

**DESAIN DAN IMPLEMENTASI *FRONT-END* SISTEM KINERJA DOSEN
UNIVERSITAS LAMPUNG BIDANG PENDIDIKAN DAN PENELITIAN
MENGUNAKAN *REACT JS***

(Skripsi)

Oleh

**AMALIA NURUL RAHMAWATI
NPM 2117051008**



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

**DESAIN DAN IMPLEMENTASI *FRONT-END* SISTEM KINERJA DOSEN
UNIVERSITAS LAMPUNG BIDANG PENDIDIKAN DAN PENELITIAN
MENGUNAKAN *REACT JS***

Oleh

AMALIA NURUL RAHMAWATI

Skripsi

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

Pada

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



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

ABSTRAK

DESAIN DAN IMPLEMENTASI *FRONT-END* SISTEM KINERJA DOSEN UNIVERSITAS LAMPUNG BIDANG PENDIDIKAN DAN PENELITIAN MENGUNAKAN *REACT JS*

Oleh

AMALIA NURUL RAHMAWATI

Sistem informasi yang efektif, efisien, dan terintegrasi diperlukan di perguruan tinggi karena kemajuan teknologi informasi. Sebagai lembaga pendidikan tinggi, Universitas Lampung membutuhkan sistem kinerja dosen yang dapat mendukung pengelolaan kinerja, terutama dalam hal pendidikan dan penelitian, serta dapat mempermudah akses data untuk akreditasi. Tujuan dari penelitian ini adalah untuk membuat dan menerapkan antarmuka pengguna (*front-end*) sistem kinerja dosen Universitas Lampung menggunakan React JS dengan pendekatan RESTful API dan autentikasi JWT. Metodologi Scrum digunakan dalam penelitian ini, yang mencakup tahapan perencanaan, desain, implementasi, dan pengujian sistem. Proses pengujian terdiri dari tiga tahapan yaitu pengujian *Black Box* untuk memverifikasi bahwa setiap fitur berjalan sesuai kebutuhan, pengujian *Lighthouse* untuk menilai performa, aksesibilitas, dan best practice, dengan hasil yang menunjukkan bahwa sistem responsif dan sesuai standar, serta pengujian *User Acceptance Testing* (UAT) untuk mengevaluasi kepuasan pengguna akhir. Hasil pengujian secara keseluruhan menunjukkan bahwa sistem yang dibuat berfungsi dengan baik, memenuhi kebutuhan pengguna, dan dapat memberikan akses mudah ke data kinerja dosen untuk membantu proses akreditasi dan pengambilan keputusan di Universitas Lampung.

Kata Kunci: Sistem Kinerja Dosen, React JS, RESTful API, JWT, Scrum, *Black Box Testing*, *Lighthouse*, *User Acceptance Testing* (UAT).

ABSTRACT

DESIGN AND IMPLEMENTATION OF THE FRONT-END LECTURER PERFORMANCE SYSTEM IN THE FIELDS OF EDUCATION AND RESEARCH AT THE UNIVERSITY OF LAMPUNG USING REACT JS

By

AMALIA NURUL RAHMAWATI

Effective, efficient, and integrated information systems are essential for higher education institutions in response to the rapid advancement of information technology. As a higher education institution, the University of Lampung requires a lecturer performance system that supports performance management, particularly in the fields of education and research, while also facilitating easier access to data for accreditation purposes. The objective of this research is to design and implement the front-end interface of the lecturer performance system at the University of Lampung using React JS, employing a RESTful API approach and JWT authentication. This study adopts the Scrum methodology, which includes the stages of planning, design, implementation, and system testing. The testing process consists of three stages: Black Box testing to verify that each feature functions as intended, Lighthouse testing to evaluate performance, accessibility, and best practices—demonstrating that the system is responsive and meets standards—and User Acceptance Testing (UAT) to assess end-user satisfaction. The overall testing results indicate that the developed system operates effectively, meets user requirements, and provides easy access to lecturer performance data, thereby supporting the accreditation process and decision-making at the University of Lampung.

Keywords: *Lecturer Performance System, React JS, RESTful API, JWT, Scrum, Black Box Testing, Lighthouse, User Acceptance Testing.*

Judul Skripsi : **DESAIN DAN IMPLEMENTASI *FRONT-END* SISTEM KINERJA DOSEN UNIVERSITAS LAMPUNG BIDANG PENDIDIKAN DAN PENELITIAN MENGGUNAKAN *REACT JS***

Nama Mahasiswa : ***Amalia Nurul Rahmawati***

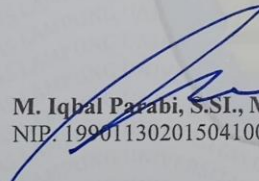
Nomor Pokok Mahasiswa : 2117051008

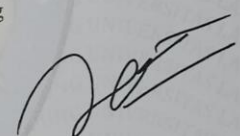
Program Studi : Ilmu Komputer (S-1)

Fakultas : Matematika dan Ilmu Pengetahuan Alam

MENYETUJUI

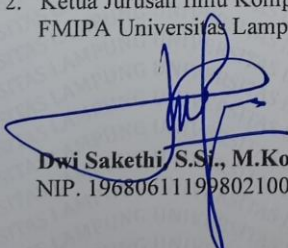
1. Komisi Pembimbing


M. Iqbal Parabi, S.ST., M.T.
NIP. 199011302015041002

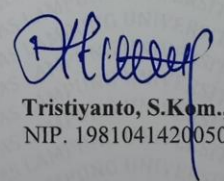

Igit Sabda Ilman, M.Kom.
NIK. 232111960101101

MENGETAHUI

2. Ketua Jurusan Ilmu Komputer
FMIPA Universitas Lampung


Dwi Sakethi, S.St., M.Kom.
NIP. 196806111998021001

3. Ketua Program Studi
FMIPA Universitas Lampung


Tristiyanto, S.Kom., M.I.S., Ph.D.
NIP. 198104142005011001

MENGESAHKAN

1. Tim Penguji

Ketua Penguji : M. Iqbal Parabi, S.SI., M.T.

Sekretaris Penguji : Igit Sabda Ilman, M.Kom.

Penguji Utama : Tristiyanto, S.Kom., M.I.S., Ph.D.

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



Dr. Eng. Heri Satria, S.Si., M.Si.
NIP. 197110012005011002

Tanggal Lulus Ujian Skripsi : 11 Agustus 2025

PERNYATAAN

Saya yang bertandatangan di bawah ini:

Nama : Amalia Nurul Rahmawati

NPM : 2117051008

Saya yang bertanda tangan di bawah ini menyatakan bahwa skripsi yang berjudul **"Desain dan Implementasi Front-End Sistem Kinerja Dosen Universitas Lampung Bidang Pendidikan dan Penelitian Menggunakan React JS"** adalah karya saya sendiri dan bukan merupakan karya orang lain. Penulisan skripsi saya telah mengikuti standar Universitas Lampung untuk penulisan karya tulis ilmiah. Jika di kemudian hari terbukti skripsi saya merupakan hasil jiplakan yang dibuat orang lain, maka saya bersedia menerima sanksi sesuai ketentuan yang berlaku.

Bandar Lampung, 28 Agustus 2025
Penulis,



Amalia Nurul Rahmawati
NPM. 2117051008

RIWAYAT HIDUP



Penulis bernama Amalia Nurul Rahmawati, lahir di Braja Harjosari pada tanggal 3 Maret 2003 sebagai anak bungsu dari dua bersaudara yang merupakan anak dari Bapak Suryadi dan Ibu Yadiyem. Pendidikan dasar yang telah penulis tempuh di SDN 1 Braja Harjosari dan lulus pada Tahun 2015, kemudian melanjutkan ke SMPN 1 Way Jepara dan lulus pada Tahun 2018, Setelah itu, penulis menyelesaikan pendidikan menengah di SMAN 1 Way Jepara dan lulus pada Tahun 2021. Pada Tahun 2021, penulis diterima perguruan tinggi melalui jalur Seleksi Nasional Masuk Perguruan Tinggi Negeri (SNMPTN) sebagai mahasiswa Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung. Selama menjadi mahasiswa penulis aktif mengikuti beberapa kegiatan yaitu:

1. Anggota Bidang Internal Himakom (Himpunan Mahasiswa Jurusan Ilmu Komputer) Universitas Lampung pada Tahun 2022.
2. Asisten dosen mata kuliah Multimedia semester Ganjil 2023.
3. Mengikuti kerja praktik di subbag OP1 pada PT Perkebunan Nusantara 1 Regional 7 pada Desember 2023 – Februari 2024.
4. Studi Independen pada kampus merdeka di infinite learning yang merupakan divisi dari PT Kinema Systrans Multimedia pada Februari – Juni 2024
5. Kuliah Kerja Nyata (KKN) Universitas Lampung periode 2 di Desa Gunung Agung, Lampung Timur pada Juni – Agustus 2024.

MOTTO

“Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya”

(Q.S Al-Baqarah: 286)

“Sesungguhnya beserta kesulitan ada kemudahan.”

(Q.S AL-Insyirah: 6)

“Keberhasilan bukanlah milik orang yang pintar, keberhasilan adalah kepunyaan mereka yang senantiasa berusaha”

(B.J Habibie)

PERSEMBAHAN

Alhamdulillah, puji syukur penulis panjatkan ke hadirat Allah SWT atas rahmat, taufik, hidayah, kesehatan dan kemudahan yang diberikan kepada penulis sehingga penulis dapat menyelesaikan skripsi ini dengan baik. Tanpa pertolongan Allah SWT, segala usaha, doa dan kerja keras tidak akan membuahkan hasil.

Dengan penuh rasa syukur, karya ini penulis persembahkan kepada:

Kedua Orang Tua dan Kakak Tercinta

Dengan penuh cinta dan hormat, untuk kedua orang tua dan kakak, yang selalu menjadi sumber kekuatan, doa, kasih sayang dan dukungan tanpa henti dalam setiap langkah hidup penulis. Sumber motivasi terbesar untuk perjalanan akademik ini adalah segala pengorbanan dan semangat yang mereka berikan.

Keluarga Tersayang

Yang selalu memberikan doa, dukungan dan kasih sayang yang tak ternilai. Kehangatan keluarga selalu menjadi penyemangat penulis dalam menghadapi berbagai rintangan dan tantangan selama menempuh pendidikan.

Sahabat-sahabat, Keluarga Besar Ilmu Komputer dan Almamater Tercinta

Yang senantiasa hadir sebagai bagian berharga dalam perjalanan ini, menjadi wadah menimba ilmu, berbagi pengalaman serta melalui proses pembelajaran.

SANWACANA

Puji syukur penulis panjatkan ke hadirat Allah SWT atas rahmat, hidayah dan karunia-Nya untuk menyelesaikan skripsi dengan judul "**Desain dan Implementasi Front-End Sistem Kinerja Dosen Universitas Lampung Bidang Pendidikan dan Penelitian Menggunakan React JS**". Shalawat dan salam senantiasa tercurah kepada junjungan Nabi Muhammad SAW, keluarga, sahabat dan pengikutnya hingga akhir kelak.

Penulis menyadari bahwa pengetahuan dan kemampuan selama proses penyusunan skripsi ini masih banyak keterbatasan baik dari segi pengetahuan maupun kemampuan. Oleh karena itu, penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membimbing, membantu dan mendukung skripsi ini. Secara khusus, terima kasih penulis sampaikan kepada:

1. Allah Subhanahu Wa Ta'ala, atas segala rahmat, nikmat dan kekuatan yang selalu diberikan kepada penulis termasuk kesehatan dan kesabaran dalam menghadapi tantangan yang muncul selama penyusunan skripsi ini. Segala proses yang dilewati tentu tidak terlepas dari pertolongan dan izin-Nya, mulai dari awal perkuliahan, penyusunan ide penelitian, pengumpulan data, hingga tahap akhir penulisan skripsi. Penulis meyakini bahwa penyusunan skripsi ini tidak akan berhasil tanpa izin-Nya.
2. Kedua orang tua dan kakak yang selalu memberi doa, kasih sayang, semangat dan dukungan. Dalam menghadapi setiap langkah perjalanan akademik, termasuk menyelesaikan skripsi ini, doa tulus menjadi kekuatan terbesar bagi penulis. Segala pengorbanan yang telah diberikan mulai dari dukungan finansial, perhatian tanpa henti dan nasihat yang penuh makna yang mendorong penulis untuk terus berusaha memberikan yang terbaik.

3. Bapak Dr. Eng. Heri. Satria, S.Si., M.Si. selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
4. Bapak Dwi Sakethi, S.Si., M.Kom. selaku Ketua Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
5. Ibu Yunda Heningtyas, M.Kom. selaku Sekretaris Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
6. Bapak Tristiyanto, S.Kom., M.I.S., Ph.D. selaku Ketua Program Studi Ilmu Komputer dan Dosen Pembahas yang telah memberikan kritik, saran dan masukan yang sangat membangun untuk memperbaiki skripsi ini.
7. Bapak Didik Kurniawan, S.SI., M.T. selaku Dosen Pembimbing Akademik yang telah memberikan arahan selama perkuliahan.
8. Bapak M. Iqbal Parabi, S.SI., M.T. selaku Dosen Pembimbing Utama yang dengan sabar dan penuh dedikasi telah membimbing, memberikan arahan, memberikan semangat serta memberikan masukan sehingga skripsi ini dapat diselesaikan dengan baik.
9. Bapak Igit Sabda Ilman, M.Kom. selaku Dosen Pembimbing Kedua yang dengan sabar mendampingi proses penelitian ini, memberikan saran dan membimbing penulis menyelesaikan skripsi dengan lebih baik.
10. Seluruh dosen Ilmu Komputer yang telah memberikan ilmu, pengalaman dan bimbingan selama masa perkuliahan dan menjadi bekal bagi penulis.
11. Seluruh staf Jurusan Ilmu Komputer atas bantuan dan pelayanannya yang mendukung kelancaran studi penulis dari awal perkuliahan hingga selesai.
12. Rekan-rekan satu tim yaitu shafa, Candra dan Abiyyi dalam penyusunan skripsi yang telah bekerja sama dalam setiap langkah mulai dari membuat ide, berdiskusi hingga menyelesaikan laporan.
13. Sahabat-sahabat yang selalu hadir yaitu Jenita, Dwika, Sarah, Salma, Cely dan Lathifah yang telah memberikan semangat, dukungan dan kebahagiaan selama perkuliahan hingga penyusunan skripsi ini serta menjadi tempat penulis berbagi keluh kesah dan solusi ketika penulis menghadapi berbagai masalah.
14. Keluarga besar Ilmu Komputer dan Himpunan Mahasiswa Ilmu Komputer yang telah menjadi bagian penting dalam proses perkuliahan serta turut

memberikan banyak pengalaman, pelajaran dan kebersamaan dalam perjalanan perkuliahan ini.

