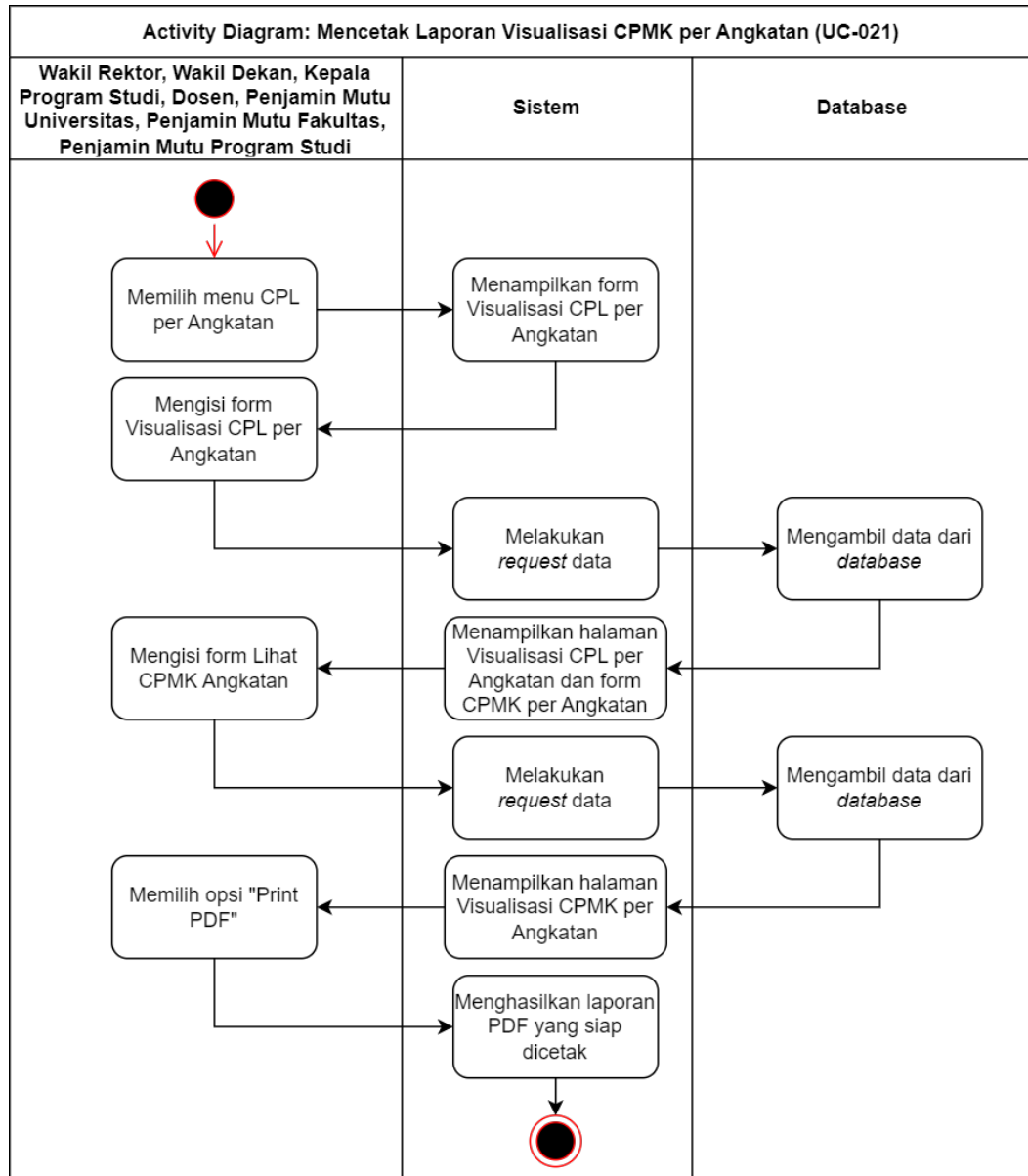
Gambar 3.23 *Activity Diagram* melihat data CPMK per Mahasiswa.

21. Mencetak Laporan Visualisasi CPMK per Angkatan (ACT-021)

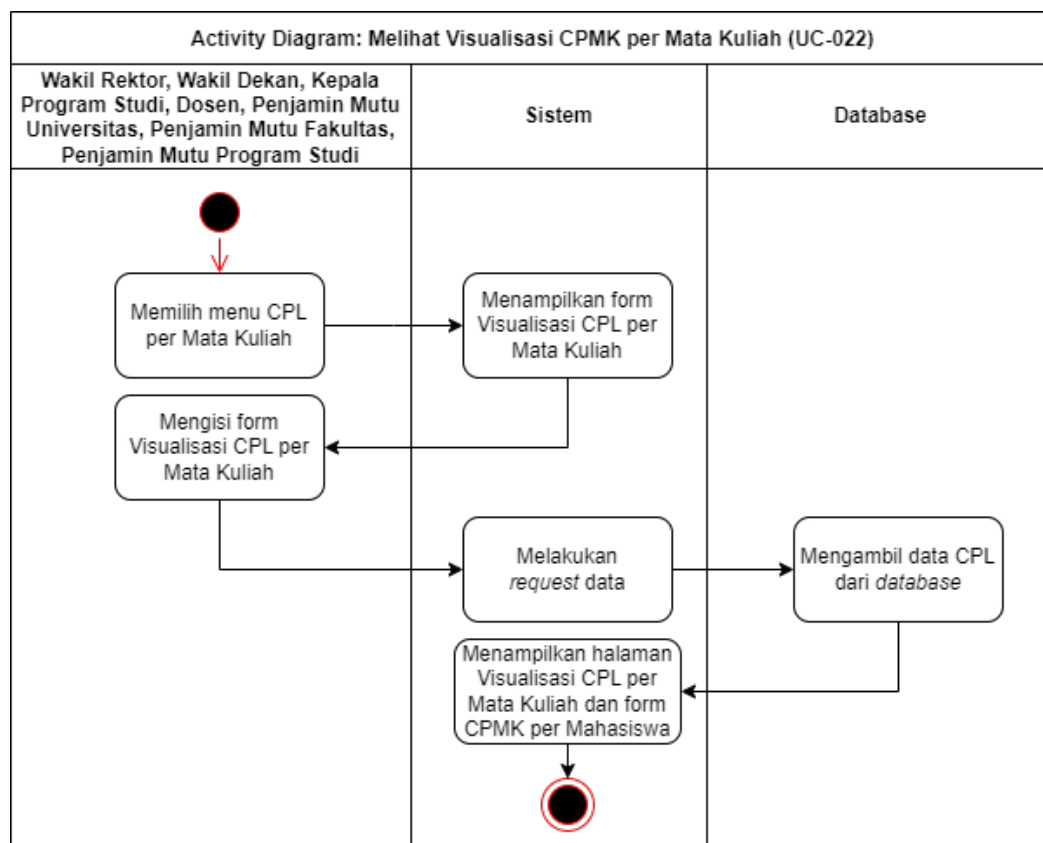
Activity Diagram pada Gambar 3.24 menjelaskan alur proses pencetakan laporan visualisasi Capaian Pembelajaran Mata Kuliah (CPMK) per angkatan oleh Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, dan Penjamin Mutu dalam sistem. Proses dimulai ketika pengguna memilih menu CPL per Angkatan, kemudian sistem menampilkan *form* Visualisasi CPL per Angkatan. Pengguna mengisi *form* tersebut dengan data yang diperlukan, lalu sistem melakukan *request* data ke *database*. *Database* mengambil data yang relevan dan mengirimkannya kembali ke sistem. Setelah data diterima, sistem menampilkan halaman Visualisasi CPL per Mahasiswa beserta *form* Lihat CPMK Mahasiswa. Pengguna kemudian mengisi *form* Lihat CPMK Mahasiswa, lalu sistem kembali melakukan *request* data ke *database*. Setelah *database* mengambil dan mengirimkan data CPMK yang relevan, sistem menampilkan halaman Visualisasi CPMK per Angkatan. Pengguna kemudian memilih opsi "Print PDF," yang menyebabkan sistem menghasilkan laporan dalam format PDF.

22. Melihat Visualisasi CPMK per Mata Kuliah (ACT-022)

Activity Diagram pada Gambar 3.25 menjelaskan alur proses melihat visualisasi Capaian Pembelajaran Mata Kuliah (CPMK) per mata kuliah oleh Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, dan Penjamin Mutu dalam sistem. Proses dimulai ketika pengguna memilih menu CPMK per Mata Kuliah, kemudian sistem menampilkan *form* Visualisasi CPMK per Mata Kuliah. Pengguna mengisi *form* tersebut dengan data yang diperlukan, lalu sistem melakukan *request* data ke *database*. *Database* mengambil data yang relevan dan mengirimkannya kembali ke sistem. Setelah data diterima, sistem menampilkan halaman Visualisasi CPL per Angkatan beserta *form* Capaian Pembelajaran Mata Kuliah (CPMK) per Mahasiswa.



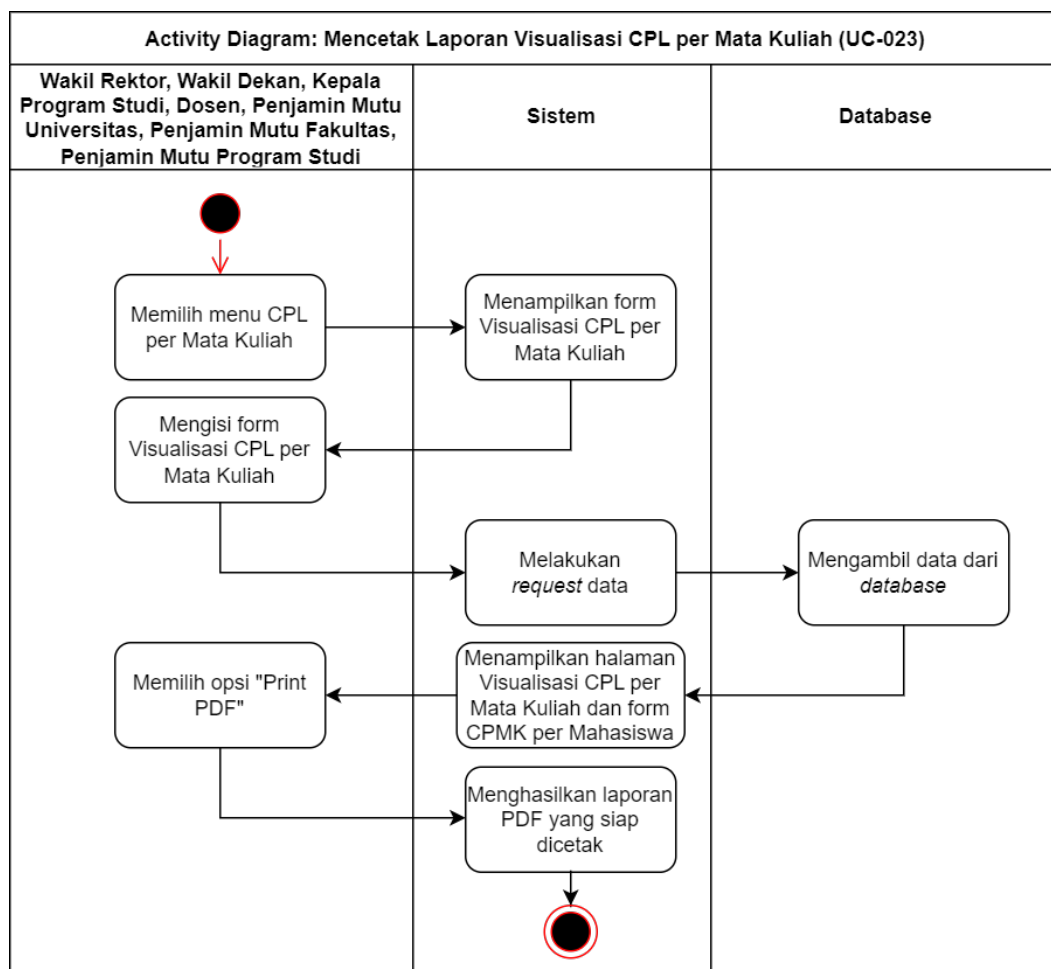
Gambar 3.24 *Activity Diagram* Mencetak Laporan Visualisasi CPMK per Angkatan.



Gambar 3.25 *Activity Diagram* Melihat Visualisasi CPMK per Mata Kuliah.