15. Kepada seluruh pihak yang hadir pada perjalanan perkuliahan ini baik sebagai teman maupun pendukung yang telah menambah makna pada proses belajar serta memberikan bantuan dan dukungan yang diberikan baik secara langsung maupun tidak langsung.
16. Terima kasih kepada diri sendiri yang telah berusaha untuk tetap kuat, sabar dan konsisten selama setiap tahap perkuliahan hingga skripsi ini selesai. Meskipun dihadapkan pada situasi yang penuh dengan kesulitan, kelelahan, dan bahkan keraguan penulis tetap semangat dan bangkit untuk melanjutkan apa yang telah penulis mulai. Penulis berterima kasih kepada dirinya sendiri atas perjalanan panjang yang penuh pembelajaran ini dengan setiap upaya, doa dan upaya yang penulis lakukan. Terima kasih telah berjuang, tidak menyerah dan terus berusaha hingga karya ini selesai dengan baik.

Semoga segala bantuan yang telah diberikan kepada penulis memperoleh balasan yang setimpal dari Allah Subhanahu wa Ta'ala. Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun demi perbaikan di masa mendatang. Besar harapan penulis agar skripsi ini dapat memberikan manfaat bagi berbagai pihak.

Bandar Lampung, 28 Agustus 2025
Penulis,



Amalia Nurul Rahmawati
NPM. 2117051008

DAFTAR ISI

	Halaman
DAFTAR TABEL	vii
DAFTAR GAMBAR.....	x
DAFTAR KODE	xii
I. PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan	3
1.5 Manfaat	3
II. TINJAUAN PUSTAKA.....	4
2.1 Penelitian Terdahulu	4
2.2 Beban Kinerja Dosen (BKD)	6
2.3 Remunerasi.....	6
2.4 <i>Website</i>	7
2.5 Sistem Informasi	7
2.6 Bidang	8
2.6.1 Pendidikan	8
2.6.2 Penelitian	8
2.7 <i>Front-End</i>	9
2.8 JavaScript	10
2.9 React JS.....	11
2.10 Next JS	12
2.11 Metode SCRUM	12
2.12 Pengujian.....	16

2.12.1 <i>Black-box Testing</i>	16
2.12.2 <i>Lighthouse</i>	16
2.12.3 Pengujian UAT	17
2.13 <i>Unified Modelling Language (UML)</i>	17
2.13.1 <i>Use Case Diagram</i>	18
2.13.2 <i>Use Case Description</i>	19
2.13.3 <i>Activity Diagram</i>	20
2.13.4 <i>Sequence Diagram</i>	21
III. METODE PENELITIAN	23
3.1 Waktu dan Tempat Penelitian	23
3.2 Alat Penelitian.....	23
3.2.1 Perangkat Lunak (<i>Software</i>)	24
3.2.2 Perangkat keras (<i>Hardware</i>).....	25
3.3 Tahapan Penelitian	25
3.3.1 Identifikasi masalah	26
3.3.2 Studi Literatur.....	28
3.3.3 Tahapan Pengumpulan Data	28
3.3.4 Penerapan Metode Scrum.....	29
IV. HASIL DAN PEMBAHASAN	82
4.1 Gambaran Umum	82
4.2 Implementasi Metode <i>SCRUM</i>	84
4.2.1 <i>Sprint 1</i>	84
4.2.2 <i>Sprint 2</i>	91
4.2.3 <i>Sprint 3</i>	101
4.2.4 <i>Sprint 4</i>	111
4.2.5 <i>Sprint 5</i>	120
4.3 Hasil Pengujian	131
V. KESIMPULAN DAN SARAN	148
5.1 Kesimpulan	148
5.2 Saran.....	148
DAFTAR PUSTAKA	150

DAFTAR TABEL

Tabel	Halaman
1. Tabel Penelitian.....	4
2. Simbol-Simbol <i>Use Case Diagram</i>	18
3. Simbol-Simbol <i>Activity Diagram</i>	20
4. Simbol <i>Sequence Diagram</i>	22
5. Waktu Penelitian	23
6. Perangkat Lunak Yang Digunakan	24
7. <i>Stakeholder</i>	30
8. <i>Product Backlog</i>	30
9. Penomoran <i>Use Case Diagram</i>	32
10. <i>Use Case Description</i> Melakukan Login	32
11. <i>Use Case Description</i> Mengelola Akun.....	34
12. <i>Use Case Description</i> Melihat Kinerja Bidang Pendidikan Tingkat Fakultas	35
13. <i>Use Case Description</i> Melihat Kinerja Bidang Penelitian Tingkat Fakultas..	37
14. <i>Use Case Description</i> Melihat Kinerja Bidang Pendidikan Tingkat Program Studi	39
15. <i>Use Case Description</i> Melihat Kinerja Bidang Penelitian Tingkat Prodi	40
16. <i>Use Case Description</i> Melihat Kinerja Bidang Pendidikan.....	42
17. <i>Use Case Description</i> Melihat Kinerja Bidang Penelitian.....	43
18. <i>Use Case Description</i> Melakukan Logout	44
19. <i>Sprint</i> ke-1	71
20. <i>Sprint</i> ke-2	71
21. <i>Sprint</i> ke-3	72
22. <i>Sprint</i> ke-4	73
23. <i>Sprint</i> ke-5	74
24. Skenario Pengujian <i>Positive Case Black Box</i>	76

25. Skenario Pengujian <i>Negative Case Black Box</i>	79
26. Skenario Pengujian <i>Responsivitas Black Box</i>	81
27. <i>Product Backlog Sprint ke-1</i>	84
28. <i>Sprint Planning Pada Sprint ke-1</i>	85
29. <i>Daily Scrum Pada Sprint ke-1</i>	85
30. <i>Increment Sprint ke-1</i>	89
31. <i>Sprint Review Sprint ke-1</i>	90
32. <i>Sprint Retrospective Sprint ke-1</i>	90
33. <i>Product Backlog Pada Sprint ke-2</i>	91
34. <i>Sprint Planning Pada Sprint ke-2</i>	91
35. <i>Daily Scrum Sprint ke-2</i>	92
36. <i>Increment Sprint ke-2</i>	98
37. <i>Sprint Review Pada Sprint ke-2</i>	100
38. <i>Sprint Retrospective Sprint ke-2</i>	100
39. <i>Product Backlog Sprint ke-3</i>	101
40. <i>Sprint Planning pada Sprint ke-3</i>	101
41. <i>Daily Scrum Sprint ke-3</i>	102
42. <i>Increment Sprint ke-3</i>	108
43. <i>Sprint Review Sprint ke-3</i>	110
44. <i>Sprint Retrospective Sprint ke-3</i>	110
45. <i>Product Backlog Sprint ke-4</i>	111
46. <i>Sprint Planning Sprint ke-4</i>	111
47. <i>Daily Scrum Sprint ke-4</i>	112
48. <i>Increment Sprint ke-4</i>	118
49. <i>Sprint Review Sprint ke-4</i>	119
50. <i>Sprint Retrospective Sprint ke-4</i>	120
51. <i>Product Backlog Sprint ke-5</i>	120
52. <i>Sprint Planning Sprint ke-5</i>	121
53. <i>Daily Scrum Sprint ke-5</i>	121
54. <i>Increment Sprint ke-5</i>	128
55. <i>Sprint Review Sprint ke-5</i>	130
56. <i>Sprint Retrospective Sprint ke-5</i>	131

57. Hasil Pengujian <i>Fungsionalitas Positive Case</i>	132
58. Hasil Pengujian <i>Fungsionalitas Negative Case</i>	137
59. Hasil Pengujian <i>Responsivitas black Box</i>	140
60. Hasil Pengujian <i>Lighthouse Desktop</i>	142
61. Hasil Pengujian <i>Lighthouse Mobile</i>	143
62. Kriteria Penilaian	145
63. Hasil UAT Kaprodi	145
64. Hasil UAT Dosen	146
65. Pengujian UAT Dekan	146

DAFTAR GAMBAR

Gambar	Halaman
1. Metode Scrum	13
2. Tahapan Penelitian	26
3. Arsitektur Sistem.....	27
4. <i>Use Case Diagram</i>	31
5. <i>Activity Diagram Login</i>	46
6. <i>Activity Diagram</i> Mengelola Akun	48
7. <i>Activity Diagram</i> Melihat Kinerja Bidang Pendidikan Tingkat Fakultas	49
8. <i>Activity Diagram</i> Melihat Kinerja Bidang Penelitian Tingkat Fakultas	51
9. <i>Activity Diagram</i> Melihat Kinerja Bidang Pendidikan Tingkat Prodi	52
10. <i>Activity Diagram</i> Melihat Kinerja Bidang Penelitian Tingkat Prodi	53
11. <i>Activity Diagram</i> Melihat Kinerja Bidang Pendidikan	54
12. <i>Activity Diagram</i> Melihat Kinerja Bidang Penelitian	55
13. <i>Activity Diagram</i> Melakukan Logout.....	56
14. <i>Sequence Diagram</i> Melakukan Login.....	57
15. <i>Sequence Diagram</i> Melakukan Mengelola Akun	58
16. <i>Sequence Diagram</i> Melihat Kinerja Bidang Pendidikan Tingkat Fakultas	59
17. <i>Sequence Diagram</i> Melihat Kinerja Bidang Penelitian Tingkat Fakultas	60
18. <i>Sequence Diagram</i> Melihat Kinerja Bidang Pendidikan Tingkat Prodi	61
19. <i>Sequence Diagram</i> Melihat Kinerja Bidang Penelitian Tingkat Prodi	62
20. <i>Sequence Diagram</i> Melihat Kinerja Bidang Pendidikan	63
21. <i>Sequence Diagram</i> Melihat Kinerja Bidang Penelitian	63
22. <i>Sequence Diagram</i> Melakukan Logout	64
23. Tampilan Login.....	65
24. Rancangan Tampilan Dasbor	65
25. Tampilan Mengelola Akun	66

26. Tampilan Tambah Akun	67
27. Tampilan Edit Akun.....	67
28. Tampilan Profile.....	68
29. Tampilan Daftar Program Studi	69
30. Tampilan Daftar Dosen	69
31. Tampilan Panduan & Bantuan Pada Sistem.....	70
32. Dokumentasi <i>Daily Scrum Sprint</i> ke-1	86
33. Dokumentasi Pembuatan <i>Use Case</i>	88
34. Dokumentasi Pembuatan <i>Activity Diagram</i>	88
35. Dokumentasi <i>Wireframe</i>	89
36. Dokumentasi <i>Hi-fi</i>	89
37. Dokumentasi <i>Daily Scrum Sprint</i> ke-2	92
38. Dokumentasi Halaman <i>Login Sprint</i> ke-2.....	94
39. Dokumentasi Halaman Dasbor Admin	96
40. Dokumentasi Halaman Dasbor Dekan	96
41. Dokumentasi Halaman Dasbor Kaprodi	97
42. Dokumentasi Halaman Dasbor Dosen	97
43. Dokumentasi <i>Daily Scrum Sprint</i> ke-3	102
44. Dokumentasi Halaman Mengelola Akun	104
45. Dokumentasi Halaman Tambah Akun	106
46. Dokumentasi Halaman Edit Akun.....	107
47. Dokumentasi <i>Daily Scrum Sprint</i> ke-4	112
48. Dokumentasi Hapus Akun	114
49. Dokumentasi Daftar Program Studi	116
50. Dokumentasi Daftar Dosen	117
51. <i>Daily Scrum Sprint</i> ke-5.....	122
52. Dokumentasi Halaman Profil	125
53. Dokumentasi Halaman kinerja fakultas	126
54. Dokumentasi Halaman Kinerja Program Studi.....	127
55. Dokumentasi Halaman Panduan & Bantuan.....	128

DAFTAR KODE

Kode	Halaman
1. Kode Program <i>Login</i>	95
2. Kode Program <i>Logout</i>	95
3. Kode Program Dasbor	98
4. Kode Melihat Daftar Pengguna.....	105
5. Kode Program Tambah Akun	107
6. Kode Edit Akun	108
7. Kode Hapus Akun	115
8. Kode Program Daftar Program Studi	117
9. Kode Daftar Dosen.....	118
10. Kode Program Halaman Profil.....	126

I. PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi informasi yang berkembang dengan sangat cepat telah menghadirkan berbagai kemudahan, khususnya dalam pemanfaatan teknologi. Perubahan ini memungkinkan akses informasi yang lebih praktis, pengelolaan data yang lebih efisien, serta penerapan teknologi yang semakin mendukung berbagai kebutuhan di berbagai bidang. Sebagai lembaga pendidikan tinggi di Indonesia, Universitas Lampung memiliki tanggung jawab besar untuk membangun sistem yang dapat mendukung pengembangan sumber daya manusia yang unggul. Kinerja dosen memengaruhi pencapaian institusi dan kualitas pendidikan dan penelitian. Oleh karena itu, sistem kompensasi atau remunerasi dosen merupakan komponen penting yang harus diperhatikan.

Universitas Lampung menggunakan sistem remunerasi yang didasarkan pada beban kerja dosen (BKD). Sistem ini didasarkan pada empat pilar Tri Dharma Perguruan Tinggi yang meliputi aspek bidang pendidikan, bidang penelitian, pengabdian, serta penunjang. Data yang dikumpulkan dari aktivitas dosen diubah menjadi tunjangan dan diberikan secara berkala. Namun, sistem ini masih memiliki keterbatasan karena data hanya disimpan secara lokal dan tidak terintegrasi dengan sistem lain terutama untuk akreditasi. Namun, data kinerja dosen diperlukan untuk proses akreditasi, terutama dalam bidang pendidikan dan penelitian. Saat ini, akses ke data tersebut terbatas hanya kepada dosen yang bersangkutan, sehingga tim akreditasi harus melalui proses administratif yang memakan waktu, yang pada akhirnya menghambat pengambilan keputusan dan akreditasi (Kinerja, 2022).

Untuk mengatasi masalah ini, dibutuhkan sistem baru yang dapat menjembatani sistem remunerasi dengan proses akreditasi. Solusi yang relevan adalah penggunaan teknologi React JS sebagai *framework* pengembangan antarmuka pengguna (*front-end*). Ini memungkinkan pengembangan antarmuka yang cepat, responsif, dan dinamis (Fariz et al., 2022). Serta memungkinkan integrasi yang mudah dengan RESTful API yang disediakan oleh *back-end*. Dengan mengintegrasikan API, itu memungkinkan pengelolaan data secara real-time dan meningkatkan pengalaman pengguna (Nursaid et al., 2020). Sehingga memudahkan program studi dan tim akreditasi untuk dapat memperoleh informasi yang diperlukan tanpa melalui proses manual.

Untuk memastikan sistem yang dibangun dapat memenuhi kebutuhan secara efektif, penelitian ini menggunakan metode Scrum yang merupakan bagian dari metodologi Agile. Scrum menekankan proses iteratif melalui *sprint* untuk menghasilkan fitur sistem yang fleksibel dan berfungsi secara bertahap. Selain itu, menjadi lebih mudah untuk menyesuaikan kebutuhan pengguna. Selain metode, *Blackbox Testing* digunakan sebagai pengujian fungsionalitas sistem tanpa melihat struktur internal program. Metode ini digunakan untuk memastikan bahwa setiap input memberikan output yang sesuai dengan persyaratan sistem. Oleh karena itu, sistem yang dibangun diharapkan dapat memenuhi standar kualitas perangkat lunak dalam hal keandalan dan kemudahan penggunaan. Diharapkan penelitian ini dapat menawarkan solusi praktis untuk meningkatkan efisiensi pengelolaan data kinerja dosen di Universitas Lampung dan juga menjadikan dasar sebagai pengembangan sistem kinerja yang lebih terpadu.

1.2 Rumusan Masalah

Berdasarkan dari latar belakang, dapat dirumuskan permasalahan, yaitu bagaimana mendesain dan mengimplementasi antarmuka *front-end* untuk sistem kinerja pada bidang pendidikan dan penelitian yang responsif untuk memungkinkan integrasi data yang akurat dan efektif dengan sistem lain yang mendukung akreditasi?

1.3 Batasan Masalah

Di dalam penelitian ini terdapat batasan masalah sebagai berikut:

1. Penelitian ini hanya akan membahas mengenai desain dan implementasi *front-end* pada sistem kinerja dosen di Universitas Lampung pada bidang pendidikan dan penelitian.
2. Sistem yang dikembangkan hanya akan mengelola data kinerja dosen dalam bidang pendidikan dan penelitian untuk keperluan akreditasi, tanpa mencakup pemrosesan atau evaluasi kompensasi dosen di bidang lain.
3. Pada sistem, proses pengambilan dan pengiriman data dilakukan dengan API yang telah disediakan oleh *backend*.

1.4 Tujuan

Penelitian ini bertujuan untuk mendesain dan mengimplementasi antarmuka *front-end* untuk sistem kinerja di bidang pendidikan dan penelitian yang responsif, mudah dipahami, dan mudah digunakan dengan menggunakan *framework* React JS. Selain itu, penelitian ini juga bertujuan untuk meningkatkan integrasi dan sinkronisasi data dengan sistem akreditasi yang relevan, sehingga informasi tentang kinerja dosen dapat diakses dengan lebih efisien dan akurat oleh program studi.

1.5 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat, yaitu:

1. Memberikan kontribusi dalam pengembangan sistem kinerja yang lebih efisien, terutama dalam memudahkan akses dosen terhadap informasi kompensasi mereka di bidang Pendidikan dan penelitian.
2. Memanfasilitasi program studi terkait program akreditasi dengan menyediakan sistem yang dapat memaksimalkan pengguna data kinerja dosen pada bidang Pendidikan dan penelitian secara akurat dan transparan.

II. TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian sebelumnya digunakan sebagai acuan dalam penelitian ini untuk memperkuat kerangka penyusunan dan memperluas pemahaman konsep yang relevan. Selain itu, penelitian sebelumnya juga berfungsi sebagai referensi untuk mempelajari literatur yang relevan dengan penelitian ini. Berikut ini penelitian terdahulu yang digunakan.

Tabel 1. Tabel Penelitian

No	Judul Penelitian	Teknologi yang digunakan	Hasil
1.	Implementasi React js Pada Pembuatan Sistem Informasi Digital Printing Berbasis <i>Website</i> (Nuraminudin et al., 2023)	React JS dipilih karena cepat, <i>scalable, fleksibel</i> , dan mempercepat pengerjaan sistem.	Sistem berbasis React JS dinilai mudah digunakan, efisien, mengurangi kesalahan, dan mendapat nilai kepuasan pengguna sebesar 3,53.
2.	DOM Benchmark Comparison Of The <i>Front-End</i> JavaScript	React, Angular, Vue, dan Svelte untuk	React dikenal memiliki kemampuan

No	Judul Penelitian	Teknologi yang digunakan	Hasil
	<i>Frameworks React, Angular, Vue, And Svelte (Levlin, 2020)</i>	perbandingan pada performa manipulasi DOM pada aplikasi <i>web</i> .	teknis yang kuat, terutama dalam hal manipulasi DOM seperti penambahan, pengeditan, dan penghapusan elemen, React adalah pilihan yang bagus untuk membangun antarmuka pengguna yang responsif.
3.	Benchmarking <i>Front-End Frameworks: A Deep Dive Into The Performance And Scalability Of React, Angular, And Vue.js</i> (Zareinia, 2024)	React, Angular, dan Vue.js untuk analisis kinerja dan skalabilitas aplikasi <i>web</i> modern. skalabilitas, dan konsumsi memori.	React unggul dalam kecepatan muat awal, efisiensi rendering, dan penggunaan memori berkat arsitektur minimalis, virtual DOM, serta pendekatan berbasis komponen. Hal ini membuat React ideal untuk

No	Judul Penelitian	Teknologi yang digunakan	Hasil
			membangun aplikasi kompleks yang tetap responsif dan mudah diskalakan.

2.2 Beban Kinerja Dosen (BKD)

Beban Kerja Dosen (BKD) merupakan tanggung jawab yang harus dipenuhi oleh dosen dalam menjalankan perannya sebagai pendidik dan ilmuwan dalam jangka waktu tertentu. Tugas ini mencakup pelaksanaan tri dharma perguruan tinggi, tugas tambahan, dan kegiatan penunjang yang harus dilaporkan setiap semester di universitas tempat dosen mengabdikan. BKD juga mengacu pada laporan kinerja dosen yang mencakup pendidikan, penelitian, pengabdian kepada masyarakat, serta kegiatan penunjang selama periode tertentu. Pelaporan BKD harus dibuat setiap semester dengan beban kerja minimal 12 SKS dan maksimal 16 SKS (*Informasi Umum Beban Kerja Dosen (BKD)*, n.d.).

2.3 Remunerasi

Remunerasi merupakan konsep yang berkaitan dengan sistem penggajian atau kompensasi pegawai, yang ditetapkan sesuai aturan dan diberikan secara rutin sesuai dengan nilai-nilai kerja atau kontribusi yang dihasilkan oleh setiap pegawai. Sistem ini bertujuan untuk menciptakan tata kelola perusahaan yang transparan, bersih, dan efektif untuk meningkatkan motivasi dan prestasi kerja pegawai melalui penghargaan yang adil atas upaya dan pencapaian mereka. Dengan demikian, sistem kompensasi diharapkan dapat menciptakan lingkungan kerja yang produktif dan profesional dengan memastikan bahwa setiap pegawai mendapatkan kompensasi yang sesuai dengan pekerjaan mereka (Samsudin, 2019).

2.4 Website

Website adalah salah satu media yang dapat digunakan untuk menyebarkan informasi dan untuk mempromosikan suatu produk serta dapat memungkinkan masyarakat luas untuk mengetahuinya (Nurlailah & Nova Wardani, 2023). Menurut (Teknologi et al., 2021) Website yaitu kumpulan halaman web yang saling berhubungan dengan membahas topik tertentu. Halaman-halaman tersebut berisi berbagai jenis media, seperti gambar, video, atau file lainnya. *Website* adalah platform digital yang memungkinkan orang mengakses berbagai informasi dan layanan untuk tujuan pribadi, bisnis, atau akademik. *Website* menjadi sumber informasi yang dinamis dan interaktif bagi pengunjungnya dengan menggabungkan berbagai konten. Setiap situs *web* dirancang untuk menyediakan informasi atau layanan, dan URL (Uniform Resource Locator) mereka berfungsi sebagai identitas unik yang memungkinkan pengguna menemukan dan mengunjungi situs tersebut di seluruh jaringan internet. Sebuah situs web adalah kumpulan halaman web yang tersedia di internet. Orang dapat mengaksesnya dengan mengetikkan alamat halaman web tersebut di halaman web mereka (Faruk Ulum & Rinaldi Muchtar, 2018).

2.5 Sistem Informasi

Menurut (Wijaya & Hendrastuty, 2022) sistem informasi adalah salah satu komponen penting yang dibutuhkan oleh suatu organisasi. Sehingga harus menerapkan sistem informasi dan teknologi terkini. Dengan menggunakan sistem ini, organisasi dapat lebih mudah meningkatkan efisiensi, memengaruhi, dan mendukung peningkatan kinerja. Menurut (Teknologi et al., 2021) sistem informasi adalah sistem yang menghasilkan keluaran berupa informasi yang berfungsi untuk berbagai tingkatan manajemen. Sistem ini sangat membantu dalam pengambilan keputusan di berbagai level organisasi, dari operasional hingga strategis. Sistem informasi membantu manajemen merencanakan, mengkoordinasikan, dan mengevaluasi kinerja organisasi dengan mengolah data menjadi informasi yang lebih terstruktur. Informasi yang relevan dan

akurat memungkinkan manajemen merespons perubahan dengan lebih cepat dan membuat keputusan yang lebih baik, yang membantu dalam mencapai tujuan dengan lebih efisien dan efektif.

2.6 Bidang

2.6.1 Pendidikan

Salah satu komponen utama Tri Dharma Perguruan Tinggi adalah pendidikan dan pengajaran, yang berfokus pada kewajiban perguruan tinggi untuk menjalankan proses belajar-mengajar yang berkualitas, relevan, dan berkelanjutan. Perguruan tinggi diposisikan sebagai lembaga utama untuk membina dan mengembangkan siswa melalui kurikulum dan pengalaman akademik yang bertujuan untuk meningkatkan kemampuan, keterampilan, dan pengetahuan siswa sehingga mereka dapat memenuhi standar kualifikasi yang dibutuhkan oleh masyarakat dan industri. Perguruan tinggi menciptakan lulusan yang unggul secara akademis dan siap menghadapi tantangan dalam dunia profesional dengan etos kerja yang tinggi dan kemampuan adaptasi yang kuat. Oleh karena itu, untuk memastikan bahwa lulusan dapat berkontribusi secara efektif terhadap kemajuan dan pembangunan bangsa, pendidikan dan pengajaran di perguruan tinggi harus terus berubah seiring dengan perubahan kebutuhan industri dan Masyarakat (Regita, 2023).

2.6.2 Penelitian

Pilar kedua dari Tri Dharma Perguruan Tinggi adalah penelitian, yang sangat penting untuk kemajuan ilmu pengetahuan, teknologi, dan seni. Perguruan tinggi berfungsi sebagai pusat inovasi dan pengembangan pemikiran kritis melalui pilar ini. Tujuannya adalah untuk memperluas batas pengetahuan saat ini dan menawarkan solusi untuk berbagai masalah yang dihadapi masyarakat. Diharapkan bahwa penelitian yang dilakukan oleh

perguruan tinggi memiliki efek positif pada perkembangan sosial dan ekonomi selain meningkatkan dunia akademik. Perguruan tinggi meningkatkan daya saing negara di tingkat internasional dengan melibatkan dosen dan mahasiswa dalam kegiatan penelitian. Ini mendorong ide-ide baru dan penemuan yang dapat digunakan untuk kemajuan teknologi atau karya seni inovatif. Selain itu, kegiatan penelitian ini memberikan peluang bagi perguruan tinggi untuk bekerja sama dengan lembaga riset, bisnis, dan pemerintah lainnya (Regita, 2023).

2.7 *Front-End*

Front-end merupakan bagian dari sistem yang berfokus pada penyediaan tampilan visual untuk pengguna. Tugas utama *front-end* adalah membuat elemen antarmuka interaktif dan mudah digunakan yang memungkinkan pengguna berinteraksi dengan sistem secara efisien. *Front-end* juga bertanggung jawab atas desain dan fungsi visual aplikasi atau situs *web*, seperti tata letak, navigasi, dan responsivitas, sehingga pengalaman pengguna menjadi mudah dipahami dan menarik (Rizaldy & Dirgahayu, 2020). Menurut (Dwiyanto & Putro, 2022) *Front-end* adalah bagian dari antarmuka sistem yang berfokus pada aspek visual seperti tampilan yang menarik dan mudah digunakan. Tugas utama *front-end* adalah memastikan antarmuka menarik dan mudah digunakan oleh pengguna. Seorang *front-end developer* bertanggung jawab untuk membuat tampilan situs *web* atau aplikasi yang interaktif dan menarik bagi pengguna. Mereka menggunakan bahasa pemrograman seperti HTML untuk membuat struktur halaman, CSS untuk mengatur gaya dan tata letak, dan JavaScript untuk menambahkan elemen yang interaktif dan dinamis. Penerjemahan desain menjadi kode yang berfungsi, memastikan tampilan bekerja dengan baik di berbagai perangkat, dan mempertimbangkan responsivitas dan pengalaman pengguna yang optimal adalah semua tanggung jawab seorang *front-end developer* (Adam, 2021). *Front-end developer* sering bekerja sama dengan *back-end developer*, dan kedua peran ini sangat penting dalam proses pengembangan aplikasi. *Front-end developer* berkonsentrasi pada

pengembangan tampilan visual dan interaksi yang langsung dilihat dan digunakan oleh pengguna, seperti tata letak, desain, dan navigasi. *Back-end* developer menangani bagian di balik layar, seperti pengelolaan *server*, database, dan database. Kolaborasi antara keduanya memastikan bahwa aplikasi berfungsi dengan baik secara teknis dan menarik secara visual (Subeni, 2023).