23. Mencetak Laporan Visualisasi CPMK per Mata Kuliah (ACT-023)

Activity Diagram pada Gambar 3.26 menjelaskan alur proses pencetakan laporan visualisasi Capaian Pembelajaran Mata Kuliah (CPMK) per mata kuliah oleh Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, dan Penjamin Mutu dalam sistem. Proses dimulai ketika pengguna memilih menu CPMK per Mata Kuliah, kemudian sistem menampilkan *form* Visualisasi CPMK per Mata Kuliah. Pengguna mengisi *form* tersebut dengan data yang diperlukan, lalu sistem melakukan *request* data ke *database*. *Database* mengambil data yang relevan dan mengirimkannya kembali ke sistem. Setelah data diterima, sistem menampilkan halaman Visualisasi CPMK per Mata Kuliah beserta *form* Capaian Pembelajaran Mata Kuliah (CPMK) per Mahasiswa. Pengguna kemudian memilih opsi "Print PDF," yang menyebabkan sistem menghasilkan laporan dalam format PDF.

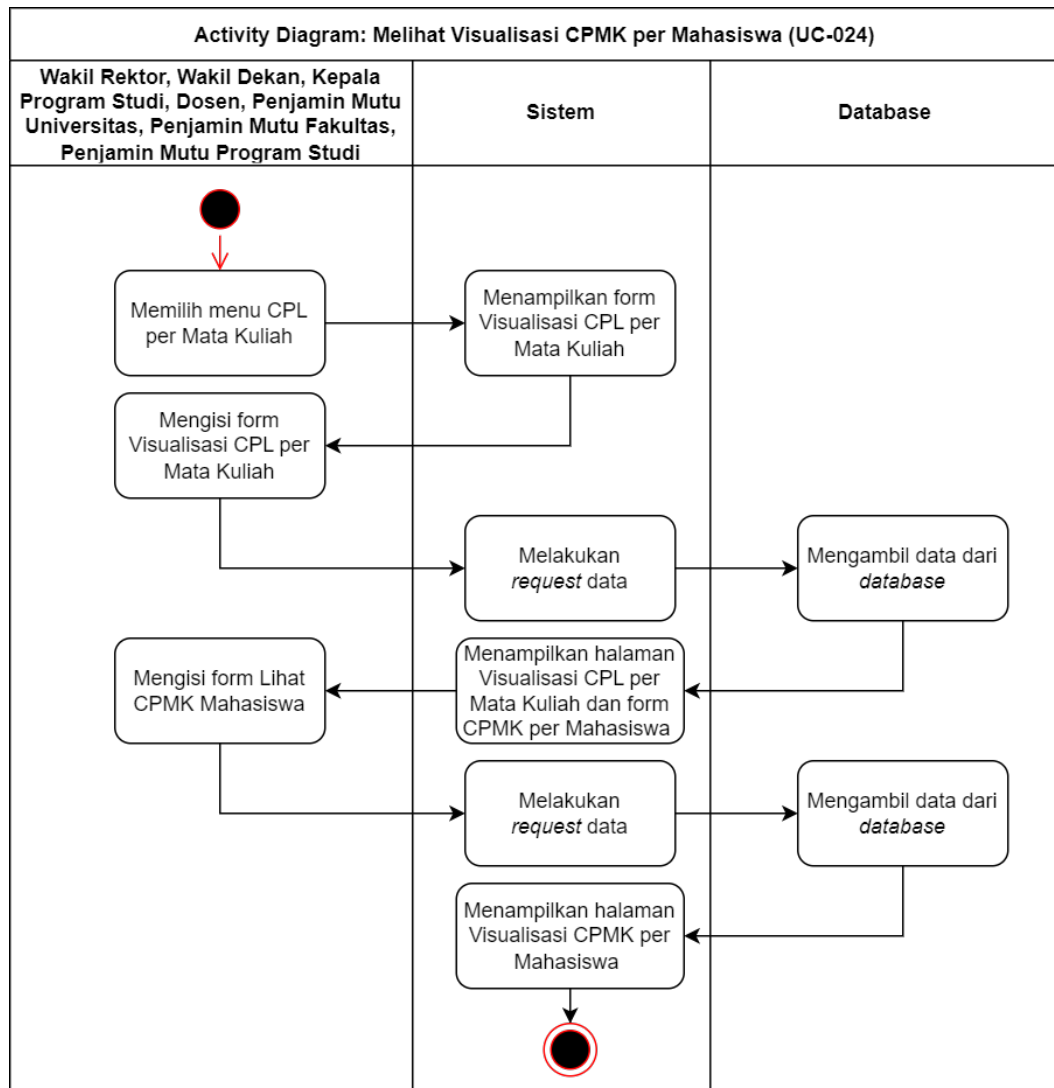


Gambar 3.26 *Activity Diagram* Mencetak Laporan Visualisasi CPMK per Mata Kuliah.

24. Melihat Visualisasi CPMK per Mahasiswa (ACT-024)

Activity Diagram pada Gambar 3.27 menjelaskan alur proses melihat visualisasi Capaian Pembelajaran Mata Kuliah (CPMK) per mahasiswa oleh Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, dan Penjamin Mutu dalam sistem. Proses dimulai ketika pengguna memilih menu CPMK per Mata Kuliah, kemudian sistem menampilkan *form* Visualisasi CPMK per Mata Kuliah. Pengguna mengisi form tersebut dengan data yang diperlukan, lalu sistem melakukan *request* data ke *database*. *Database* mengambil data yang relevan dan mengirimkannya kembali ke sistem. Setelah data diterima, sistem menampilkan halaman Visualisasi CPMK per Mata Kuliah beserta *form* Lihat CPMK Mahasiswa. Pengguna kemudian mengisi *form* Lihat CPMK

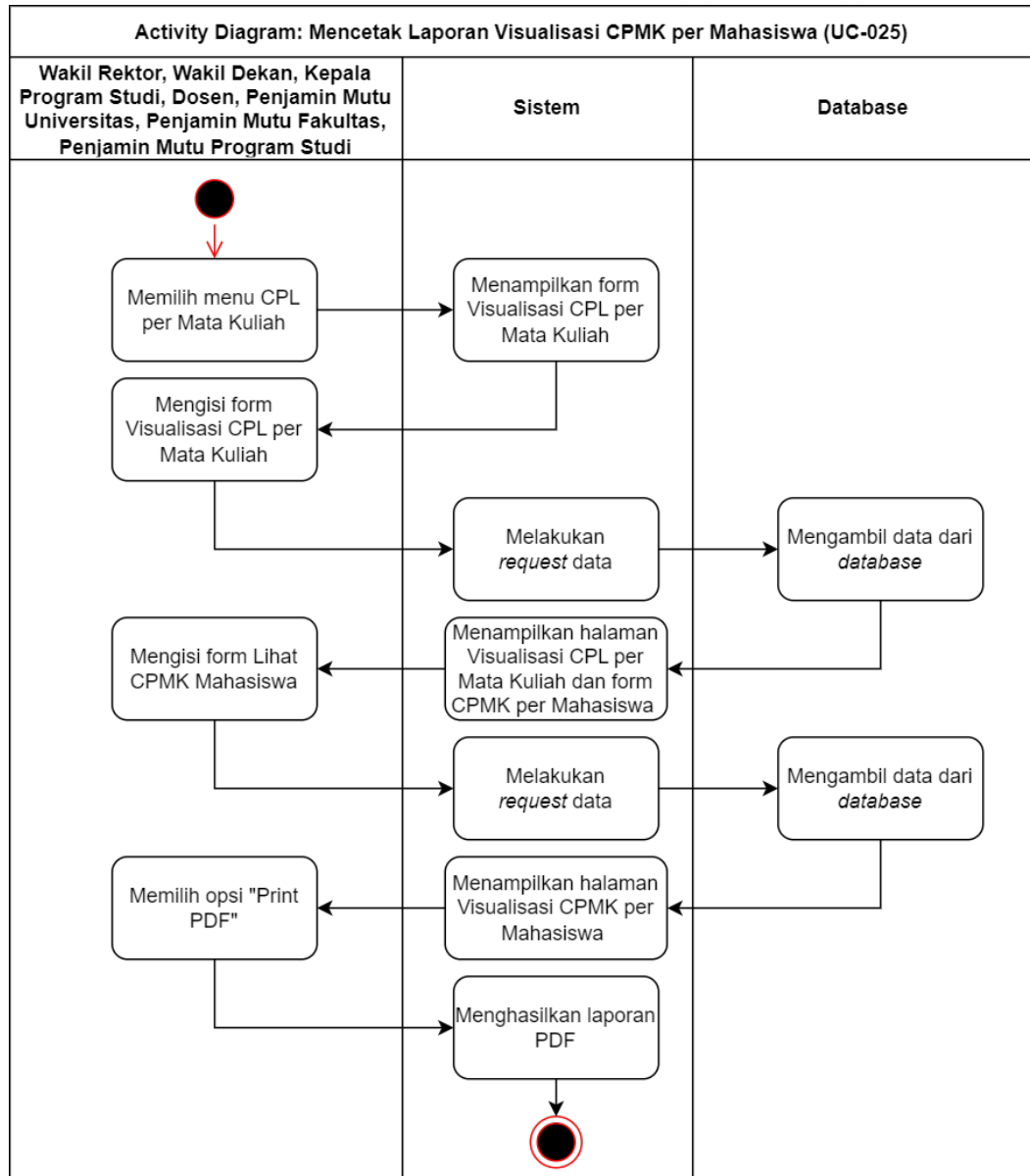
Mahasiswa, lalu sistem kembali melakukan *request* data ke *database*. Setelah *database* mengambil dan mengirimkan data CPMK yang relevan, sistem menampilkan halaman Visualisasi CPMK per Mahasiswa.



Gambar 3.27 *Activity Diagram* melihat data CPMK per Angkatan.

25. Mencetak Laporan Visualisasi CPMK per Mahasiswa (ACT-025)

Activity Diagram pada Gambar 3.28 menjelaskan alur proses pencetakan laporan visualisasi Capaian Pembelajaran Mata Kuliah (CPMK) per mahasiswa oleh Wakil Rektor, Wakil Dekan, Kepala Program Studi, Dosen, dan Penjamin Mutu dalam sistem. Proses dimulai ketika pengguna memilih menu CPMK per Mata Kuliah, kemudian sistem menampilkan *form* Visualisasi CPMK per Mata Kuliah. Pengguna mengisi *form* tersebut dengan data yang diperlukan, lalu sistem melakukan *request* data ke *database*. *Database* mengambil data yang relevan dan mengirimkannya kembali ke sistem. Setelah data diterima, sistem menampilkan halaman Visualisasi CPMK per Mata Kuliah beserta *form* Lihat CPMK Mahasiswa. Pengguna kemudian mengisi *form* Lihat CPMK Mahasiswa, lalu sistem kembali melakukan *request* data ke *database*. Setelah *database* mengambil dan mengirimkan data CPMK yang relevan, sistem menampilkan halaman Visualisasi CPMK per Mahasiswa. Pengguna kemudian memilih opsi "Print PDF," yang menyebabkan sistem menghasilkan laporan dalam format PDF.



Gambar 3.28 *Activity Diagram* Mencetak Laporan Visualisasi CPMK per Mahasiswa.

3.5.14 *Class Diagram*

Class Diagram merupakan salah satu diagram struktur pada *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan kelas-kelas, atribut, metode (operations), dan hubungan antar kelas dalam sebuah sistem perangkat lunak. Diagram ini memberikan pandangan statis mengenai struktur sistem dengan menunjukkan komponen-komponen utama sistem dan bagaimana mereka saling terkait. Secara umum, sebuah *class diagram* terdiri dari representasi visual kelas sebagai sebuah kotak yang terbagi menjadi tiga bagian: nama kelas, atribut kelas, dan metode kelas. Hubungan antar kelas seperti asosiasi (association), agregasi (aggregation), komposisi (composition), dan pewarisan (inheritance atau generalisasi) juga digambarkan untuk menunjukkan bagaimana kelas-kelas tersebut berinteraksi dan saling bergantung.

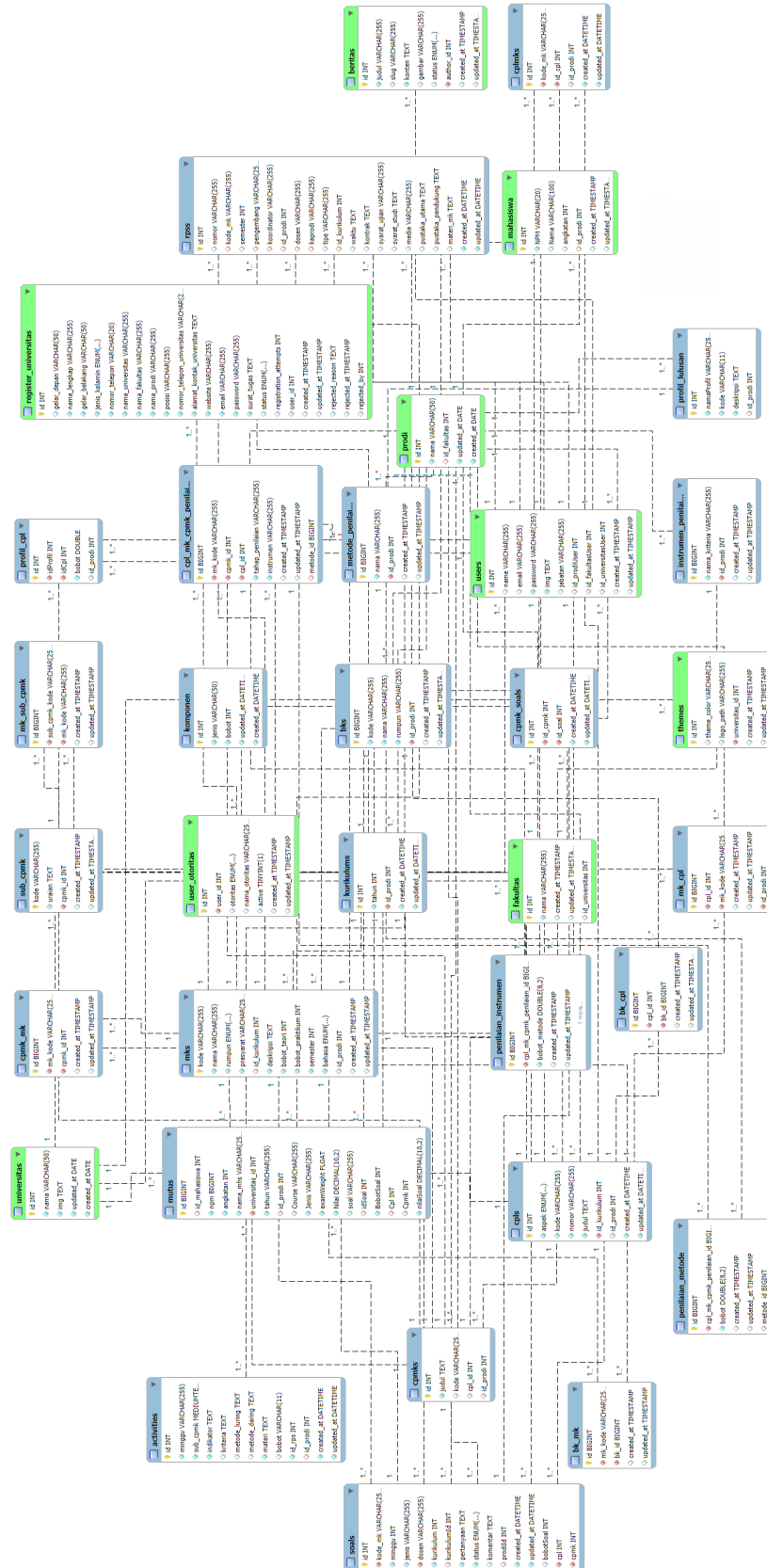
Model fungsional dalam penelitian ini juga memanfaatkan *Class Diagram* untuk merepresentasikan struktur kelas serta hubungan antar kelas dalam sistem penjaminan mutu yang dikembangkan secara lebih rinci. *Class Diagram* yang disajikan menggambarkan atribut-atribut penting dan metode utama dari setiap kelas yang terlibat dalam modul yang dikembangkan. Selain itu, diagram ini juga secara khusus menyoroti modul yang menjadi fokus pengembangan, yaitu Manajemen *User* dan *File Laporan*. Modul ini ditandai dengan warna hijau muda dalam diagram untuk membedakan bagian yang dikembangkan oleh peneliti dari bagian lain dalam sistem yang mungkin telah ada sebelumnya atau dikembangkan oleh pihak lain. Penandaan ini bertujuan untuk memperjelas kontribusi spesifik peneliti dalam pengembangan sistem secara keseluruhan serta memudahkan pembaca dalam memahami cakupan implementasi yang dilakukan. *Class Diagram* lengkap dari Pengembangan Sistem Penjamin Mutu Jurusan Ilmu Komputer Modul Manajemen *User* dan *File Laporan*, yang mencakup detail kelas-kelas yang dikembangkan beserta relasinya, dapat dilihat pada Gambar 3.28.

Gambar 3.29 *Class Diagram* Pengembangan Sistem Penjamin Mutu Jurusan Ilmu Komputer Modul Manajemen *User* dan *File* Laporan.

3.5.15 *Entity Relationship Diagram (ERD)*

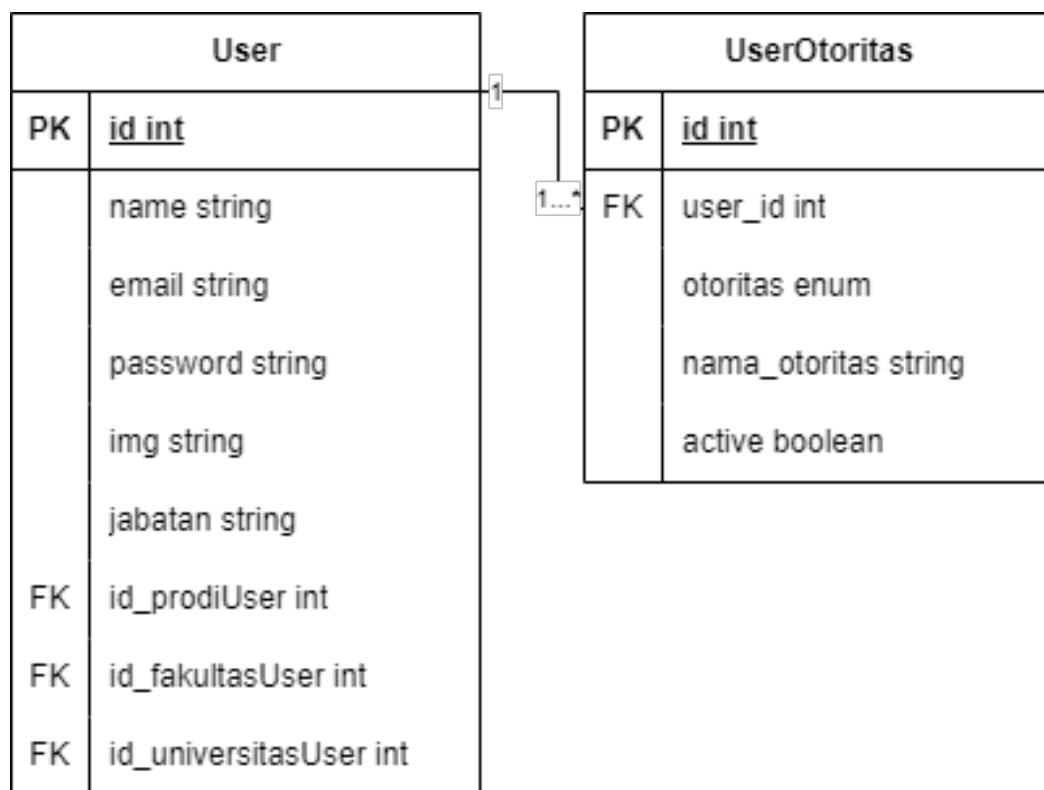
Entity Relationship Diagram (ERD) merupakan sebuah model data konseptual yang esensial dalam perancangan basis data, berfungsi untuk merepresentasikan struktur logis dari data yang akan dikelola. Diagram ini secara visual menggambarkan entitas-entitas utama, yaitu objek atau konsep yang relevan dalam sistem dan datanya perlu disimpan, yang umumnya direpresentasikan sebagai bentuk persegi panjang. Setiap entitas memiliki atribut-atribut, yang merupakan properti atau karakteristik yang mendeskripsikan entitas tersebut; atribut ini sering kali digambarkan sebagai oval yang terhubung ke entitasnya atau dicantumkan langsung di dalam kotak entitas pada beberapa notasi modern. Lebih lanjut, ERD memetakan relasi atau hubungan antar entitas, yang digambarkan sebagai garis penghubung dan sering kali diberi nama untuk menjelaskan sifat hubungan tersebut, seperti "memiliki" atau "mengelola". Aspek krusial dari relasi adalah kardinalitasnya, yang menunjukkan jumlah instans entitas yang dapat berpartisipasi dalam suatu hubungan, contohnya meliputi relasi *one-to-one*, *one-to-many*, atau *many-to-many*. Tujuan utama dari pembuatan ERD adalah untuk memodelkan data beserta interkoneksinya secara jelas, sehingga dapat menjadi landasan yang kokoh dalam perancangan skema basis data fisik yang efektif dan efisien.

Dalam konteks studi ini, *Entity Relationship Diagram (ERD)* digunakan untuk memvisualisasikan relasi antara data dan berbagai entitas secara lebih efektif, khususnya dalam menggambarkan struktur basis data serta hubungan antar entitas yang membentuk Sistem Penjamin Mutu Jurusan Ilmu Komputer. Diagram ini sangat membantu dalam memahami bagaimana berbagai elemen data saling terhubung dan mengalir di dalam sistem, serta berfungsi sebagai acuan penting selama proses perancangan dan implementasi basis data aktual. ERD yang disajikan memberikan gambaran menyeluruh mengenai skema data yang secara spesifik mendasari pengembangan Modul Manajemen *User* dan *File* Laporan. ERD lengkap yang menunjukkan detail entitas dan relasi dalam modul yang dikembangkan ini dapat dilihat pada Gambar 3.30.



3.5.16 Arsitektur Manajemen Peran dan Keamanan

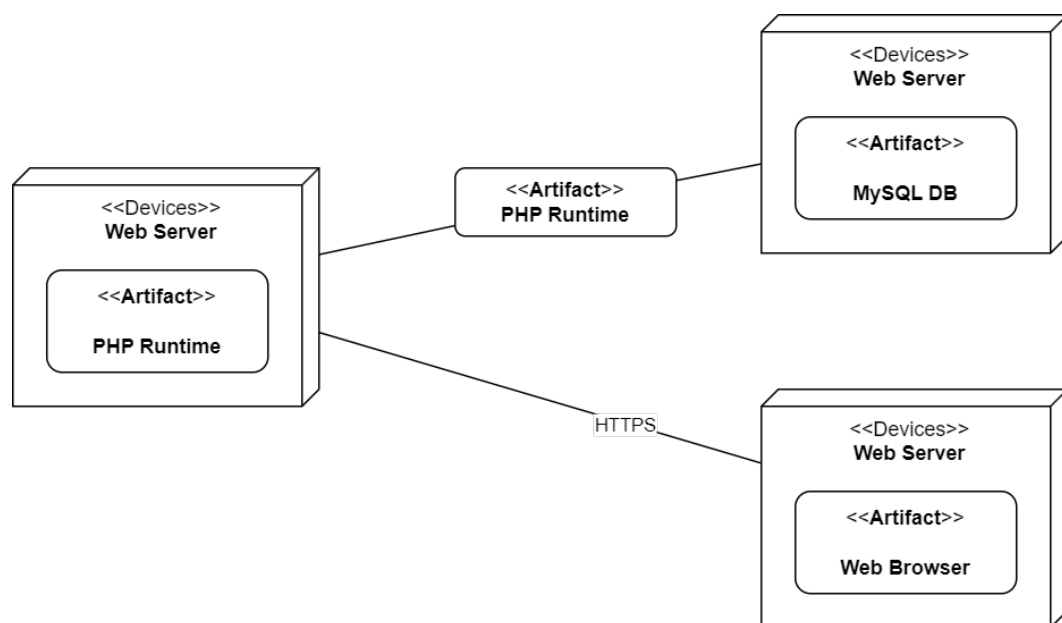
Implementasi *Role-Based Access Control* (RBAC) pada sistem ini direalisasikan melalui struktur *database* dengan tiga entitas utama yaitu “users”, “otoritas”, dan “user_otoritas”. Tabel “users” menyimpan data dasar pengguna. Tabel otoritas berfungsi sebagai master data untuk semua peran yang ada di sistem. Hubungan *many-to-many* antara pengguna dan peran dijembatani oleh tabel pivot “user_otoritas”, yang memungkinkan seorang pengguna untuk memiliki lebih dari satu peran. Tabel ini juga memiliki kolom *active* untuk menentukan peran mana yang sedang aktif digunakan oleh pengguna pada satu sesi, yang mendukung fungsionalitas multi-otoritas secara dinamis.