2.8 JavaScript

JavaScript merupakan bahasa pemrograman populer untuk membuat aplikasi *web* interaktif. JavaScript awalnya dirancang untuk memungkinkan interaktivitas pada halaman *web* dengan menjalankan skrip di sisi klien. Ini berarti skrip dapat dijalankan pada perangkat pengguna, seperti browser *web*, dan memungkinkan pengguna untuk mengubah elemen halaman dan menanggapi peristiwa tanpa perlu memuat ulang halaman. Pada akhirnya, platform seperti Node.js memungkinkan pengembang untuk menggunakan JavaScript di sisi *server* dan memungkinkan integrasi yang lebih baik antara logika di sisi klien dan *server* saat mengembangkan aplikasi *web* (Noviantoro et al., 2022). JavaScript memiliki sintaks yang serupa dengan bahasa pemrograman lain seperti Java dan C. Sebagai bahasa pemrograman yang mendukung paradigma berorientasi objek, JavaScript memungkinkan pengembang untuk membuat aplikasi *web* yang dinamis dan interaktif di mana elemen-elemen halaman *web* dapat berinteraksi secara langsung dengan pengguna tanpa perlu memuat ulang halaman *web* secara keseluruhan. JavaScript juga dikenal fleksibel karena mendukung berbagai gaya pemrograman. Ekosistem JavaScript yang luas menempatkan JavaScript sebagai salah satu kekuatan terbesar. Banyaknya *framework* dan pustaka (*library*) yang telah dikembangkan untuk menyederhanakan dan mempercepat proses pembuatan aplikasi *web*. Misalnya, *framework* seperti React memungkinkan pengembang untuk berkonsentrasi pada logika bisnis dan desain tanpa menghabiskan banyak waktu pada pengaturan teknis dasar. Mereka juga menawarkan alat yang efektif untuk membuat antarmuka pengguna (UI) yang modern dan responsif serta memfasilitasi manajemen

komponen-komponen yang kompleks di dalam sebuah aplikasi (Rahmadhani et al., 2024).

2.9 React JS

ReactJS adalah kerangka kerja open source yang membangun antarmuka pengguna dengan *library* JavaScript. ReactJS sering digunakan untuk mengembangkan aplikasi *single-page* dan *mobile* karena kemampuan untuk memberikan kecepatan, kesederhanaan, dan skalabilitas dalam pengembangan aplikasi (Khuat & Tung, 2018). ReactJS menggunakan pendekatan berbasis komponen, yang memungkinkan pengembang membagi antarmuka pengguna menjadi komponen kecil yang dapat diatur secara mandiri. Setiap komponen memiliki state yang berbeda dan dapat digabungkan untuk membuat antarmuka yang lebih kompleks. Pendekatan ini menghemat waktu dan meningkatkan konsistensi tampilan karena bagian-bagian tersebut dapat digunakan kembali di berbagai bagian aplikasi tanpa menulis ulang kode, serta dapat meningkatkan efisiensi pengembangan.

Salah satu fitur utama ReactJS adalah penggunaan Virtual DOM, yang secara signifikan meningkatkan kinerja aplikasi. Ketika data berubah, ReactJS tidak langsung memperbarui DOM sebenarnya, tetapi membandingkan perubahan tersebut dengan Virtual DOM. Mekanisme ini mengurangi operasi manipulasi DOM yang mahal, sehingga menghasilkan pengalaman pengguna menjadi lebih responsive (Rahmadhani et al., 2024).

Di dalam react js juga terdapat keunggulan dibandingkan framework *frontend* lainnya, seperti kecepatan rendering yang tinggi, kemampuan untuk membangun komponen UI yang interaktif. *Framework* ini didukung oleh berbagai pustaka pendukung seperti *react router* sebagai navigasi halaman dan *redux* sebagai manajemen *state* global, yang memperluas kemampuannya dalam membangun aplikasi berskala besar. Selain itu, dukungan komunitas yang kuat berarti tersedianya sumber belajar yang melimpah, dokumentasi

lengkap, dan solusi untuk berbagai masalah pengembangan yang dapat muncul (Iswari & Nasution, 2021). ReactJS dapat diintegrasikan dengan berbagai teknologi lain karena fleksibilitasnya, seperti *react native* untuk pengembangan aplikasi *mobile* dan pengembangan web progresif (PWA). Ini menjadikannya pilihan yang serbaguna untuk berbagai kebutuhan pengembangan perangkat lunak kontemporer

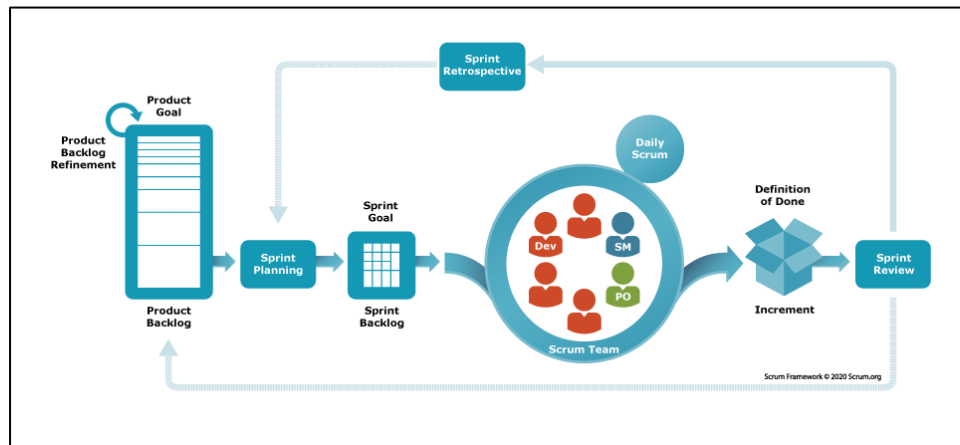
2.10 Next JS

Next.js, *framework* yang dibangun di atas React.js, membantu mengembangkan aplikasi *web* dengan fitur seperti rendering sisi *server* (SSR), pengelolaan rute otomatis, dan pembagian kode yang efektif. Next.js memungkinkan pengembang meningkatkan kinerja aplikasi dan pengalaman pengguna dengan memungkinkan pre-rendering halaman. Selain itu, *framework* ini memungkinkan pengembangan aplikasi dinamis dan statis dan memungkinkan integrasi yang mudah dengan berbagai API dan layanan *backend* (Srivastava et al., 2024). Hal ini menyoroti popularitas Next.js di kalangan pengembang karena kemudahan penggunaannya dan kemampuan untuk membuat aplikasi *web* kontemporer

2.11 Metode SCRUM

Metode Scrum adalah pendekatan pengembangan sistem yang mengutamakan kerja tim dan interaksi yang efektif. Fokus utama metode ini yaitu strategi, fleksibilitas, dan kecepatan dalam merespons kebutuhan sistem yang dinamis. Scrum membagi proses pengembangan menjadi unit kerja kecil yang terstruktur, memungkinkan setiap anggota tim untuk berkontribusi secara efektif pada tujuan bersama. Untuk memastikan bahwa setiap tugas dapat diselesaikan secara efisien dan responsif terhadap perubahan, kolaborasi yang kuat dan komunikasi yang teratur antara anggota tim sangat penting. Dengan metode ini, Scrum membantu memastikan bahwa sistem dibuat sesuai dengan kebutuhan pengguna dan siap untuk disesuaikan jika diperlukan perubahan,

menjaga kelincahan proses di setiap tahap pengembangan (Ardiansyah et al., 2021).



Gambar 1. Metode Scrum

Di dalam scrum terdapat *stakeholder* yang terlibat didalamnya yaitu:

a. Product Owner

Product Owner adalah salah satu orang yang bertanggung jawab untuk memaksimalkan nilai dari produk yang dikembangkan oleh tim Scrum. Tugas utama *Product Owner* adalah mengelola dan memprioritaskan Backlog Produk, yang mencakup daftar fitur, perbaikan, dan kebutuhan lainnya yang harus diselesaikan oleh tim.

b. Scrum Master

Scrum Master adalah fasilitator yang memastikan bahwa tim memahami dan mengikuti prinsip-prinsip dan praktik dari Scrum, *Scrum Master* membantu mereka menemukan dan mengatasi masalah yang dapat mengganggu proses. *Scrum Master* juga membantu tim dalam memahami cara kerja *Scrum*, sehingga proses pengembangan produk dapat berjalan lancar dan efisien.

c. Developer tim

Development Team adalah tim pengembang yang bertanggung jawab untuk membuat produk yang dapat digunakan dan memenuhi standar kualitas yang ditetapkan.

d. Business Stakeholder

Business stakeholder adalah pihak yang memiliki kepentingan dalam produk atau proyek yang dikembangkan dan akan memanfaatkannya baik secara langsung maupun tidak langsung. Mereka biasanya berasal dari luar tim pengembang, seperti pelanggan atau pengguna. *Stakeholder* bisnis memainkan peran penting dalam memberikan masukan strategis, memberikan umpan balik, dan memastikan bahwa produk memenuhi kebutuhan pasar dan tujuan. Mereka tidak terlibat secara langsung dalam proses pengembangan sehari-hari, tetapi peran mereka sangat penting untuk memastikan bahwa produk yang mereka buat memiliki nilai tambah dan membantu keberhasilan bisnis.

Berikut ini merupakan tahapan dari metode *SCRUM* sebagai berikut:

a. Product Backlog

Ini mencakup semua fitur yang diinginkan, perbaikan, dan kebutuhan untuk pengembangan sistem. *Product backlog* dikelola oleh *product Owner* yang bertugas untuk menyusun item-item yang terdapat di dalam *backlog* berdasarkan prioritas, sehingga tim dapat berkonsentrasi pada kebutuhan yang paling penting.

b. Sprint Planning

Pada tahapan ini, tim Scrum berkumpul untuk merencanakan *sprint* berikutnya. Mereka memilih item dari *product backlog* yang dapat diselesaikan dalam waktu *sprint*, biasanya satu hingga empat minggu.

Kemudian item-item ini dipindahkan ke *Sprint Backlog*, atau daftar tugas untuk *sprint* tersebut.

c. Sprint

Setiap *sprint* memiliki tujuan yang jelas, dan tim berusaha menyelesaikan pekerjaan yang telah direncanakan agar fitur atau fungsi tertentu dapat diterapkan atau diuji pada akhir *sprint*. *Sprint* juga merupakan periode waktu terfokus di mana tim bekerja untuk menyelesaikan item dalam *backlog sprint*.

d. Daily Scrum

Ini merupakan pertemuan singkat yang diadakan setiap hari selama *sprint*, yang biasanya berdurasi sekitar 15 menit. Tujuannya adalah agar setiap anggota tim melaporkan apa yang telah mereka lakukan, apa yang akan mereka lakukan selanjutnya, dan potensi tantangan. Pertemuan ini membantu tim tetap sinkron dan menangani masalah dengan cepat.

e. Sprint Review

Setelah *sprint* berakhir, tim mengadakan pertemuan *sprint review* untuk menilai hasil pekerjaan mereka. Tim mempresentasikan apa yang telah mereka lakukan, dan *Product Owner* memutuskan apakah hasil tersebut sesuai dengan ekspektasi atau perlu dilakukan penyesuaian. Untuk memungkinkan pemangku kepentingan lain untuk memberikan umpan balik langsung, peninjauan simpul juga melibatkan mereka.

f. Sprint Retrospective

Setelah *sprint review*, tim melakukan *sprint retrospective*, di mana mereka memikirkan proses kerja selama *sprint*, menemukan hal-hal yang berjalan baik serta hal-hal yang perlu ditingkatkan, dan membuat

rencana untuk perbaikan untuk *sprint* berikutnya. Pada titik ini, tujuan adalah untuk terus meningkatkan kinerja dan efisiensi tim sehingga dapat dicapai dalam jangka Panjang

2.12 Pengujian

2.12.1 Black-box Testing

Black box testing dilakukan untuk memastikan bahwa sistem bekerja sesuai dengan persyaratan fungsional yang telah ditentukan. Pengujian black box dilakukan dengan memberikan berbagai input ke sistem, memeriksa outputnya, dan membandingkannya dengan spesifikasi yang diharapkan. Metode ini fokus pada evaluasi fungsionalitas tanpa memperhatikan logika atau kode internal sistem (Bota & Setiyawati, 2022)

2.12.2 Lighthouse

Lighthouse testing adalah alat yang digunakan untuk mengevaluasi kualitas dan performa aplikasi *web*, termasuk aplikasi berbasis *framework* seperti Code Igniter (Syarifudin et al., 2021). Dalam konteks sistem penjualan yang dibangun dengan arsitektur Model View Controller (MVC), *Lighthouse* dapat mengevaluasi metrik seperti aksesibilitas, waktu muat halaman, dan praktik terbaik pengembangan *web*. *Lighthouse* membantu pengembang menemukan area yang perlu diperbaiki untuk meningkatkan pengalaman pengguna dan efisiensi aplikasi. Ini memungkinkan pengembang memastikan bahwa aplikasi tidak hanya beroperasi dengan baik tetapi juga memenuhi standar kualitas yang tinggi.

2.12.3 Pengujian UAT

Proses evaluasi sistem oleh pengguna akhir untuk memastikan bahwa sistem memenuhi kebutuhan dan harapan mereka disebut *User Acceptance Testing* (UAT). Proses ini dimulai dengan persiapan, yang mencakup pemahaman mendalam tentang spesifikasi fungsionalitas dan persyaratan bisnis (Aliyah Aliyah et al., 2024). Selanjutnya, UAT dilanjutkan dengan pembuatan rencana pengujian yang mencakup skenario pengujian. Berbeda dengan unit testing, yang berkonsentrasi pada pengujian komponen sistem tertentu (Wulandari et al., 2023), UAT mengutamakan pengalaman pengguna secara keseluruhan. Pengembang dapat menggunakan metode ini untuk memastikan bahwa perangkat lunak yang mereka buat benar-benar memenuhi ekspektasi dan diterima secara positif oleh pengguna (Fitriastuti et al., 2024)

2.13 *Unified Modelling Language* (UML)

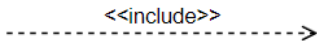
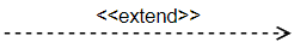
Unified Modeling Language (UML) adalah sebuah bahasa yang divisualisasikan melalui gambar atau grafik yang dapat digunakan untuk memberikan gambaran dan spesifikasi yang jelas tentang pengembangan dan dokumentasi sistem berorientasi objek melalui penggunaan gambar atau grafik. UML menyediakan standar pembuatan cetak biru sistem (blueprint). Standar ini mencakup berbagai elemen, seperti konsep proses bisnis, perancangan kelas yang dapat digunakan dalam bahasa pemrograman tertentu, perancangan basis data, dan sejumlah komponen yang diperlukan untuk proses pengembangan sistem. UML terdiri dari diagram yang memodelkan masalah dan solusi, yang membantu dalam merancang sistem yang lebih terorganisir dan efisien (Siska Narulita et al., 2024). Adapun beberapa diagram yang dideskripsikan oleh UML sebagai berikut.

2.13.1 Use Case Diagram

Use case *diagram* menunjukkan bagaimana aktor berinteraksi dengan sistem dan bagaimana mereka melakukan aktivitas tertentu yang berkaitan dengan proses yang dilakukan oleh sistem. Diagram ini berfungsi untuk menjelaskan bagaimana sistem informasi bekerja dengan menampilkan skenario atau tindakan yang dilakukan oleh aktor saat menggunakan sistem yang akan dibangun (Hasanah, 2020). Adapun simbol – simbol *use case diagram* disajikan pada tabel berikut.

Tabel 2. Simbol-Simbol *Use Case Diagram*

No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Aktor melakukan tugas untuk berinteraksi dengan sistem.
2		<i>Association</i>	<i>Association</i> adalah relasi yang menghubungkan antara satu objek dengan objek lain.
3		<i>Dependency</i>	<i>Dependency</i> adalah relasi yang menghubungkan perubahan salah satu elemen yang berpengaruh pada elemen lain.
4		<i>Use Case</i>	<i>Use Case</i> menjelaskan interaksi antara aktor dengan sistem.

No	Gambar	Nama	Keterangan
			
5		Relasi <i>Include</i>	Relasi ini menghubungkan antara <i>use case</i> dengan <i>use case</i> tambahan, di mana <i>use case</i> tambahan tidak dapat menjalankan fungsinya tanpa <i>use case</i> .
6		Relasi <i>Extend</i>	Relasi ini menghubungkan suatu <i>use case</i> dengan <i>use case</i> tambahan, di mana <i>use case</i> tambahan dapat menjalankan fungsinya walau tanpa <i>use case</i> .

2.13.2 Use Case Description

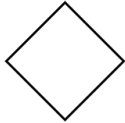
Dalam penelitian, *use case description* digunakan untuk menggambarkan interaksi antara aktor dan sistem dalam situasi tertentu. Deskripsi ini mencakup elemen penting seperti aktor, tujuan, dan langkah-langkah yang diperlukan untuk mencapai tujuan tersebut. Dengan menggunakan pendekatan studi kasus, peneliti dapat mengeksplorasi fenomena secara menyeluruh dan memahami konteks yang mempengaruhi interaksi sehingga mereka dapat membuat rekomendasi yang lebih relevan. (Sitorus, 2021) Jurnal tersebut menyatakan bahwa deskripsi contoh penggunaan membantu dalam dokumentasi kebutuhan pengguna dan memberikan pemahaman yang lebih baik tentang sistem yang sedang dipelajari.

2.13.3 Activity Diagram

Activity diagram menggambarkan berbagai alur aktivitas yang ada dalam sistem yang dirancang. Diagram ini mencakup bagaimana alur tersebut dimulai, keputusan yang mungkin terjadi, dan bagaimana alur tersebut berakhir. Selain itu, diagram ini dapat menampilkan proses paralel yang mungkin terjadi selama eksekusi. *Activity diagram* adalah jenis state diagram yang unik di mana sebagian besar state adalah aksi dan sebagian besar transisi disebabkan oleh selesainya state sebelumnya (pemrosesan internal). Oleh karena itu, aktivitas diagram tidak secara akurat menggambarkan interaksi antar subsistem atau perilaku internal sistem. Sebaliknya, mereka lebih memfokuskan pada memberikan gambaran umum proses dan jalur aktivitas dari tingkat yang lebih tinggi. Diagram ini digunakan untuk menunjukkan langkah-langkah atau aktivitas yang ada dalam suatu sistem, biasanya, setiap kasus memiliki satu *Activity diagram*. Dalam model bisnis, diagram ini sering digunakan untuk menunjukkan alur proses bisnis karena menunjukkan urutan aktivitas yang terlibat dalam proses bisnis. *Activity diagram* terdiri dari satu atau lebih *use case diagram* dan menunjukkan bagaimana proses berlangsung dalam sistem. Sebuah *Activity diagram* dapat menunjukkan proses yang sedang berjalan, sedangkan *use case* menunjukkan bagaimana aktor menggunakan sistem untuk menjalankan aktivitas tersebut (Hasanah, 2020). Adapun simbol – simbol *Activity Diagram*.

Tabel 3. Simbol-Simbol Activity Diagram

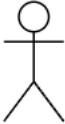
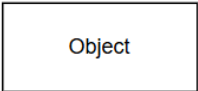
No	Gambar	Nama	Keterangan
1		<i>Initial State</i>	Tanda dimulainya aktivitas yang dilakukan oleh sistem.
2		<i>Final Node</i>	Tanda selesainya aktivitas yang dilakukan oleh sistem.

No	Gambar	Nama	Keterangan
			
3		<i>Activity</i>	Simbol aktivitas yang dilakukan oleh sistem dan dimulai dengan kata kerja.
4		<i>Decision</i>	Simbol percabangan dengan opsi lebih dari satu.
5		<i>Join</i>	Simbol cabang dengan opsi lebih dari satu

2.13.4 Sequence Diagram

Dalam pemodelan sistem, *sequence diagram* adalah alat penting yang digunakan untuk menunjukkan interaksi antara objek dalam urutan waktu tertentu. *Sequence diagram* digunakan untuk menunjukkan alur komunikasi antar objek dalam aplikasi berbasis data. Menurut jurnal (Kharisma Raharjana, 2015), pengembang dapat menggunakan teknik *reverse engineering* untuk membuat *sequence diagram* yang menunjukkan interaksi yang terjadi dalam aplikasi yang sudah ada, sehingga meningkatkan pemahaman tentang alur interaksi dan mendukung pengembangan perangkat lunak yang lebih terstruktur dan konsisten. Berikut merupakan tabel simbol-simbol dalam *sequence diagram*.

Tabel 4. Simbol *Sequence Diagram*

NO	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Entitas eksternal yang berinteraksi dengan sistem.
2.		<i>Object</i>	Entitas di dalam sistem yang melakukan tugas tertentu.
3.		<i>Message</i>	Sebuah aksi mengirimkan <i>request</i> .
4.		<i>Lifeline</i>	Sebuah titik waktu di mana elemen berinteraksi dengan <i>actor</i> atau <i>object</i> .
5.		<i>Action Bar</i>	Proses durasi pada aktivitas sebuah <i>actor</i> atau <i>object</i> .
6.		Reply Message	Sebuah aksi mengirim <i>request</i> .

III. METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian ini dilakukan pada semester ganjil tahun 2024/2025 di Laboratorium Perangkat Lunak yang terletak di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung yang terletak di Jalan Soemantri Brojonegoro Nomor 1, Gedong Meneng, Bandar Lampung.

Tabel 5. Waktu Penelitian

Aktivitas	2024								2025															
	Januari				Februari				Maret				April				Mei							
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4				
Planning																								
Sprint 1																								
Sprint 2																								
Sprint 3																								
Sprint 4																								
Sprint 5																								
Pengujian																								

3.2 Alat Penelitian

Dalam penelitian ini menggunakan dua jenis alat yaitu perangkat lunak dan perangkat keras, untuk mendukung pelaksanaan sistem dan pengembangannya. Berikut merupakan spesifikasi alat yang digunakan selama penelitian.

3.2.1 Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan selama penelitian yaitu:

Tabel 6. Perangkat Lunak Yang Digunakan

NO	Nama Perangkat Lunak	Deskripsi
1.	<i>Windows 11 Home Single Language</i>	Untuk <i>software</i> yang digunakan selama proses pengembangan aplikasi, digunakan sebagai platform utama untuk menjalankannya.
2.	<i>Visual Studio Code</i>	Digunakan untuk menulis, mengedit, dan melakukan debugging program, visual studio code adalah editor kode yang sederhana namun kuat.
3.	<i>Draw io</i>	Draw.io adalah aplikasi berbasis <i>web</i> yang dapat digunakan untuk membuat dan mengedit berbagai macam diagram, seperti diagram flowchart, diagram UML dan masih banyak lagi.
4.	<i>Figma</i>	Digunakan untuk mengembangkan, membuat prototipe, dan bekerja sama dengan pengembangan aplikasi atau <i>web</i> , terutama pada tahap perencanaan tampilan dan pengalaman pengguna.
5.	<i>Github</i>	<i>GitHub</i> adalah <i>platform</i> berbasis <i>Git</i> yang mendukung kolaborasi tim dan membantu mengelola versi proyek. Ini memungkinkan penyimpanan kode, pemantauan perubahan, dan kerja sama tim dalam satu repositori.
6.	<i>Publish or Perish 8</i>	Digunakan untuk mencari referensi jurnal yang digunakan untuk menganalisis data sitasi.

7.	<i>Mendeley</i>	Digunakan untuk menyimpan dan mengatur data sitasi serta dapat membantu menemukan sumber yang relevan.
8.	<i>Microsoft Word</i>	<i>Microsoft Word</i> adalah aplikasi yang dapat digunakan untuk membuat, mengedit, dan memformat dokumen teks. Selain itu, aplikasi ini dapat menyisipkan gambar, tabel, grafik, dan melakukan pengecekan ejaan dan tata bahasa, sehingga membuat dokumen menjadi lebih rapi dan profesional.

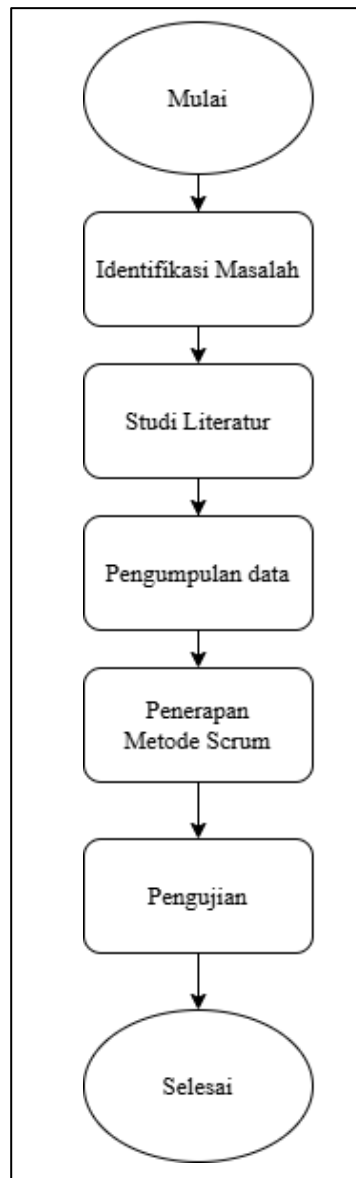
3.2.2 Perangkat keras (*Hardware*)

Perangkat keras yang digunakan selama penelitian yaitu:

- a. *System Manufacture* : Asus
- b. *System Model* : Vivobook Go E1404FA_E1404FA
- c. Processor : AMD Ryzen 3 7320U, 2.4 GHz, 4 Cores, 8 Threads
- d. CPU : AMD Ryzen 3 7320U
- e. RAM : 8 GB
- f. *System Type* : x64-based PC

3.3 Tahapan Penelitian

Berikut ini merupakan tahapan penelitian yang dilakukan pada sistem kinerja sebagai berikut :



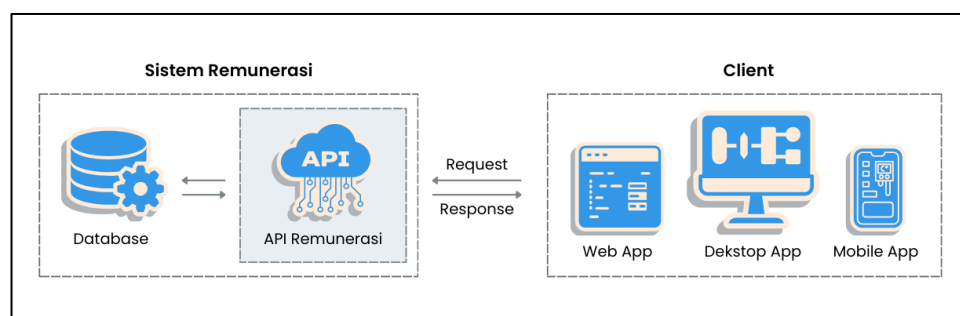
Gambar 2. Tahapan Penelitian

3.3.1 Identifikasi masalah

Pada tahap awal penelitian ini, dilakukan identifikasi terhadap masalah dan kebutuhan sistem di Universitas Lampung. Pada tahap ini, pengambilan keputusan dan pemecahan masalah terkait kebutuhan sistem dilakukan. Sangat penting untuk memahami komponen yang diperlukan untuk

membangun sistem yang lebih efisien dan efektif. Salah satu masalah utama dalam sistem saat ini adalah bahwa data kinerja dosen hanya dapat diakses oleh dosen yang bersangkutan. Ini memperlambat proses evaluasi, akreditasi, dan pengambilan keputusan.