Gambar 3.31 Arsitektur Basis Data untuk Manajemen Peran (RBAC).

3.5.17 Deployment Diagram

Sebagaimana telah dijelaskan pada Bab 2.7.3, *Deployment Diagram* digunakan untuk memodelkan arsitektur fisik sistem perangkat lunak. Arsitektur implementasi fisik untuk Sistem Penjamin Mutu Jurusan Ilmu Komputer Modul Manajemen *User* dan *File*. Diagram tersebut mengilustrasikan *node-node* utama yang terlibat, yaitu *Node* Server Aplikasi tempat aplikasi berbasis Laravel dijalankan, *Node* Server Basis Data untuk penyimpanan MySQL, dan *Node* Klien yang mengakses sistem melalui web browser. Jalur komunikasi antar *node* juga ditunjukkan, termasuk koneksi HTTPS antara klien dan server aplikasi, serta interaksi antara server aplikasi dan server basis data yang difasilitasi oleh *PHP Runtime*.



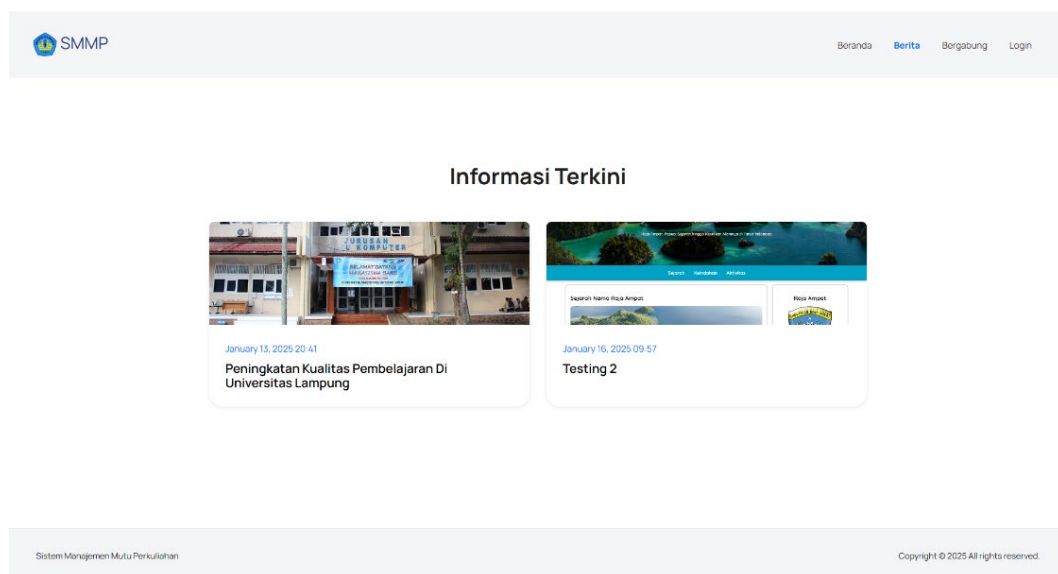
Gambar 3.32 *Deployment Diagram* Sistem Manajemen *User* dan *File* Laporan.

3.5.18 Rancangan Tampilan Antarmuka Pengguna

Rancangan tampilan antarmuka pengguna dalam sistem ini dibuat dengan pendekatan *live prototyping*, di mana setiap elemen antarmuka langsung diterapkan pada sistem yang sedang dikembangkan untuk mendapatkan umpan balik secara langsung dari pengguna. Pendekatan ini memungkinkan iterasi desain yang lebih cepat dan fleksibel, serta mempermudah validasi terhadap kebutuhan pengguna. Pada subbab ini ditampilkan beberapa contoh tampilan antarmuka berdasarkan masing-masing fitur utama sistem.

1. Melihat Berita (UC-001/ACT-001)

Tampilan pada Gambar 3.33 merupakan rancangan antarmuka untuk melihat berita pada halaman *landing page*. Halaman ini menampilkan daftar berita terkini dalam bentuk kartu yang berisi gambar, tanggal publikasi, dan judul berita.



Gambar 3.33 Rancangan tampilan Melihat Berita.

2. Menambahkan Registrasi Universitas (UC-002/ACT-002)

Tampilan pada Gambar 3.34 merupakan rancangan antarmuka untuk menambahkan registrasi universitas dalam sistem. *Form* ini mencakup empat bagian utama, yaitu data PIC Prodi, informasi universitas, data kredensial, dan surat tugas. Pada bagian data PIC Prodi, pengguna diminta mengisi gelar depan, nama lengkap, gelar belakang, jenis kelamin, dan nomor telepon. Selanjutnya, bagian informasi universitas mencakup nama universitas, nama fakultas, nama program studi, posisi PIC, nomor telepon universitas, alamat kontak, dan website. Bagian data kredensial digunakan untuk membuat akun *login* berupa email, *password*, dan konfirmasi *password*. Terakhir, bagian surat tugas memungkinkan pengguna mengunggah dokumen surat tugas dalam format PDF sebelum menekan tombol Register untuk menyelesaikan proses pendaftaran.

The image shows a web form titled "Register Form". It is organized into four columns and rows of sections:

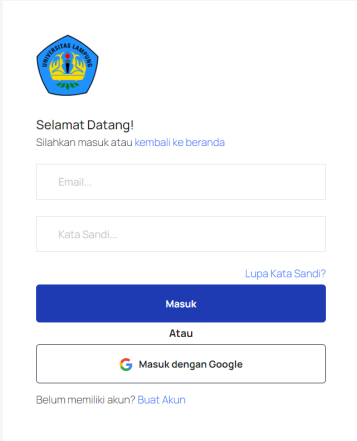
- Data PIC Prodi:** Includes fields for "Gelar Depan", "Nama Lengkap", "Gelar Belakang", "Jenis Kelamin" (with radio buttons for Pria and Wanita), and "Nomor Telepon".
- Informasi Universitas:** Includes fields for "Nama Universitas", "Nama Fakultas", "Nama prodi", "Posisi", "Nomor Telepon Universitas", "Alamat Kontak Universitas", and "Website".
- Data Kredensial:** Includes fields for "Email", "Password", and "Konfirmasi Password".
- Surat Tugas:** Includes a "Surat Tugas (PDF)" field with a "Choose File" button and a "No file chosen" status. Below it, it says "Maksimal ukuran file: 2MB".

A blue "Register" button is positioned at the bottom right of the form area.

Gambar 3.34 Rancangan tampilan Menambahkan Registrasi Universitas.

3. Melakukan *Login* (UC-003/ACT-003)

Tampilan pada Gambar 3.35 merupakan rancangan antarmuka untuk melakukan login ke dalam sistem. Halaman ini menampilkan formulir login yang terdiri dari pesan sambutan “Selamat Datang!”, tautan kembali ke halaman beranda, input email, input *password*, serta tombol *Login*. Selain metode autentikasi konvensional ini, rancangan antarmuka juga menyertakan tombol ‘Masuk dengan Google’. Fitur ini dirancang sebagai jalur autentikasi alternatif yang aman dan efisien, yang memungkinkan pengguna untuk masuk ke sistem menggunakan akun Google mereka, dengan syarat utama bahwa email yang terasosiasi dengan akun Google tersebut telah terdaftar sebelumnya di dalam sistem. Di bagian bawah formulir, terdapat juga tautan yang mengarah ke halaman registrasi universitas bagi pengguna yang belum memiliki akun pada sistem.

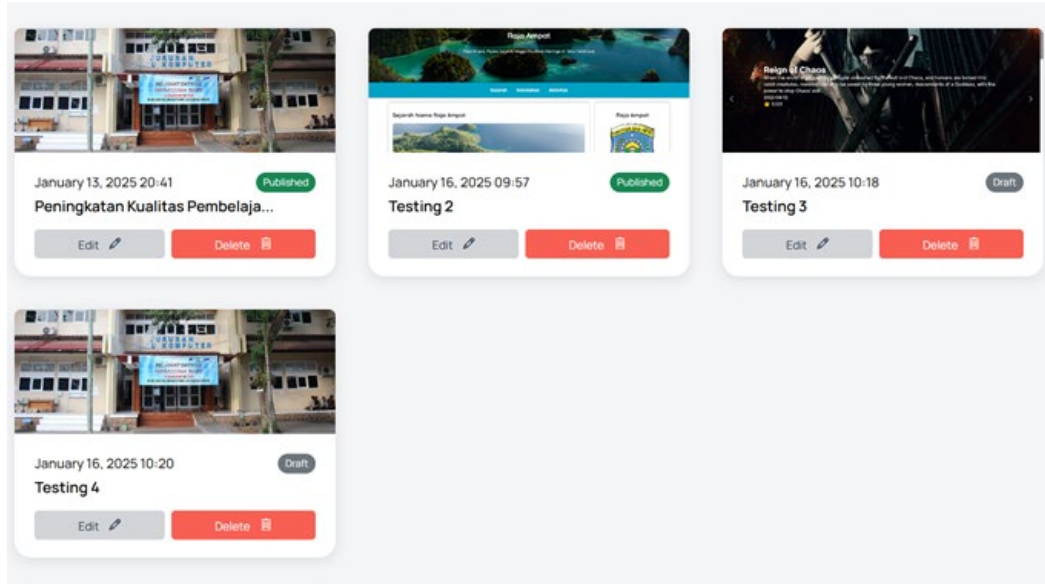


Gambar 3.35 Rancangan tampilan Melakukan *Login*.

4. Mengelola Berita (UC-004/ACT-004)

Tampilan pada Gambar 3.36, Gambar 3.37, dan Gambar 3.38 merupakan rancangan antarmuka untuk mengelola berita dalam sistem. Terdapat tiga halaman utama yang ditampilkan, yaitu halaman untuk melihat daftar berita yang telah ditambahkan, halaman *form* untuk menambahkan berita dengan

mengisi judul, konten, dan gambar, serta halaman *form* untuk mengubah berita yang memungkinkan pengguna mengedit judul, konten, status berita (*published* atau *draft*), dan mengganti gambar.

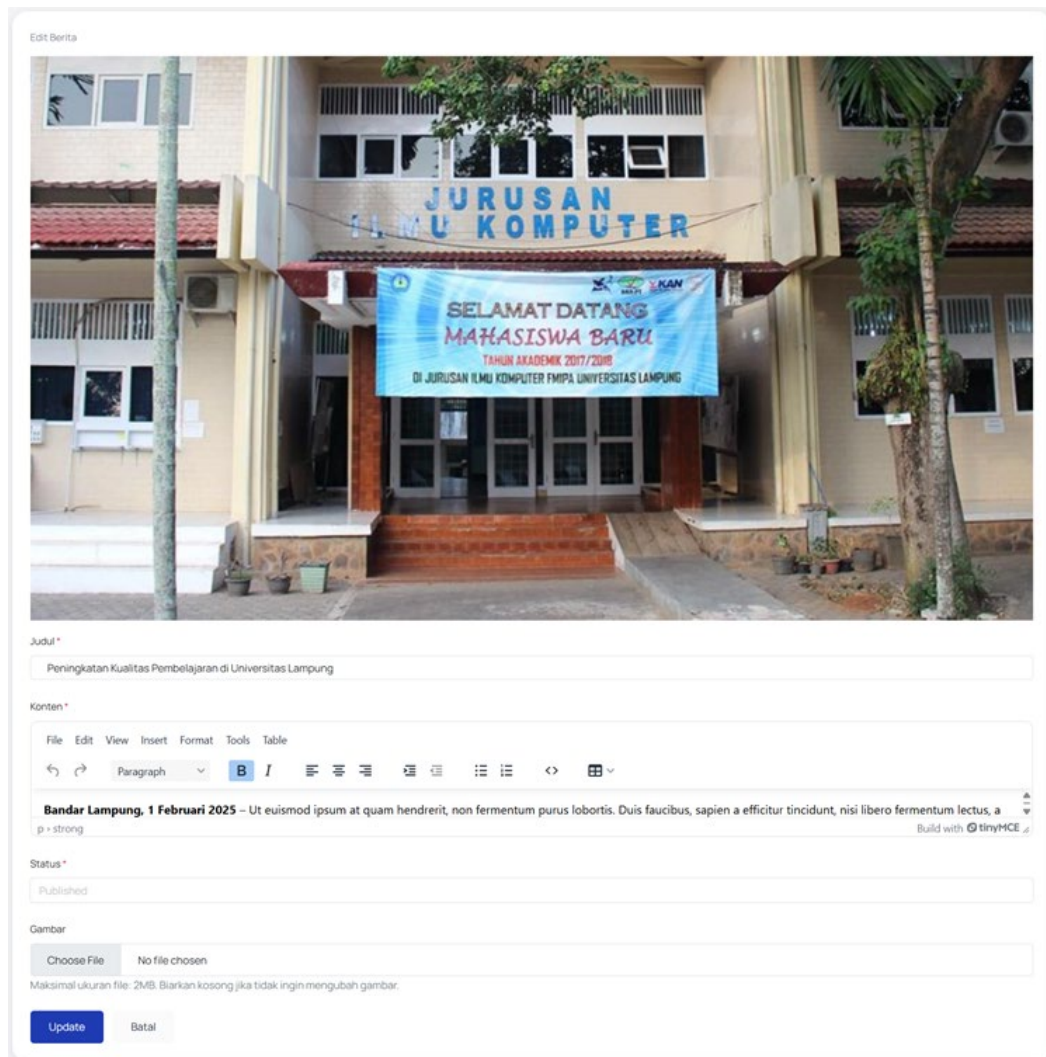


Gambar 3.36 Rancangan tampilan Mengelola Berita bagian Melihat Berita.