Selain itu, tidak adanya integrasi dengan sistem lain yang relevan membatasi penggunaan data yang optimal, dan proses administratif membutuhkan waktu lebih lama ketika informasi diperlukan untuk penilaian kinerja atau akreditasi. Ini menunjukkan bahwa Universitas Lampung membutuhkan sebuah sistem yang tidak hanya responsif dan cepat dalam memberikan akses cepat ke data, tetapi juga mampu mengintegrasikan data kinerja dosen dengan sistem lain yang relevan. Oleh karena itu, arsitektur sistem ini dirancang untuk memastikan bahwa data kinerja dosen dapat diakses oleh pihak berwenang, seperti dekan dan ketua program studi, dengan tetap menjamin keamanan serta integritas data. Melalui sistem yang terintegrasi, proses evaluasi, akreditasi, dan pengambilan keputusan akademik diharapkan dapat berlangsung lebih efektif dan tepat waktu, sehingga berkontribusi pada peningkatan kualitas pendidikan di Universitas Lampung.



Gambar 3. Arsitektur Sistem

Dalam sistem ini, proses bisnis dimulai dengan pelaksanaan aktivitas Tri Dharma yaitu pendidikan, penelitian, pengabdian, dan kegiatan penunjang. Setelah semua aktivitas tersebut dicatat di sistem SISTER, dosen

mengunduh laporan BKD untuk melaporkan kinerjanya. Kemudian, untuk sinkronisasi dan mengunggah laporan BKD untuk mengajukan klaim kinerja tambahan, dosen masuk ke sistem kompensasi. Poin capaian, yang akan digunakan untuk menghitung kompensasi, akan diproses dan kemudian disimpan dalam database sistem.

Sistem yang dikembangkan dalam penelitian ini disesuaikan untuk peran pengguna seperti dosen, kaprodi, dan dekan. Antarmuka pada system ini lebih informatif dan mudah digunakan. Diharapkan bahwa visualisasi dan penyajian data yang terstruktur dan mudah dipahami akan memudahkan proses pengawasan kinerja, pembuatan dokumen akreditasi, dan evaluasi kinerja institusi. Selain itu, integrasi dengan API *backend* memastikan bahwa data selalu sinkron dan valid.

3.3.2 Studi Literatur

Pada tahapan studi literatur ini dilakukan untuk memahami bagaimana *framework* React JS bekerja dan mengidentifikasi permasalahan yang sering muncul dalam pengembangan *front-end* berbasis *web*. Studi ini juga mempelajari bagaimana menggunakan keunggulan React JS untuk membuat antarmuka yang interaktif dan responsif. Studi ini akan menjadi dasar untuk pengembangan sistem yang lebih efisien dan efektif karena selain mengidentifikasi masalah, juga menganalisis kendala umum dalam integrasi data dan aksesibilitas sistem.

3.3.3 Tahapan Pengumpulan Data

Dalam penelitian ini, Pengumpulan data dilakukan dengan mendapatkan informasi yang dibutuhkan dalam menganalisis kebutuhan dan merancang sistem. Data yang dikumpulkan dibagi menjadi dua jenis yaitu data primer dan data sekunder. Masing-masing digunakan untuk membantu proses analisis kebutuhan sistem agar lebih tepat sasaran.

1. Data Primer

Data primer didapatkan dengan cara melakukan wawancara dengan pihak yang terlibat langsung dalam penggunaan sistem, yaitu Ketua Program Studi. Tujuan dari wawancara ini adalah untuk mendapatkan pemahaman yang lebih baik tentang kebutuhan pengguna, tantangan yang dihadapi dalam pengelolaan data kinerja dosen, dan harapan terhadap fitur dan fungsi yang diharapkan dari sistem yang akan dibangun.

2. Data Sekunder

Data sekunder berasal dari sumber yang sudah ada sebelumnya, seperti dokumentasi dan data dari sistem remunerasi dosen Universitas Lampung. Informasi tentang kinerja dosen, terutama tentang kegiatan bidang pendidikan dan penelitian. Data dianalisis untuk mendapatkan pemahaman tentang cara sistem yang sudah ada berfungsi, pola penggunaan data, dan sejauh mana data dapat diintegrasikan untuk memenuhi persyaratan akreditasi program studi.

3.3.4 Penerapan Metode Scrum

Metode Scrum adalah salah satu pendekatan Agile yang digunakan untuk mengelola dan mengembangkan produk atau perangkat lunak yang sangat kompleks. Ini menawarkan kerangka kerja kolaboratif yang memungkinkan tim bekerja secara iteratif dan beradaptasi secara cepat dengan perubahan. Ini dilakukan dengan membagi pekerjaan dalam siklus pendek atau *sprint*.

Tabel 7. *Stakeholder*

No	Nama	Peran
1.	UPT TIK Unila	<i>Business Stakeholder</i>
2.	M. Iqbal Parabi, S.SI., M.T.	<i>Product Owner</i>
3.	Amalia Nurul Rahmawati	<i>Developer</i>
4.	Chandra Prasetya Putra	<i>Developer</i>
5.	Shafa Aulia	<i>Developer</i>
6.	Fathimah Abiyyi Khairunnisa	<i>Developer</i>

1. *Product Backlog*

Produk backlog disusun berdasarkan fitur-fitur yang telah direncanakan untuk dikembangkan di situs *web* remunerasi dosen. Fitur-fitur ini diprioritaskan dan disusun dalam bentuk daftar yang mencakup deskripsi untuk setiap tugas yang ditugaskan. *product backlog* dipaparkan dalam table berikut.

Tabel 8. *Product Backlog*

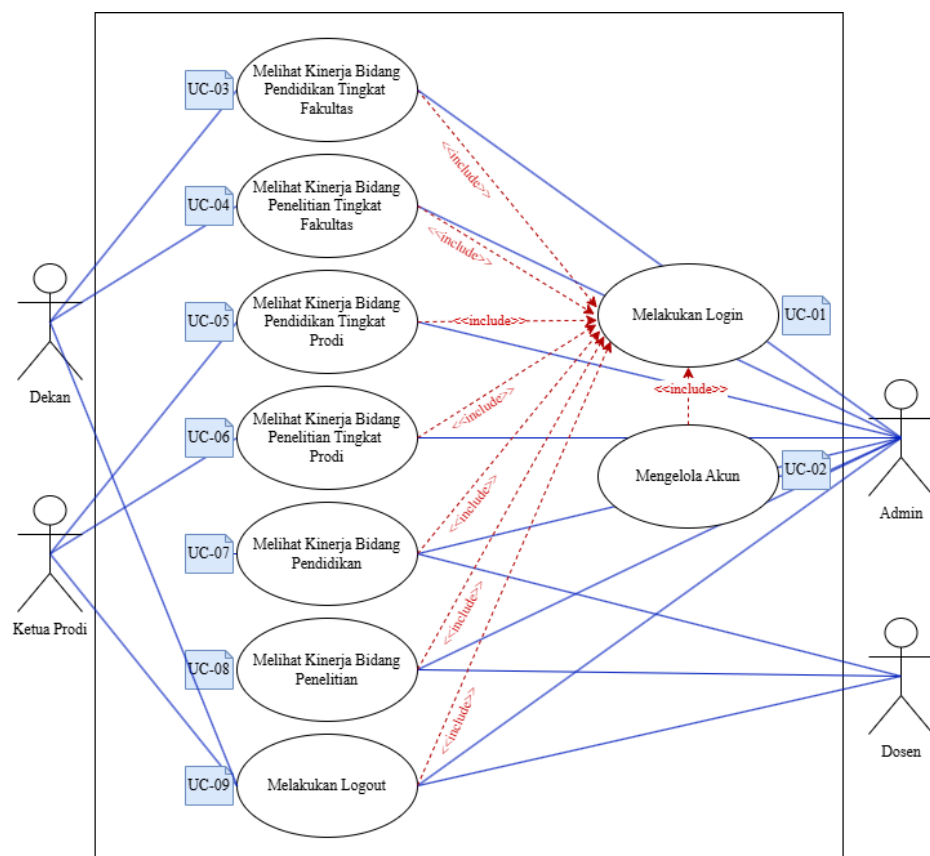
Kode	<i>Product Backlog</i>	Kepentingan	Perkiraan Waktu
PB-01	Modeling	Tinggi	8 Hari
PB-02	Pembuatan <i>Mockup</i>	Sedang	6 Hari
PB-03	<i>Login</i> dan <i>logout</i>	Tinggi	9 Hari
PB-04	Mengelola akun	Tinggi	7 Hari
PB-05	Tambah akun	Sedang	4 Hari
PB-06	Edit akun	Sedang	3 Hari
PB-07	Hapus akun	Sedang	3 Hari
PB-08	Profil	Tinggi	7 Hari
PB-09	Daftar program studi	Sedang	4 Hari
PB-10	Daftar dosen	Sedang	7 Hari
PB-11	Dasbor	Sedang	5 Hari
PB-12	Kinerja Fakultas	Sedang	3 Hari
PB-13	Kinerja Program Studi	Sedang	3 Hari
PB-14	Panduan & Bantuan	Rendah	1 Hari

2. Desain Sistem

Pada penelitian ini, desain bertujuan untuk memberikan Gambaran menyeluruh mengenai alur kerja sistem serta antarmuka pengguna yang akan dikembangkan.

a. Use Case

Berikut ini terdapat *use case* diagram yang digunakan untuk merancang *website* remunerasi dosen. Dalam *use case* diagram ini digambarkan alur interaksi antara empat actor yaitu Admin, Dekan, Ketua Prodi, dan Dosen.



Gambar 4. Use Case Diagram

Tabel 9. Penomoran *Use Case Diagram*

Kode	<i>Use Case Diagram</i>
UC-01	Melakukan Login
UC-02	Mengelola Akun
UC-03	Melihat Kinerja Bidang Pendidikan Tingkat f Fakultas
UC-04	Melihat Kinerja Bidang Penelitian Tingkat Fakultas
UC-05	Melihat kinerja bidang Pendidikan Tingkat prodi
UC-06	Melihat kinerja bidang penelitian Tingkat prodi
UC-07	Melihat Kinerja Bidang Pendidikan
UC-08	Melihat Kinerja Bidang Penelitian
UC-09	Melakukan <i>Logout</i>

Adapun *use case description* yang menjelaskan mengenai interaksi antara aktor dengan sistem dalam menjalankan suatu proses. *Use case description* ini mencakup alur kejadian, kondisi awal, kondisi akhir. Berikut ini *use case description* yang digunakan dalam penelitian ini.

1) *Use Case Description* Melakukan Login (UC-01)

Use case description melakukan login menunjukkan bagaimana pengguna dapat mengakses sistem dengan memasukkan kredensial yang benar.

Tabel 10. *Use Case Description* Melakukan Login

<i>Use Case Name</i>	: Melakukan Login
ID	: UC-01
<i>Priority</i>	: High
<i>Actor</i>	: Admin, Dekan, Kaprodi, Dosen
<i>Description</i>	: <i>Actor</i> dapat melakukan <i>login</i> ke sistem menggunakan NIDN dan <i>password</i> yang sudah terdaftar pada <i>database</i> .
<i>Trigger</i>	: <i>Actor</i> melakukan <i>login</i>

Preconditions	: <i>Actor</i> sudah terdaftar di sistem.
Normal Course	
1. <i>Actor</i> mengakses halaman login. 2. <i>Actor</i> memasukkan NIDN dan <i>password</i>. 3. Sistem meminta data pengguna. 4. <i>Database</i> mengirim data akun. 5. Sistem mencocokkan kredensial <i>actor</i>. 6. Jika <i>valid</i>, sistem mengirim token <i>authorization</i>. 7. <i>Virtual database</i> menyimpan token. 8. Sistem memvalidasi token. 9. Jika <i>valid</i>, sistem akan menampilkan halaman <i>user</i>.	
Postconditions	: <i>Actor</i> berhasil masuk dan dapat mengakses fitur-fitur sistem sesuai dengan perannya
Sub Flows	: <i>Forget password</i>
Alternate/Exceptional Flows:	
A1: Jika NIDN atau <i>password</i> salah, maka sistem akan menampilkan pesan error dan meminta <i>actor</i> untuk mencoba lagi. A2: Jika <i>actor</i> telah melewati batas waktu sesi tertentu, secara otomatis token akan berubah menjadi tidak aktif dan sesi akan berakhir. A3: Jika <i>actor</i> lupa <i>password</i>, sistem menampilkan pesan untuk meminta <i>actor</i> menghubungi admin.	

2) Use Case Description Mengelola Akun (UC-02)

Use case description mengelola akun menjelaskan bagaimana admin dapat menambah, mengedit, dan menghapus akun pengguna setelah berhasil login. Sebelum disimpan ke database, sistem akan memverifikasi setiap tindakan sebelum menampilkan data terbaru. Jika terjadi kesalahan seperti format data yang salah atau proses yang dibatalkan, sistem akan menampilkan pesan error atau membatalkan perubahan sesuai kondisi yang terjadi.

Tabel 11. *Use Case Description* Mengelola Akun

Use Case Name	: Mengelola Akun Pengguna
ID	: UC-02
Priority	: High
Actor	: Admin
Description	: <i>Actor</i> dapat melakukan kelola akun pengguna seperti menambah akun baru, mengubah data akun, atau menghapus akun pengguna.
Trigger	: <i>Actor</i> memilih salah satu aksi kelola akun
Preconditions	: <i>Actor</i> sudah login sebagai Admin dan memiliki status sesi yang aktif
Normal Course	
1. <i>Actor</i> sudah berada dalam sistem dan memiliki status sesi yang aktif. 2. Sistem menampilkan daftar berisi data akun pengguna. 3. <i>Actor</i> dapat memilih salah satu aksi Kelola akun. 1. sistem menampilkan formulir tambah akun 2. <i>Actor</i> mengisi form berisi data yang dibutuhkan untuk menambahkan akun. 3. <i>Actor</i> melakukan submit form. 4. Sistem mengirimkan request berisi data akun baru yang diinputkan actor. 5. Database menyimpan dan mengembalikan response daftar akun yang baru. 4. <i>Actor</i> memilih aksi mengubah data akun. 1. sistem menampilkan formulir ubah data akun. 2. <i>Actor</i> Mengubah form berisi data sebelumnya menjadi data yang ingin diubah. 3. <i>Actor</i> submit form. 4. Sistem mengirimkan request berisi data akun baru yang diubah actor. 5. Database menyimpan dan mengembalikan response data akun yang baru. 5. <i>Actor</i> memilih aksi menghapus data akun.	

	1. Actor memvalidasi pilihan hapus akun.
	2. Sistem mengirimkan request hapus akun.
	3. Database menghapus akun dan mengembalikan response.
	6. Sistem menampilkan seluruh daftar akun pengguna.
Postconditions	: Actor dapat menambah, mengubah, dan menghapus akun pengguna.
Sub Flows	: Actor gagal Menambah Akun, Mengedit Data Akun, Menghapus Akun
Alternate/Exceptional Flows:	
	A1: Jika data pengguna yang dimasukkan tidak valid, maka sistem akan menampilkan pesan error dan meminta actor untuk memperbaiki data.
	A2: Jika data pengguna yang ingin dihapus dibatalkan, sistem akan mengembalikan ke halaman daftar pengguna tanpa perubahan.

3) Use Case Description Melihat Kinerja Bidang Pendidikan Tingkat Fakultas (UC-03)

Use case description melihat kinerja dosen bidang pendidikan tingkat fakultas mencakup aktivitas pengguna dalam memperoleh data kinerja dosen di bidang pendidikan pada lingkup fakultas.

Tabel 12. *Use Case Description* Melihat Kinerja Bidang Pendidikan Tingkat Fakultas

Use Case Name	: Melihat Kinerja Bidang Pendidikan Tingkat Fakultas
ID	: UC-03
Priority	: <i>Medium</i>
Actor	: Admin, Dekan
Description	: Actor mengakses fitur melihat kinerja bidang

	pendidikan tingkat fakultas berdasarkan program studi dan dosen yang dipilih.
Trigger	: <i>Actor</i> memilih fitur melihat kinerja bidang pendidikan tingkat fakultas.
Preconditions	: <i>Actor</i> sudah login sebagai Dekan dan memiliki status sesi yang aktif.
Normal Course	
1. <i>Actor</i> sudah berada dalam sistem dan memiliki status sesi yang aktif. 2. Sistem menampilkan data fakultas beserta daftar program studi. 3. <i>Actor</i> memilih program studi. 4. Sistem meminta data program studi. 5. <i>Database</i> mengembalikan data program studi. 6. Sistem menampilkan data program studi dan daftar dosen. 7. <i>Actor</i> memilih dosen yang ingin dilihat.	
Postconditions	: <i>Actor</i> berhasil memilih program studi dan dosen serta mendapatkan data kinerja bidang pendidikan berdasarkan pilihan tersebut.
Sub Flows	: <i>Actor</i> dapat melakukan filter data kinerja bidang pendidikan berdasarkan tahun.
Alternate/Exceptional Flows:	
A1: Jika data dosen tidak ditemukan, maka sistem akan menampilkan pesan “Data tidak ditemukan”.	
A2: Jika actor melakukan filter data kinerja bidang pendidikan berdasarkan tahun, namun data kinerja tidak ada, maka sistem akan menampilkan pesan “Data tidak ditemukan”.	

4) *Use Case Description* Melihat Kinerja Bidang Penelitian Tingkat Fakultas (UC-04)

Use case description melihat kinerja dosen bidang penelitian tingkat fakultas mencakup proses memperoleh informasi kinerja dosen dalam bidang penelitian di lingkungan fakultas. Hal ini hanya dapat dilakukan setelah aktor berhasil login dan memiliki sesi aktif di sistem. Untuk melihat data kinerja, aktor dapat memilih program studi dan nama dosen dari daftar yang tersedia. Sistem kemudian akan mengambil data dari database dan menampilkannya dalam format yang rapi dan terstruktur.

Tabel 13. *Use Case Description* Melihat Kinerja Bidang Penelitian Tingkat Fakultas

<i>Use Case Name</i>	: Melihat Kinerja Bidang Penelitian Tingkat Fakultas
ID	: UC-04
<i>Priority</i>	: <i>Medium</i>
<i>Actor</i>	: Admin, Dekan
<i>Description</i>	: <i>Actor</i> mengakses fitur melihat kinerja bidang penelitian tingkat fakultas berdasarkan program studi dan dosen yang dipilih.
<i>Trigger</i>	: <i>Actor</i> memilih fitur melihat kinerja bidang penelitian tingkat fakultas.
<i>Preconditions</i>	: <i>Actor</i> sudah login sebagai Dekan dan memiliki status sesi yang aktif.
<i>Normal Course</i>	
1. <i>Actor</i> sudah berada dalam sistem dan memiliki status sesi yang aktif. 2. Sistem menampilkan data fakultas beserta daftar program studi. 3. <i>Actor</i> memilih program studi. 4. Sistem meminta data program studi.	

-
5. *Database* mengembalikan data program studi.
 6. Sistem menampilkan data program studi dan daftar dosen.
 7. *Actor* memilih dosen yang ingin dilihat.
 8. Sistem meminta data dosen.
 9. *Database* mengembalikan data dosen.
 10. Sistem menampilkan data dosen dan bidang kinerja.
 11. *Actor* memilih kinerja bidang penelitian.
 12. Sistem meminta data bidang penelitian.
 13. *Database* mengembalikan data bidang penelitian.
 14. Sistem menampilkan data bidang penelitian.
-

Postconditions : *Actor* berhasil memilih program studi dan dosen serta mendapatkan data kinerja bidang penelitian berdasarkan pilihan tersebut.

Sub Flows : *Actor* dapat melakukan filter data kinerja bidang penelitian berdasarkan tahun.

Alternate/Exceptional Flows:

A1: Jika data dosen tidak ditemukan, maka sistem akan menampilkan pesan “Data tidak ditemukan”.

A2: Jika actor melakukan filter data kinerja bidang penelitian berdasarkan tahun, namun data kinerja tidak ada, maka sistem akan menampilkan pesan “Data tidak ditemukan”.

5) *Use Case Description* Melihat Kinerja Bidang Pendidikan Tingkat Program Studi (UC-05)

Use case description melihat kinerja dosen bidang pendidikan tingkat program studi menggambarkan proses akses data kinerja dosen dalam bidang pendidikan pada tingkat program studi. Pada use case description ini admin dan kaprodi dapat mengakses data kinerja dosen pada tingkat program studi. Akses ini hanya dapat dilakukan setelah pengguna memiliki sesi aktif di sistem dan telah login. Sistem akan menampilkan data program studi dan daftar dosen setelah masuk. Pengguna dapat memilih salah satu

dosen untuk melihat detail kinerjanya di bidang pendidikan. Data akan diambil dari database dan ditampilkan dalam format yang terstruktur dan mudah dipahami oleh sistem.

Tabel 14. *Use Case Description* Melihat Kinerja Bidang Pendidikan Tingkat Program Studi

<i>Use Case Name</i>	: Melihat Kinerja Bidang Pendidikan Tingkat Program Studi
<i>ID</i>	: UC-05
<i>Priority</i>	: Medium
<i>Actor</i>	: Admin, Kaprodi (Ketua Program Studi)
<i>Description</i>	: Actor mengakses fitur melihat kinerja bidang pendidikan tingkat program studi berdasarkan dosen yang dipilih.
<i>Trigger</i>	: Actor memilih fitur melihat kinerja bidang pendidikan tingkat program studi.
<i>Preconditions</i>	: Actor sudah login sebagai Kaprodi dan memiliki status sesi yang aktif.
<i>Normal Course</i>	
1. Actor sudah berada dalam sistem dan memiliki status sesi yang aktif. 2. Sistem menampilkan data program studi beserta daftar dosen. 3. Actor memilih dosen yang ingin dilihat. 4. Sistem meminta data dosen. 5. Database mengembalikan data dosen. 6. Sistem menampilkan data dosen dan bidang kinerja. 7. Actor memilih kinerja bidang pendidikan. 8. Sistem meminta data bidang pendidikan. 9. Database mengembalikan data bidang pendidikan. 10. Sistem menampilkan data bidang pendidikan.	
<i>Postconditions</i>	: Actor berhasil memilih dosen serta mendapatkan data kinerja bidang pendidikan berdasarkan pilihan tersebut.

Sub Flows : *Actor* dapat melakukan filter data kinerja bidang pendidikan berdasarkan tahun.

Alternate/Exceptional Flows:

A1: Jika data dosen tidak ditemukan, maka sistem akan menampilkan pesan “Data tidak ditemukan”.

A2: Jika actor melakukan filter data kinerja bidang pendidikan berdasarkan tahun, namun data kinerja tidak ada, maka sistem akan menampilkan pesan “Data tidak ditemukan”.

6) **Use Case Description Melihat Kinerja Bidang Penelitian Tingkat Program Studi (UC-06)**

Use case description melihat kinerja dosen bidang penelitian tingkat program studi mencakup proses memperoleh data kinerja dosen di bidang penelitian pada lingkungan program studi. Pada use case description ini admin dan kaprodi dapat mengakses data kinerja dosen pada tingkat program studi. Akses ini hanya dapat dilakukan setelah pengguna memiliki sesi aktif di sistem dan telah login. Sistem akan menampilkan data program studi dan daftar dosen setelah masuk. Pengguna dapat memilih salah satu dosen untuk melihat detail kinerjanya di bidang penelitian. Data akan diambil dari database dan ditampilkan dalam format yang terstruktur dan mudah dipahami oleh sistem.

Tabel 15. *Use Case Description* Melihat Kinerja Bidang Penelitian Tingkat Prodi

Use Case Name	: Melihat Kinerja Bidang Penelitian Tingkat Program Studi
ID	: UC-06
Priority	: Medium
Actor	: Admin, Kaprodi (Ketua Program Studi)

Description	: <i>Actor</i> mengakses fitur melihat kinerja bidang penelitian tingkat program studi berdasarkan dosen yang dipilih.
Trigger	: <i>Actor</i> memilih fitur melihat kinerja bidang penelitian tingkat program studi.
Preconditions	: <i>Actor</i> sudah login sebagai Kaprodi dan memiliki status sesi yang aktif.
Normal Course	
1. <i>Actor</i> sudah berada dalam sistem dan memiliki status sesi yang aktif. 2. Sistem menampilkan data program studi beserta daftar dosen. 3. <i>Actor</i> memilih dosen yang ingin dilihat. 4. Sistem meminta data dosen. 5. <i>Database</i> mengembalikan data dosen. 6. Sistem menampilkan data dosen dan bidang kinerja. 7. <i>Actor</i> memilih kinerja bidang penelitian. 8. Sistem meminta data bidang penelitian. 9. <i>Database</i> mengembalikan data bidang penelitian. 10. Sistem menampilkan data bidang penelitian.	
Postconditions	: <i>Actor</i> berhasil memilih dosen serta mendapatkan data kinerja bidang penelitian berdasarkan pilihan tersebut.
Sub Flows	: <i>Actor</i> dapat melakukan filter data kinerja bidang penelitian berdasarkan tahun.
Alternate/Exceptional Flows:	
A1: Jika data dosen tidak ditemukan, maka sistem akan menampilkan pesan “Data tidak ditemukan”.	
A2: Jika actor melakukan filter data kinerja bidang penelitian berdasarkan tahun, namun data kinerja tidak ada, maka sistem akan menampilkan pesan “Data tidak ditemukan”.	

7) *Use Case Description* Melihat Kinerja Bidang Pendidikan (UC-07)

Use case description melihat kinerja dosen bidang pendidikan menjelaskan proses pengguna dalam mengakses data kinerja dosen di bidang pendidikan. Pada *use case description* ini admin dan dosen dapat mengakses data kinerja dosen bidang pendidikan. Setelah pengguna melakukan login dan berhasil masuk, sistem akan menampilkan data kinerja bidang pendidikan tersebut.

Tabel 16. *Use Case Description* Melihat Kinerja Bidang Pendidikan

<i>Use Case Name</i>	: Melihat Kinerja Bidang Pendidikan
ID	: UC-07
<i>Priority</i>	: High
<i>Actor</i>	: Admin, Dosen
<i>Description</i>	: <i>Actor</i> mengakses fitur melihat kinerja bidang pendidikan yang hanya menampilkan data miliknya sendiri.
<i>Trigger</i>	: <i>Actor</i> memilih fitur melihat kinerja bidang pendidikan.
<i>Preconditions</i>	: <i>Actor</i> sudah login sebagai Dosen dan memiliki status sesi yang aktif.
<i>Normal Course</i>	
1. <i>Actor</i> sudah berada dalam sistem dan memiliki status sesi yang aktif. 2. Sistem menampilkan data dosen beserta kinerjanya. 3. <i>Actor</i> memilih kinerja bidang pendidikan. 4. Sistem meminta data bidang pendidikan. 5. <i>Database</i> mengembalikan data bidang pendidikan. 6. Sistem menampilkan data bidang pendidikan	
<i>Postconditions</i>	: <i>Actor</i> berhasil mendapatkan data kinerja bidang pendidikan.
<i>Sub Flows</i>	: <i>Actor</i> dapat melakukan filter data kinerja bidang pendidikan berdasarkan tahun.
<i>Alternate/Exceptional Flows:</i>	

A1: Jika actor melakukan filter data kinerja bidang pendidikan berdasarkan tahun, namun data kinerja tidak ada, maka sistem akan menampilkan pesan “Data tidak ditemukan”.