The screenshot shows a form titled 'Add Berita'. It contains the following elements:

- Judul *:** A text input field with the placeholder 'Masukkan judul berita'.
- Konten *:** A rich text editor with a toolbar containing options like 'File', 'Edit', 'View', 'Insert', 'Format', 'Tools', and 'Table'. The editor area has the placeholder 'Masukkan konten'.
- Gambar *:** A section for uploading an image. It includes a 'Choose File' button, a 'No file chosen' status, and a file size limit of 'Maksimal ukuran file: 2MB'.
- Submit:** A blue button at the bottom of the form.

Gambar 3.37 Rancangan tampilan Mengelola Berita bagian *form* Tambah Berita.



Edit Berita

JURUSAN
ILMU KOMPUTER

SELAMAT DATANG
MAHASISWA BARU
TAHUN AKADEMIK 2017/2018
DI JURUSAN ILMU KOMPUTER FMIPA UNIVERSITAS LAMPUNG

Judul *

Peningkatan Kualitas Pembelajaran di Universitas Lampung

Konten *

File Edit View Insert Format Tools Table

Paragraph B I

Bandar Lampung, 1 Februari 2025 – Ut euismod ipsum at quam hendrerit, non fermentum purus lobortis. Duis faucibus, sapien a efficitur tincidunt, nisi libero fermentum lectus, a p + strong

Status *

Published

Gambar

Choose File No file chosen

Maksimal ukuran file: 2MB. Biarkan kosong jika tidak ingin mengubah gambar.

Update Batal

Gambar 3.38 Rancangan tampilan Mengelola Berita bagian *form* Ubah Berita.

5. Mengelola Registrasi Universitas (UC-005/ACT-005)

Tampilan pada Gambar 3.39 dan Gambar 3.40, dan Gambar 3.41 merupakan rancangan antarmuka untuk mengelola registrasi universitas. Rancangan ini mencakup halaman yang menampilkan daftar registrasi universitas yang telah masuk ke sistem, serta halaman detail registrasi dari masing-masing universitas. Pada halaman detail, terdapat dua tombol aksi yaitu *Approve* yang disertai dengan dialog konfirmasi, dan *Reject* yang akan menampilkan *form* untuk mengisi alasan penolakan.

List Registrasi Universitas

Show entries

Search:

#	Nama Universitas	Nama Kontak PIC	Email	Nomor Telepon	Surat Tugas	Status	Detail
1	Universitas Testing	Universitas Testing, S.Kom	ramadhanraziq@gmail.com	0812345678	Preview PDF	Approved	Detail

Showing 1 to 1 of 1 entries

Previous 1 Next

Gambar 3.39 Rancangan tampilan Mengelola Registrasi Universitas bagian Melihat Daftar Registrasi Universitas.

Konfirmasi Persetujuan ✕

Apakah Anda yakin ingin menyetujui pendaftaran universitas ini?

[Batal](#) [Ya, Setujui](#)

Gambar 3.40 Rancangan tampilan Mengelola Registrasi Universitas bagian Menyetujui Registrasi Universitas

Tolak Pendaftaran Universitas ✕

Alasan Penolakan

[Batal](#) [Tolak](#)

Gambar 3.41 Rancangan tampilan Mengelola Registrasi Universitas bagian Menolak Registrasi Universitas.

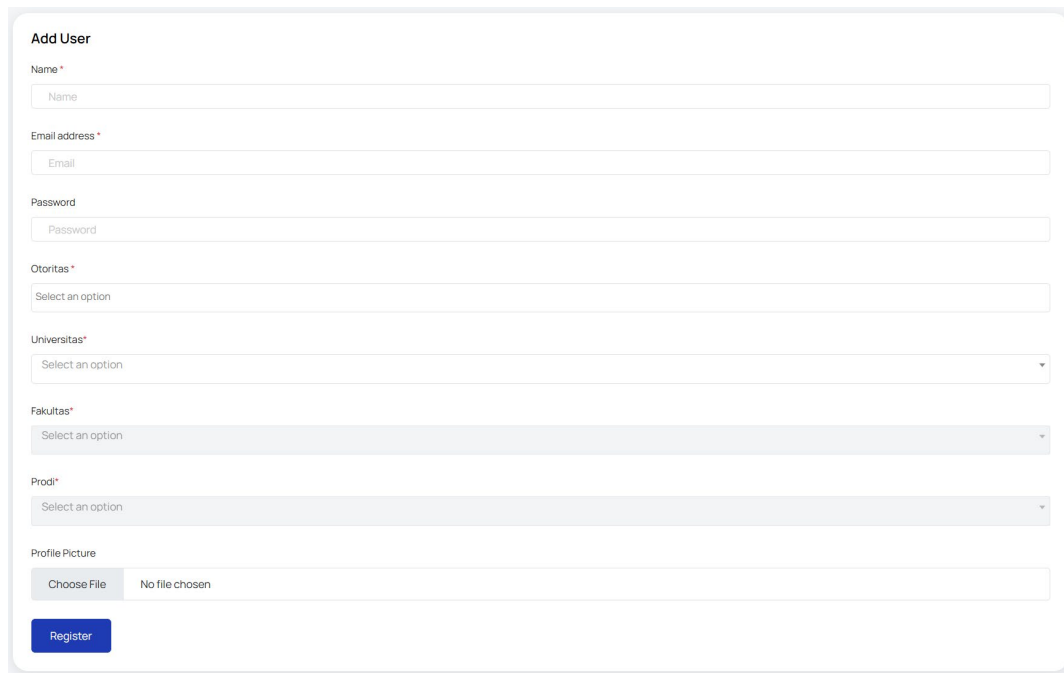
6. Mengelola *User* (UC-006/ACT-006)

Tampilan pada Gambar 3.42, Gambar 3.43, dan Gambar 3.44 merupakan rancangan antarmuka untuk mengelola *user* dalam sistem. Terdapat tiga halaman utama yang ditampilkan, yaitu halaman untuk melihat daftar *user* yang sudah terdaftar, halaman *form* untuk menambahkan *user* baru dengan mengisi nama, email, *password*, memilih otoritas yang mendukung multitoritas, serta *dropdown* untuk memilih universitas, fakultas, dan program studi. Selain itu, terdapat juga halaman *form* untuk mengubah data *user* dengan elemen *form* yang sama.

The screenshot displays the 'List User' interface. At the top, there are two dropdown menus for 'Fakultas' (Pilih Fakultas) and 'Program Studi' (Pilih Program Studi). Below these are 'Filter' and 'Reset' buttons. A 'Show' dropdown is set to '10 entries', and a 'Search' input field is on the right. The main content is a table with columns: #, Name, Email, Otoritas, Program Studi, and Fakultas. The table lists 10 users. At the bottom, it shows 'Showing 1 to 10 of 46 entries' and a pagination bar with 'Previous', '1', '2', '3', '4', '5', and 'Next' buttons.

#	Name	Email	Otoritas	Program Studi	Fakultas
1	multi otoritas	multitoritas@gmail.com	Wakil Dekan, Kepala Program Studi, Dosen, Penjamin Mutu Program Studi, Penjamin Mutu Fakultas, Penjamin Mutu Universitas, Wakil Rektor	S1 - Ilmu Komputer	Fakultas Matematika dan Ilmu Pengetahuan
2	Anggie Tamara	anggie.t17210@gmail.com	Dosen	S1 - Ilmu Komputer	Fakultas Matematika dan Ilmu Pengetahuan
3	Romdoni	acron@gmail.com	Dosen	S1 - Ilmu Komputer	Fakultas Matematika dan Ilmu Pengetahuan
4	Rico Andrian, S.Si., M.Kom	kangrico@gmail.com	Dosen	S1 - Ilmu Komputer	Fakultas Matematika dan Ilmu Pengetahuan
5	Aristoteles, S.Si., M.Si	aristoteles1981@fmipa.unila.ac.id	Dosen	S1 - Ilmu Komputer	Fakultas Matematika dan Ilmu Pengetahuan
6	Rizky Prabowo, M.Kom	rizky.prabowo@fmipa.unila.ac.id	Dosen	S1 - Ilmu Komputer	Fakultas Matematika dan Ilmu Pengetahuan
7	Tristiyanto, S.Kom., M.I.S., Ph.D	tristiyanto1981@fmipa.unila.ac.id	Dosen	S1 - Ilmu Komputer	Fakultas Matematika dan Ilmu Pengetahuan
8	Yunda Heningtyas, M.Kom	yunda.heningtyas@fmipa.unila.ac.id	Dosen	S1 - Ilmu Komputer	Fakultas Matematika dan Ilmu Pengetahuan
9	Favorisen R. Lumbanraja, Ph.D	favorisen.lumbanraja@fmipa.unila.ac.id	Dosen	S1 - Ilmu Komputer	Fakultas Matematika dan Ilmu Pengetahuan
10	Muhaqiqin, S.Kom., M.T.I.	muhaqiqin@fmipa.unila.ac.id	Dosen	S1 - Ilmu Komputer	Fakultas Matematika dan Ilmu Pengetahuan

Gambar 3.42 Rancangan tampilan Mengelola *User* bagian Melihat *User*.



Add User

Name *

Email address *

Password

Otoritas *

Universitas *

Fakultas *

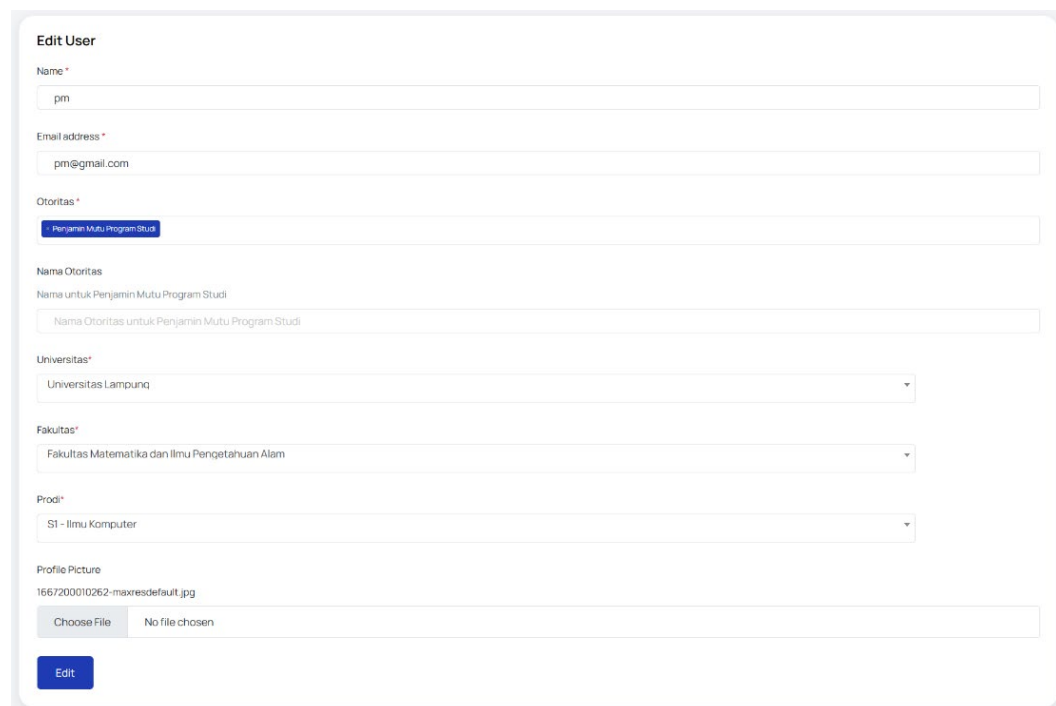
Prodi *

Profile Picture

Choose File No file chosen

Register

Gambar 3.43 Rancangan tampilan Mengelola *User* bagian *form* Tambah *User*.



Edit User

Name *

Email address *

Otoritas *

Nama Otoritas

Universitas *

Fakultas *

Prodi *

Profile Picture

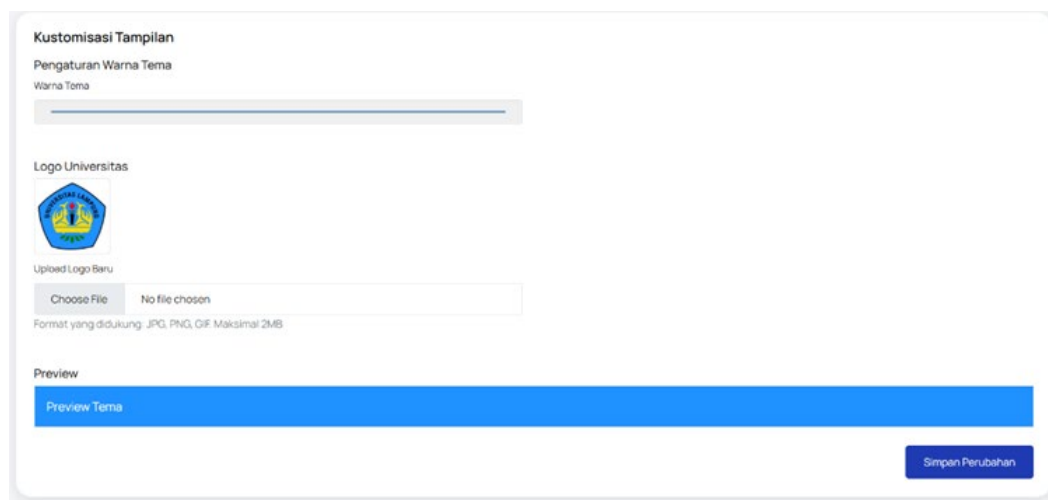
Choose File No file chosen

Edit

Gambar 3.44 Rancangan tampilan Mengelola *User* bagian *form* Ubah *User*.

7. Mengelola Tema Universitas (UC-007/ACT-007)

Tampilan pada Gambar 3.45 merupakan rancangan antarmuka untuk mengelola tema universitas. Halaman ini menyediakan *form* yang memungkinkan admin universitas melakukan kustomisasi tampilan dengan mengatur warna tema pada bagian *header* dan *footer*, serta mengunggah logo universitas yang baru. Tersedia pula fitur *preview* untuk melihat hasil perubahan tema sebelum disimpan.



The screenshot shows a web interface titled "Kustomisasi Tampilan" (Customize Appearance). It includes a section for "Pengaturan Warna Tema" (Theme Color Settings) with a "Warna Tema" (Theme Color) input field. Below this is the "Logo Universitas" (University Logo) section, which displays a current logo and an "Upload Logo Baru" (Upload New Logo) button. A file selection area shows "Choose File" and "No file chosen". A note specifies supported formats: "Format yang didukung: .JPG, .PNG, .GIF. Maksimal 2MB". At the bottom, there is a "Preview" section with a "Preview Tema" button and a "Simpan Perubahan" (Save Changes) button.

Gambar 3.45 Rancangan tampilan Mengelola Tema Universitas.

8. Mengelola Profil (UC-008/ACT-008)

Tampilan pada Gambar 3.46 merupakan rancangan antarmuka untuk mengelola profil pengguna. Halaman ini menampilkan informasi pengguna seperti nama, email, dan otoritas yang sedang aktif, serta menyediakan *form* untuk mengubah *password*. Bagi pengguna yang memiliki lebih dari satu otoritas, tersedia fitur untuk memilih dan menyimpan otoritas yang sedang digunakan.

The interface consists of three main sections:

- Profile Card:** Displays a circular profile picture placeholder, a name field with the value 'multi otoritas', an email field with 'multitoritas@gmail.com', and an authority field with 'Penjamin Mutu Program Studi'.
- Change Password:** A form with three input fields: 'Old Password', 'New Password', and 'Password Confirmation', followed by a 'Submit' button.
- Pilih Otoritas:** A dropdown menu currently showing 'Penjamin Mutu Program Studi' and a 'Simpan Otoritas' button.

Gambar 3.46 Rancangan tampilan Mengelola Profil.

9. Mengelola Prodi (UC-009/ACT-009)

Tampilan pada Gambar 3.47 dan Gambar 3.48 merupakan rancangan antarmuka untuk Mengelola Prodi. Tampilan ini menampilkan dua halaman yaitu halaman untuk melihat daftar program studi yang telah terdaftar di dalam sistem, serta halaman *form* untuk menambahkan program studi baru dengan mengisi *dropdown* nama universitas, *dropdown* nama fakultas, dan input nama prodi.

List Prodi

Universitas: Fakultas: Filter Reset

Show entries

Search:

#	Prodi	Fakultas	Universitas	Action
1	D3 - Manajemen Informatika	Fakultas Matematika dan Ilmu Pengetahuan Alam	Universitas Lampung	Delete
2	S1 - Ilmu Komputer	Fakultas Matematika dan Ilmu Pengetahuan Alam	Universitas Lampung	Delete
3	S1 - Sistem Informasi	Fakultas Matematika dan Ilmu Pengetahuan Alam	Universitas Lampung	Delete
4	S1 - Ilmu Komputer PSDKU	Fakultas Matematika dan Ilmu Pengetahuan Alam	Universitas Lampung	Delete
5	S2 - Ilmu Komputer	Fakultas Matematika dan Ilmu Pengetahuan Alam	Universitas Lampung	Delete
6	Prodi Testing	Fakultas Testing	Universitas Testing	Delete

Showing 1 to 6 of 6 entries

Previous 1 Next

Gambar 3.47 Rancangan tampilan Mengelola Prodi bagian Melihat Prodi.

Gambar 3.48 Rancangan tampilan Mengelola Prodi bagian *form* Tambah Prodi.

10. Mengelola Fakultas (UC-010/ACT-010)

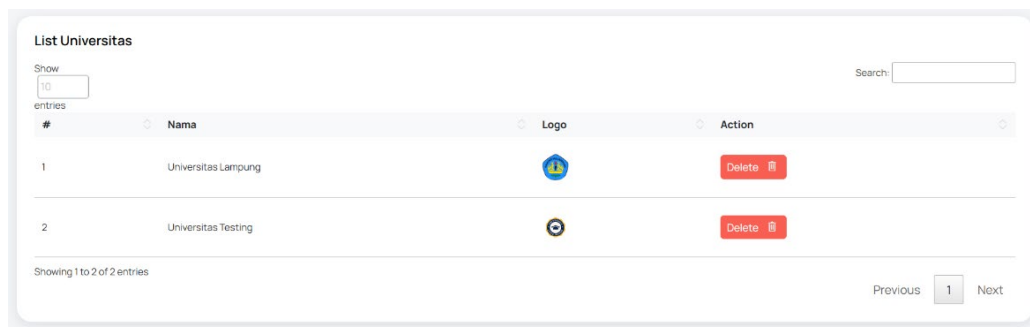
Tampilan pada Gambar 3.49 dan Gambar 3.50 merupakan rancangan antarmuka untuk Mengelola Fakultas. Tampilan ini menampilkan dua halaman yaitu halaman untuk melihat daftar Fakultas yang telah terdaftar di dalam sistem, serta halaman *form* untuk menambahkan fakultas baru dengan mengisi *dropdown* nama universitas, dan *input* nama fakultas.

Gambar 3.49 Rancangan tampilan Mengelola Fakultas bagian Melihat Fakultas.

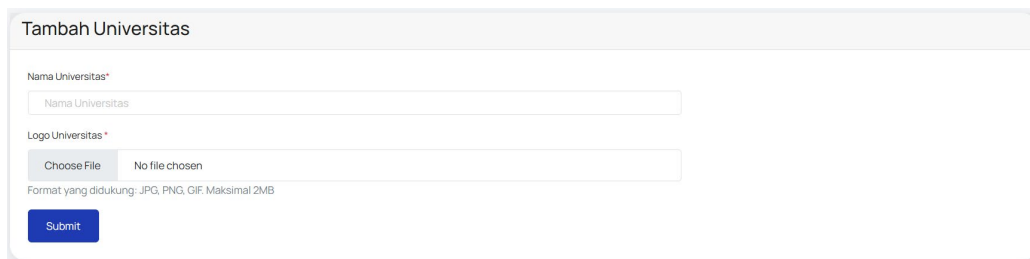
Gambar 3.50 Rancangan tampilan Mengelola Fakultas bagian *form* Tambah Fakultas.

11. Mengelola Universitas (UC-011/ACT-011)

Tampilan pada Gambar 3.51 dan Gambar 3.52 merupakan rancangan antarmuka untuk Mengelola Universitas. Tampilan ini menampilkan dua halaman yaitu halaman untuk melihat daftar universitas yang telah terdaftar di dalam sistem, serta halaman *form* untuk menambahkan universitas baru dengan mengisi *input* nama universitas dan mengunggah logo universitas dengan format JPEG, JPG, PNG dengan ukuran maksimal 2MB.



Gambar 3.51 Rancangan tampilan Mengelola Universitas bagian Melihat Universitas.



Gambar 3.52 Rancangan tampilan Mengelola Fakultas bagian *form* Tambah Fakultas.

12. Mengelola Mahasiswa (UC-012/ACT-012)

Tampilan pada Gambar 3.53, Gambar 3.54, dan Gambar 3.55 merupakan rancangan antarmuka untuk Mengelola Mahasiswa. Terdapat tiga halaman utama yang ditampilkan, yaitu halaman untuk melihat daftar mahasiswa yang sudah ditambahkan, halaman *form* untuk menambahkan mahasiswa dengan mengisi NPM, nama, dan angkatan. Selain itu, terdapat juga halaman *form* untuk mengubah data mahasiswa dengan elemen *form* yang sama.

List Mahasiswa

Show entries

Search:

#	NPM	Nama Mahasiswa	Angkatan	Action
1	2117051008	Hafiz Chandra	2021	Edit Delete
2	2117051010	Rasik Ahmad	2021	Edit Delete
3	2117051012	Jenifer Lusi	2021	Edit Delete
4	2217051008	Citra Pratiwi	2022	Edit Delete
5	2217051014	Surya Putra	2022	Edit Delete
6	2217051020	Wijaya Candra	2022	Edit Delete

Showing 1 to 6 of 6 entries

Previous **1** Next

Gambar 3.53 Rancangan tampilan Mengelola Mahasiswa bagian Melihat Mahasiswa.

Tambah Mahasiswa

NPM

Nama

Angkatan

[Submit](#)

Gambar 3.54 Rancangan tampilan Mengelola Mahasiswa bagian form Tambah Mahasiswa.

Edit Mahasiswa

NPM

Nama

Angkatan

[Update](#)

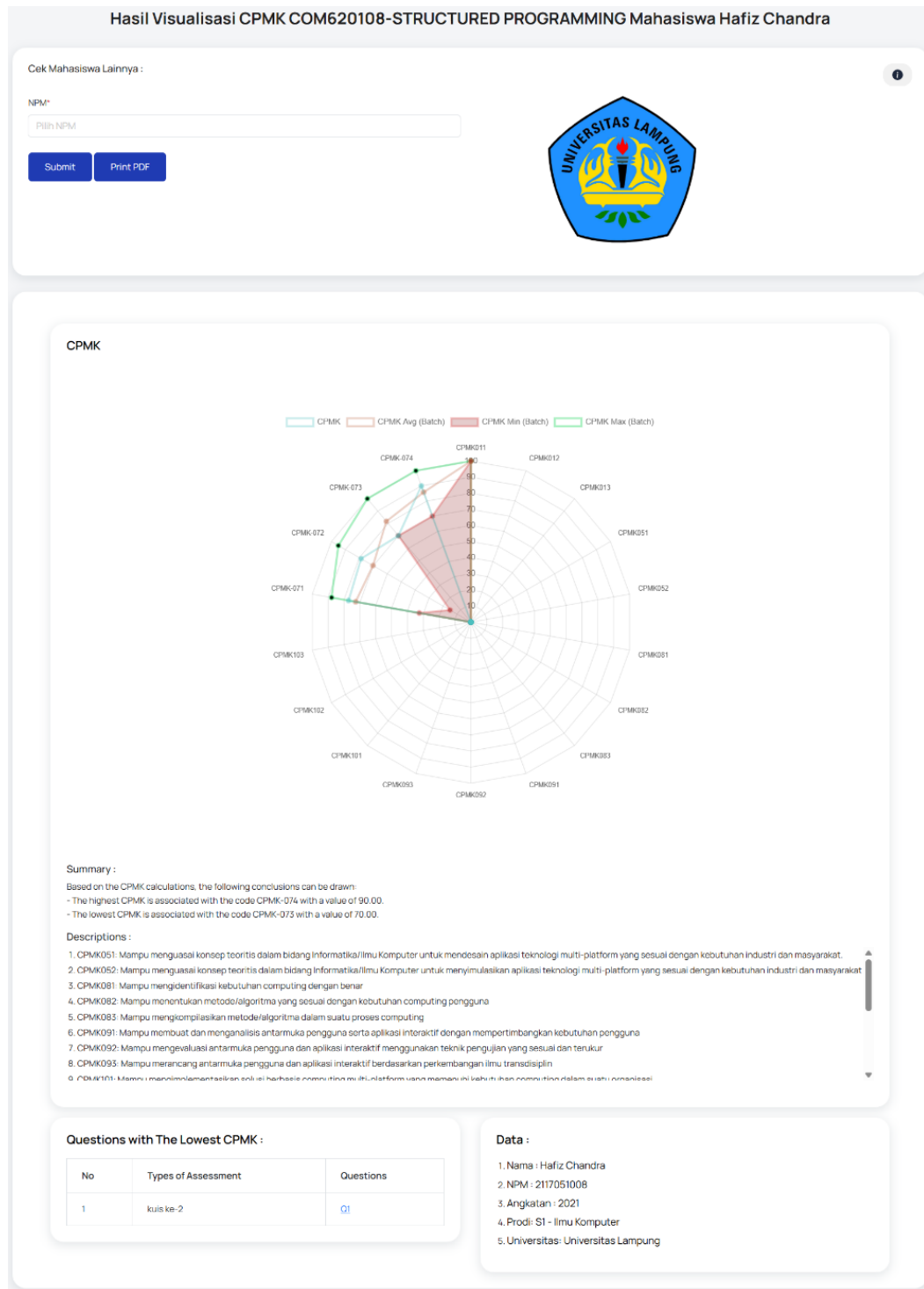
Gambar 3.55 Rancangan tampilan Mengelola Mahasiswa bagian *form* Ubah Mahasiswa.

13. Melihat Visualisasi CPL per Mahasiswa (UC-013/ACT-013) dan Mencetak laporan Visualisasi CPL per Mahasiswa (UC-014/ACT-014)

Tampilan pada Gambar 3.56 merupakan rancangan antarmuka untuk Melihat Visualisasi CPL per Mahasiswa dan Mencetak Laporan Visualisasi CPL per Mahasiswa. Tampilan ini menyediakan *form* yang terdiri dari *dropdown* Prodi yang hanya muncul untuk pengguna selain otoritas Penjamin Mutu Program Studi dan Dosen, *dropdown* untuk memilih Angkatan dan NPM, serta tombol Print PDF yang pada awalnya dalam keadaan nonaktif dan akan aktif setelah *form* *disubmit*. Setelah melakukan *submit*, sistem akan menampilkan beberapa komponen utama yaitu *form* Lihat CPMK untuk melihat Visualisasi CPMK per Mahasiswa (UC-016/ACT-016 atau UC-024/ACT-024), data mahasiswa, grafik Ketercapaian CPL Mahasiswa dalam bentuk *radar chart*, grafik Pemetaan CPL Angkatan Mahasiswa, tabel Soal dengan Nilai Terendah yang memengaruhi ketercapaian CPL, serta grafik dan tabel Pemetaan Profil Lulusan Berdasarkan CPL.

14. Melihat Visualisasi CPMK per Mahasiswa (UC-015/ACT-015 dan UC-023/ACT-023) dan Mencetak laporan Visualisasi CPMK per Mahasiswa (UC-016/ACT-016 dan UC-024/ACT-024)

Tampilan pada Gambar 3.57 merupakan rancangan antarmuka untuk Melihat Visualisasi CPMK per Mahasiswa dan Mencetak laporan Visualisasi CPMK per Mahasiswa. Tampilan ini menyediakan *form dropdown* untuk memilih NPM jika pengguna ingin melihat Visualisasi CPMK dari mahasiswa lain, serta tombol *Print PDF* untuk mencetak laporan. Setelah proses *submit*, sistem akan menampilkan grafik radar visualisasi ketercapaian CPMK mahasiswa pada mata kuliah tertentu, grafik pemetaan CPMK mahasiswa, tabel Soal dengan Nilai Terendah, serta data mahasiswa. Tampilan ini dapat diakses melalui *form* Lihat CPMK pada halaman Visualisasi CPL per Mahasiswa (UC-015/ACT-015) atau melalui *form* Lihat CPMK Individu yang terdapat pada Visualisasi CPMK per Mata Kuliah (UC-023/ACT-023).



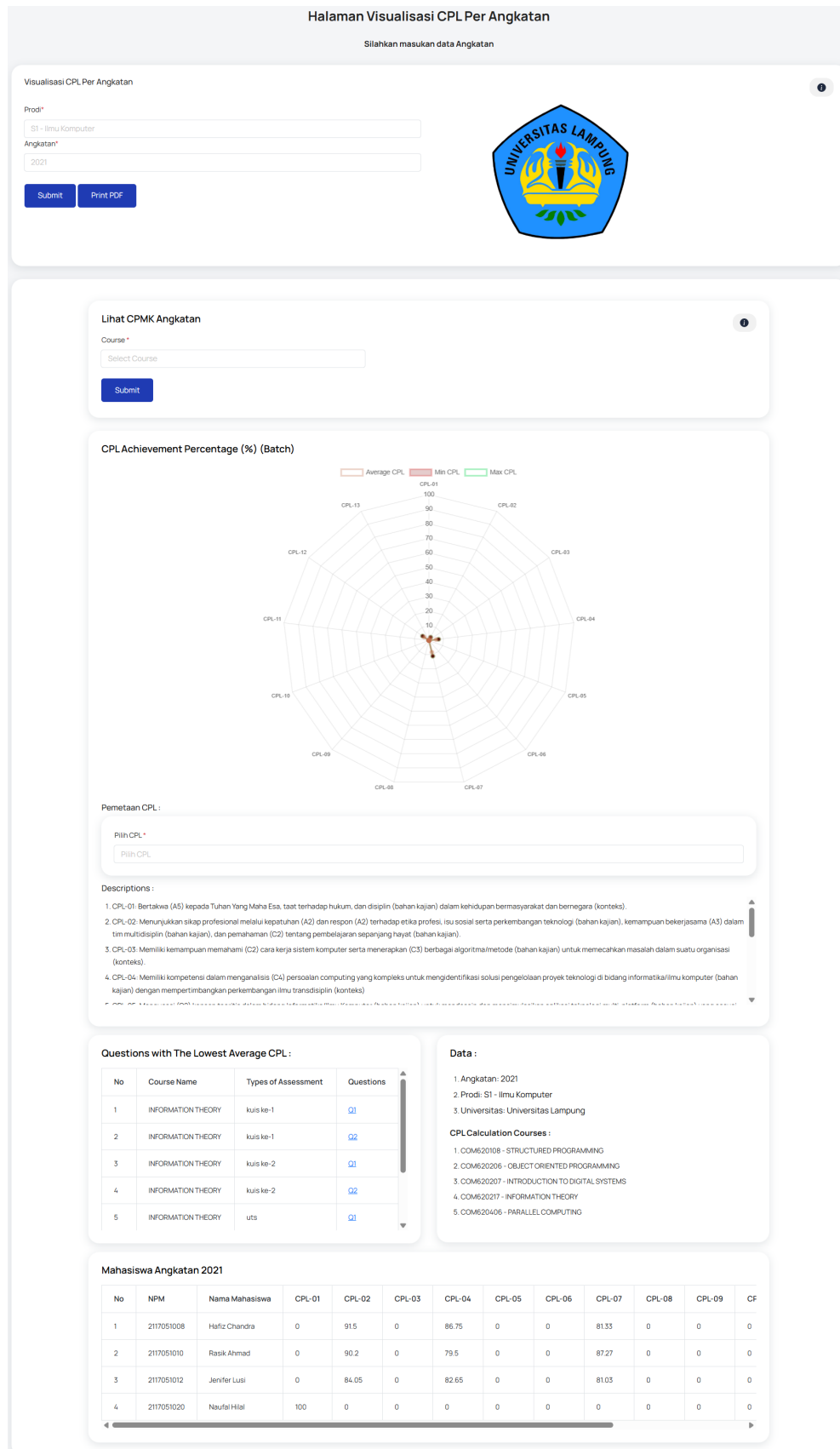
Gambar 3.57 Rancangan Tampilan Visualisasi CPMK per Mahasiswa.

15. Melihat Visualisasi CPL per Angkatan (UC-017/ACT-017) dan Mencetak laporan Visualisasi CPL per Angkatan (UC-018/ACT-018)

Tampilan pada Gambar 3.58 merupakan rancangan antarmuka untuk Melihat Visualisasi CPL per Angkatan dan Mencetak laporan Visualisasi CPL per Angkatan. Tampilan ini menyediakan *form* isian yang terdiri dari *dropdown* Prodi yang hanya muncul bagi pengguna selain otoritas Penjamin Mutu Program Studi dan Dosen serta *dropdown* Angkatan, serta tombol Print PDF yang awalnya dalam keadaan nonaktif dan akan aktif setelah *form* *disubmit*. Setelah melakukan *submit*, sistem akan menampilkan beberapa komponen utama yaitu *form* Lihat CPMK Angkatan untuk melihat Visualisasi CPMK per Angkatan (UC-020/ACT-020), grafik *radar chart* yang menggambarkan persentase ketercapaian CPL angkatan termasuk nilai rata-rata, minimum, dan maksimum, tabel soal dengan rata-rata CPL terendah, data informasi angkatan, serta tabel mahasiswa angkatan yang memuat nilai ketercapaian masing-masing CPL oleh setiap mahasiswa.

16. Melihat Visualisasi CPMK per Angkatan (UC-019/ACT-019) dan Mencetak laporan Visualisasi CPMK per Angkatan (UC-020/ACT-020)

Tampilan pada Gambar 3.59 merupakan rancangan antarmuka untuk melihat Visualisasi CPMK per Angkatan dan mencetak laporan Visualisasi CPMK per Angkatan. Tampilan ini menyediakan *form dropdown* untuk memilih Angkatan jika pengguna ingin melihat data CPMK dari angkatan lain, serta tombol Print PDF untuk mencetak laporan. Setelah proses *submit*, sistem akan menampilkan grafik radar visualisasi ketercapaian CPMK angkatan yang mencakup nilai rata-rata, minimum, dan maksimum, tabel Soal dengan nilai CPMK terendah, serta informasi data angkatan. Tampilan ini dapat diakses melalui form Lihat CPMK Angkatan pada halaman Visualisasi CPL per Angkatan (UC-017/ACT-017).

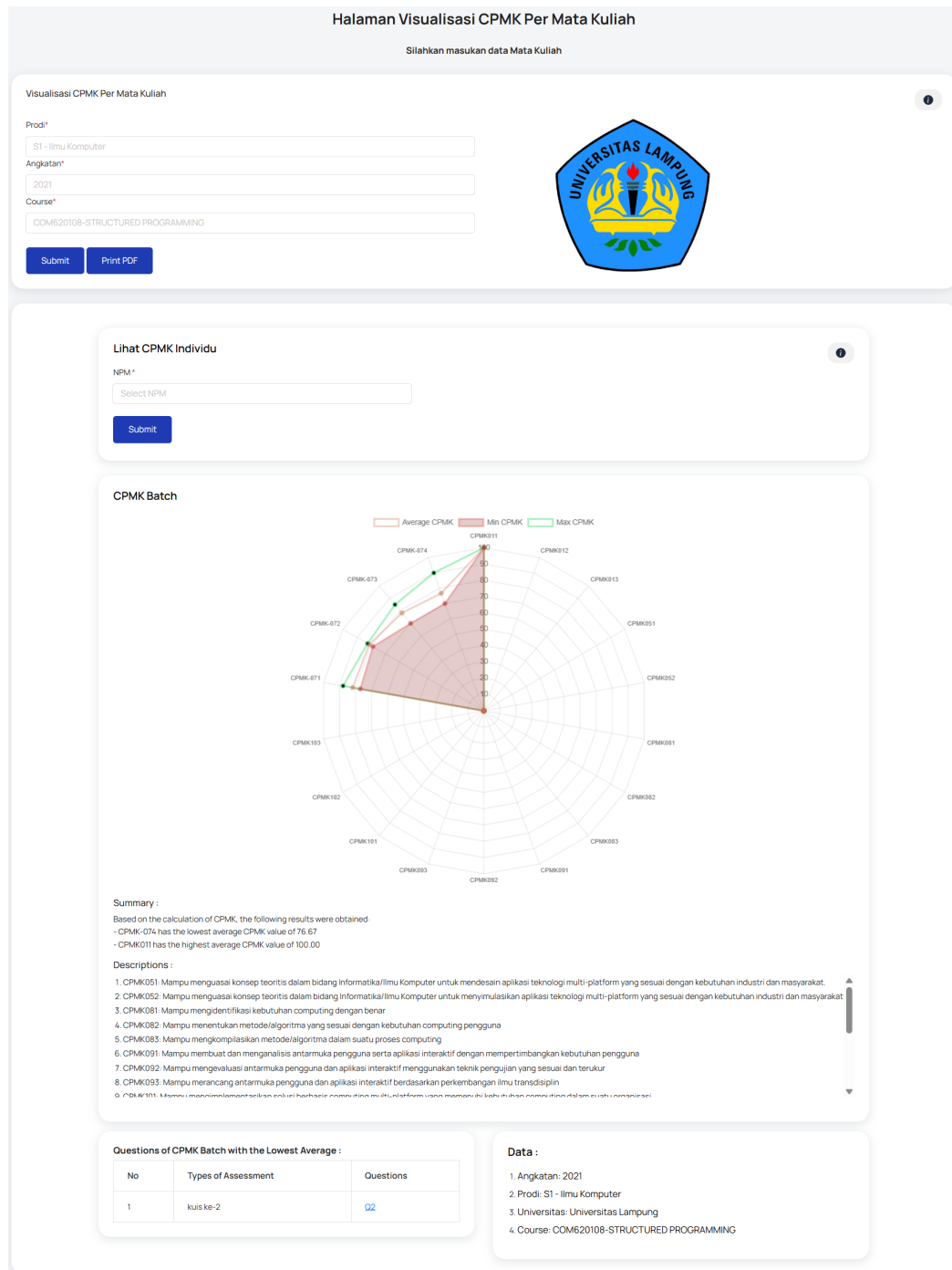


Gambar 3.58 Rancangan Tampilan Visualisasi CPL per Angkatan.



17. Melihat Visualisasi CPMK per Mata Kuliah (UC-021/ACT-021) dan Mencetak laporan Visualisasi CPMK per Mata Kuliah (UC-022/ACT-022)

Tampilan pada Gambar 3.60 merupakan rancangan antarmuka untuk Melihat Visualisasi CPMK per Mata Kuliah dan Mencetak laporan Visualisasi CPMK per Mahasiswa. Tampilan ini menyediakan *form* yang terdiri dari *dropdown* Prodi yang hanya muncul untuk pengguna selain otoritas Penjamin Mutu Program Studi dan Dosen, *dropdown* untuk memilih Angkatan dan Mata Kuliah, serta tombol Print PDF yang pada awalnya dalam keadaan nonaktif dan akan aktif setelah *form* *disubmit*. Setelah melakukan *submit*, sistem akan menampilkan beberapa komponen utama yaitu *form* Lihat CPMK Individu untuk melihat Visualisasi CPMK per Mahasiswa (UC-016/ACT-016 atau UC-024/ACT-024), grafik radar visualisasi CPMK angkatan yang menunjukkan nilai rata-rata, minimum, dan maksimum CPMK, tabel pertanyaan dengan nilai CPMK rata-rata terendah, serta informasi data angkatan, program studi, universitas, dan mata kuliah.



Gambar 3.60 Rancangan Tampilan Visualisasi CPMK per Mata Kuliah.

V. SIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, maka diperoleh kesimpulan sebagai berikut:

1. Telah dikembangkan modul manajemen pengguna dengan sistem multi-otoritas, yang memastikan setiap peran memiliki hak akses yang sesuai. Fitur ini mendukung keamanan data dan menyelaraskan alur kerja sistem dengan struktur organisasi akademik.
2. Telah diimplementasikan modul pelaporan yang mampu menghasilkan dokumen PDF dari data visualisasi CPL dan CPMK. Fitur ini mendukung proses analisis capaian pembelajaran secara terdokumentasi, baik pada level individu maupun angkatan.
3. Pengujian fungsional dengan metode *black-box* menunjukkan bahwa seluruh fitur utama sistem, mulai dari manajemen pengguna hingga pembuatan laporan, berjalan sesuai dengan semua skenario uji dan memenuhi kebutuhan fungsional.
4. Pengujian non-fungsional memvalidasi kualitas sistem dari sisi usabilitas, keamanan, dan kinerja. Sistem menunjukkan waktu muat halaman yang responsif, serta stabilitas tinggi dengan tingkat *error* 0.00% pada uji beban menggunakan 50 pengguna simultan, yang mendukung kesiapan sistem untuk diimplementasikan di lingkungan nyata.

5.2 Saran

Saran yang dapat diberikan untuk pengembangan sistem ke depannya adalah menyempurnakan fitur pengelolaan Rencana Pembelajaran Semester (RPS) serta mengembangkan fitur-fitur lain yang telah ada pada Sistem Penjamin Mutu Jurusan Ilmu Komputer.

DAFTAR PUSTAKA

- Astuti, T. P., Sofyana, L., & Bima, A. C. A. (2025). Multi-Roles Based Pada Rancang Bangun Surat Dengan Metode Rapid Application Development (RAD). *Journal Of Informatics And Busines*, 02(04), 555–566. <https://jurnal.ittc.web.id/index.php/jibs/article/view/2335>
- Aung, S. T., Aung, L. H., Funabiki, N., Yamaguchi, S., Syaifudin, Y. W., & Kao, W. C. (2024). An Implementation of Web-based Personal Platform for Programming Learning Assistant System with Instance File Update Function. *Engineering Letters*, 32(2).
- Bhatt, B., & Nandu, M. (2021). An Overview of Structural UML Diagrams. *International Research Journal of Engineering and Technology*, 8(8).
- Budiman, T., Kurniawan, E., Roland Hasibuan Program Studi Teknik Informatika, D., Studi Teknik Informatika, P., Studi Sistem Informasi, P., Studi Manajemen Informatika, P., & Jayakarta, S. (2023). RANCANG BANGUN SISTEM INFORMASI MANAJEMEN PROYEK PADA PT ABC. *Jurnal Manajemen Informatika Jayakarta*, 3(2), 128–141. <https://doi.org/10.52362/jmijayakarta.v3i2.1137>
- Elsayed, M., Elkashef, N., & Hassan, Y. F. (2020). Mapping UML sequence diagram into the web ontology language OWL. *International Journal of Advanced Computer Science and Applications*, 11(5). <https://doi.org/10.14569/IJACSA.2020.0110542>
- Friadi, J., & Gulo, J. R. (2020). Pengembangan Sistem Informasi Monitoring Prakrind Dengan Model Rapid Application Development. *Seminar Nasional Teknologi Informasi, Komunikasi dan Industri*, 12.
- Gupta, A., Rawal, A., & Barge, Y. (2021). Comparative Study of Different SDLC Models. *International Journal for Research in Applied Science and Engineering Technology*, 9(11). <https://doi.org/10.22214/ijraset.2021.38736>
- Gupta, S., Banga, J., Dabas, S., & Bhatia, Dr. M. K. (2022). A Comprehensive Study of Software Development Life Cycle Models. *International Journal for Research in Applied Science and Engineering Technology*, 10(12). <https://doi.org/10.22214/ijraset.2022.47868>
- Hidayat, A., & Irvanda, M. (2022). Optimalisasi Penyusunan dan Pembuatan Laporan untuk Mewujudkan Good Governance. *Jurnal Ilmiah Hospitality*, 11(1). <https://doi.org/https://doi.org/10.47492/jih.v11i1.1611>
- Indrianto. (2023). PERFORMANCE TESTING ON WEB INFORMATION SYSTEM USING APACHE JMETER AND BLAZEMETER. *Jurnal Ilmiah Ilmu Terapan Universitas Jambi*, 7(2). <https://doi.org/10.22437/jiituj.v7i2.28440>

- Jing, S., Wang, Q., Tan, K., & Guo, L. (2020). Develop and Implementation of Laboratory Document Management System. *International Conference on Education, Management, Computer and Society (EMCS2020)*, 343–348. <https://doi.org/10.38007/Proceedings.0001811>
- Kulkarni, Dr. R. N., & Prasad, P. P. R. (2021). Abstraction Of UML Class Diagram From The Input Java Program. *International Journal of Advanced Networking and Applications*, 12(04). <https://doi.org/10.35444/ijana.2021.12406>
- Kulkarni, Dr. R. N., & Srinivasa, C. K. (2021). Novel approach to transform UML Sequence diagram to Activity diagram. *Journal of University of Shanghai for Science and Technology*, 23(07). <https://doi.org/10.51201/jusst/21/07300>
- Laravel. (2025). *Laravel 8 Documentation*. Laravel. <https://laravel.com/docs/8.x>
- Matanari, W. S. P. (2024). *Pengembangan Sistem Manajemen Mutu Penilaian Mata Kuliah di Jurusan Ilmu Komputer* [Skripsi]. Universitas Lampung.
- Molla, M. M. I., Ahmad, J., & Wan Kadir, W. M. N. (2024). A Comparison of Transforming the User Stories and Functional Requirements into UML Use Case Diagram. *International Journal of Innovative Computing*, 14(1), 29–36. <https://doi.org/10.11113/ijic.v14n1.463>
- Noly, N. H., Usman, U., Tulus, T. J., & Iswan, I. E. (2022). Sistem Penjamin Mutu Internal dan Eksternal pada Lembaga Pendidikan Dasar. *Multiverse: Open Multidisciplinary Journal*, 1(2). <https://doi.org/10.57251/multiverse.v1i2.595>
- Palinggi, O., Sudarya Triana, Y., Permana, M. B., Huda, D. F., & Priyono, K. A. (2024). Entity-Relationship Diagram Technique in Database. *Collabits Journal*, 1(1). <https://doi.org/10.22441/collabits.2024.v1i1.001>
- Sholeh, M., Gisfas, I., Cahiman, & Fauzi, M. A. (2021). Black Box Testing on ukmbantul.com Page with Boundary Value Analysis and Equivalence Partitioning Methods. *Journal of Physics: Conference Series*, 1823(1). <https://doi.org/10.1088/1742-6596/1823/1/012029>
- Tamara, A. (2023). *Pengembangan Modul Admin Sistem Manajemen Mutu Perkuliahan di Jurusan Ilmu Komputer*. [Skripsi]. Universitas Lampung.
- Tenzin, S. (2022). PHP Framework for Web Application Development. *International Advanced Research Journal in Science, Engineering and Technology*, 9(2).
- Wahyudi, J., Asbari, M., Sasono, I., Pramono, T., & Novitasari, D. (2022). Database Management Education in MYSQL. *Edumaspul: Jurnal Pendidikan*, 6(2). <https://doi.org/10.33487/edumaspul.v6i2.4570>