8) *Use Case Description* Melihat Kinerja Bidang Penelitian (UC-08)

Use case description melihat kinerja dosen bidang penelitian mencakup proses memperoleh informasi kinerja dosen dalam lingkup penelitian. Pada *use case description* ini admin dan dosen dapat mengakses data kinerja dosen bidang penelitian. Setelah pengguna melakukan login dan berhasil masuk, sistem akan menampilkan data kinerja bidang penelitian tersebut.

Tabel 17. *Use Case Description* Melihat Kinerja Bidang Penelitian

<i>Use Case Name</i>	: Melihat Kinerja Bidang Penelitia
<i>ID</i>	: UC-08
<i>Priority</i>	: High
<i>Actor</i>	: Dosen
<i>Description</i>	: Actor mengakses fitur melihat kinerja bidang penelitian yang hanya menampilkan data miliknya sendiri.
<i>Trigger</i>	: Actor memilih fitur melihat kinerja bidang penelitian.
<i>Preconditions</i>	: Actor sudah login sebagai Dosen dan memiliki status sesi yang aktif.
<i>Normal Course</i>	
1. <i>Actor</i> sudah berada dalam sistem dan memiliki status sesi yang aktif.	

2.	Sistem menampilkan data dosen beserta kinerjanya.
3.	<i>Actor</i> memilih kinerja bidang penelitian.
4.	Sistem meminta data bidang penelitian.
5.	<i>Database</i> mengembalikan data bidang penelitian.
6.	Sistem menampilkan data bidang penelitian
Postconditions	: <i>Actor</i> berhasil mendapatkan data kinerja bidang penelitian.
Sub Flows	: <i>Actor</i> dapat melakukan filter data kinerja bidang penelitian berdasarkan tahun.
Alternate/Exceptional Flows:	
A1: Jika actor melakukan filter data kinerja bidang penelitian berdasarkan tahun, namun data kinerja tidak ada, maka sistem akan menampilkan pesan “Data tidak ditemukan”.	

9) Use Case Description Melakukan Logout

Use case description melakukan *logout* menjelaskan langkah pengguna dalam keluar dari sistem guna menyudahi sesi yang sedang berlangsung. Ketika pengguna memilih untuk *logout*, sistem akan menangani permintaan dengan mengubah status sesi di database menjadi tidak aktif, dan kemudian mengarahkan pengguna ke halaman login sebagai tanda bahwa sesi sudah berakhir. Selain itu, ada situasi lain, seperti sesi otomatis berakhir karena waktu habis atau saat pengguna *logout* dari satu perangkat ke perangkat lain yang sebelumnya digunakan untuk login juga otomatis mengakhiri sesi.

Tabel 18. *Use Case Description* Melakukan Logout

Use Case Name	: Melakukan Logout
ID	: UC-015
Priority	: Medium
Actor	: Admin, Dekan, Kaprodi, Dosen

Description	: <i>Actor</i> melakukan <i>logout</i> untuk keluar dari sistem.
Trigger	: <i>Actor</i> melakukan <i>logout</i> .
Preconditions	: <i>Actor</i> sudah login dan memiliki status sesi yang aktif.
Normal Course	
1. <i>Actor</i> sudah berada dalam sistem dan memiliki status sesi yang aktif. 2. <i>Actor</i> memilih menu <i>logout</i>. 3. Sistem memproses permintaan <i>logout</i>. 4. <i>Database</i> memperbarui status sesi menjadi nonaktif. 5. Sistem akan mengarahkan <i>actor</i> ke halaman <i>login</i>.	
Postconditions	: <i>Actor</i> berhasil <i>logout</i> dan keluar dari sistem.
Sub Flows	: <i>Status sesi actor</i> berubah menjadi nonaktif.
Alternate/Exceptional Flows:	
A1: Jika waktu sesi habis, maka sistem akan mengakhiri sesi secara otomatis.	
A2: Jika <i>actor</i> <i>logout</i> dari satu perangkat tetapi masih <i>login</i> di perangkat lain, maka sistem akan secara otomatis mengakhiri sesi di semua perangkat.	

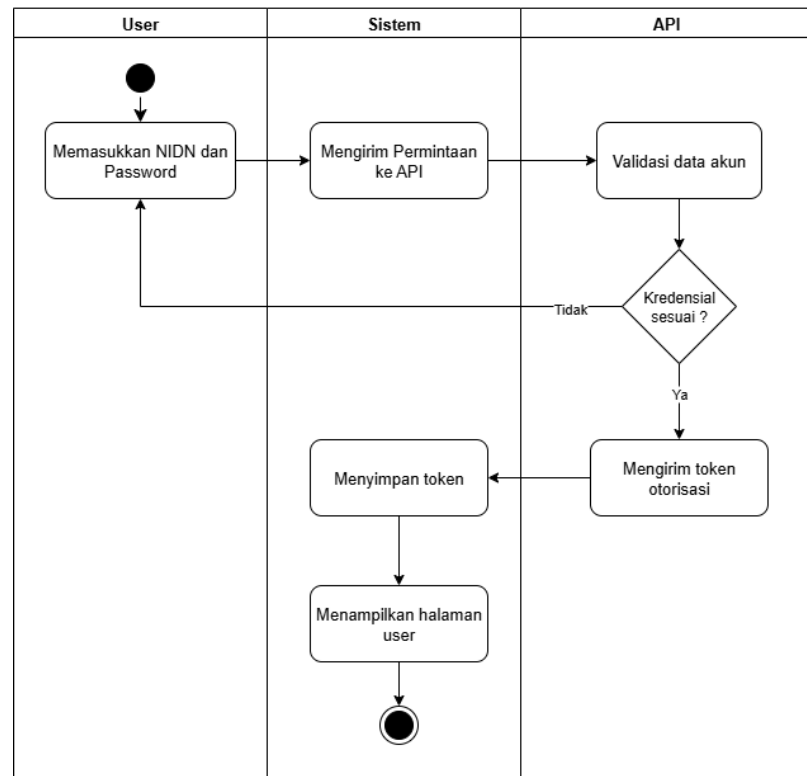
b. Activity Diagram

Berdasarkan *use case diagram* yang telah dibuat sebelumnya, Terdapat *activity diagram* yang digunakan untuk menjelaskan alur pada fitur tersebut.

1) Activity Diagram Melakukan Login (UC-01)

Activity diagram tersebut menunjukkan proses *login* pada sistem. Proses dimulai ketika user memasukkan NIDN dan password, lalu sistem mengirimkan permintaan untuk memeriksa data akun di database. Database mengambil data

yang relevan dan mengembalikannya ke sistem untuk validasi. Jika ID dan password sesuai, halaman user ditampilkan, sedangkan jika tidak sesuai, user diminta untuk mengulangi proses *login*.



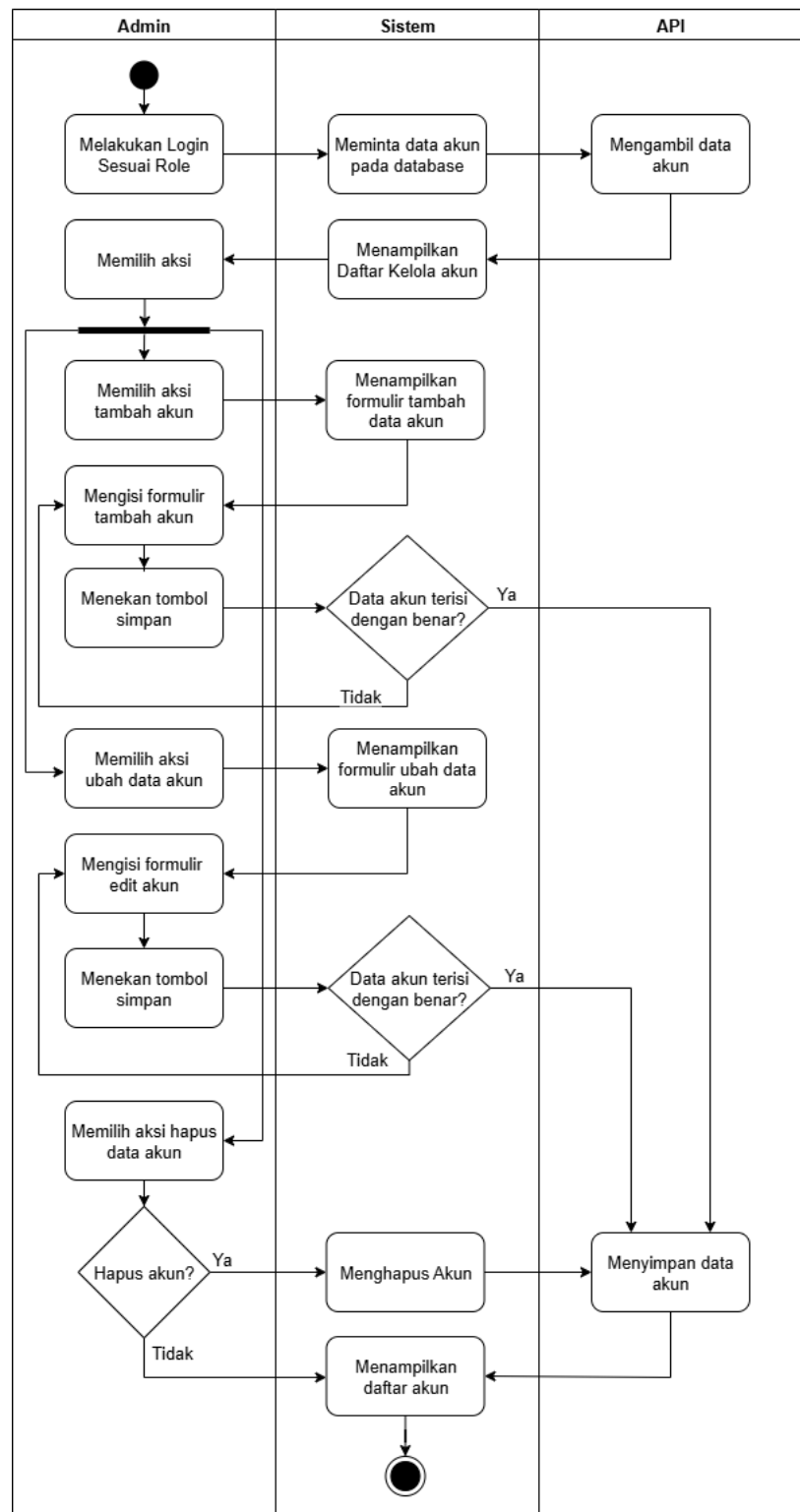
Gambar 5. *Activity Diagram Login*

2) *Activity Diagram* Mengelola Akun Pengguna (UC-02)

Activity diagram berikut menjelaskan proses pengelolaan data akun admin yang dapat menambah, mengubah, dan menghapus data akun. Proses dimulai ketika admin memilih menu akun, serta sistem mengambil dan menampilkan data akun dari database. Untuk menambah akun, admin memilih menu tambah, mengisi formulir tambah data akun, dan menekan tombol simpan. Jika data diisi dengan benar, data baru disimpan ke database. Untuk mengubah akun, admin memilih menu ubah, dan mengisi formulir ubah data akun,

dan menekan. Jika data telah diperbarui dan valid, data tersebut akan disimpan. Untuk menghapus akun, admin memilih menu hapus akun, dan sistem memverifikasi keputusan tersebut. Data akun dihapus dari database jika admin mengonfirmasi penghapusan. Daftar akun terbaru ditampilkan saat proses diakhiri.

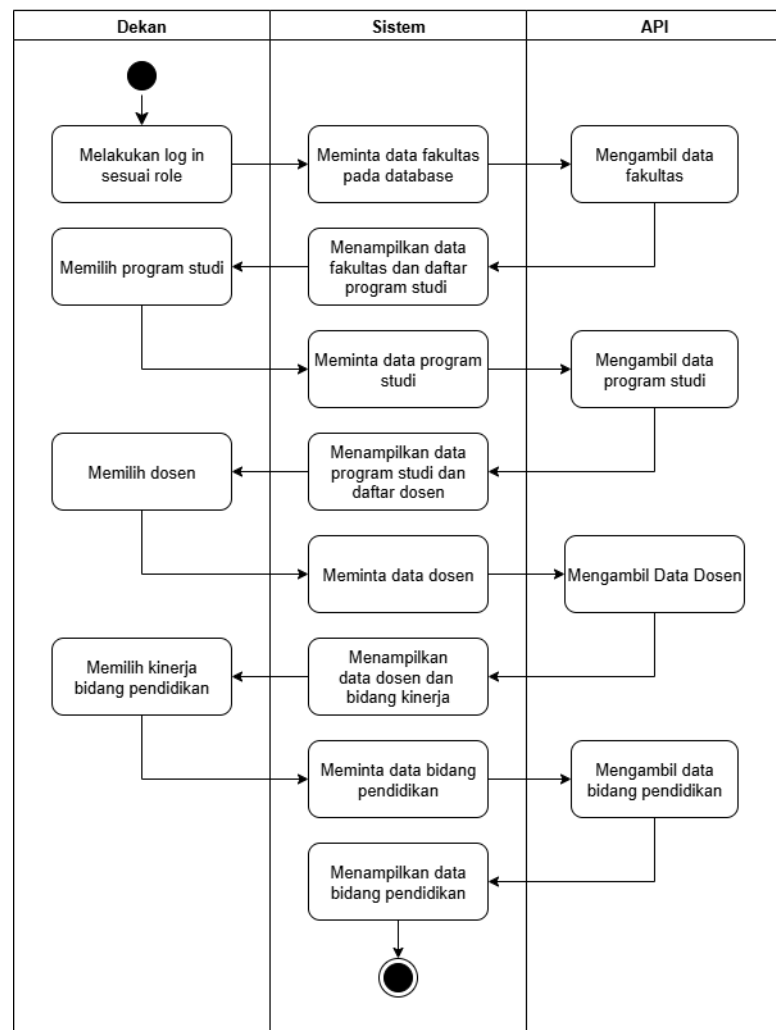
Activity Diagram Mengelola Akun



Gambar 6. Activity Diagram Mengelola Akun

3) Activity Diagram Melihat Kinerja Bidang Pendidikan Tingkat Fakultas (UC-03)

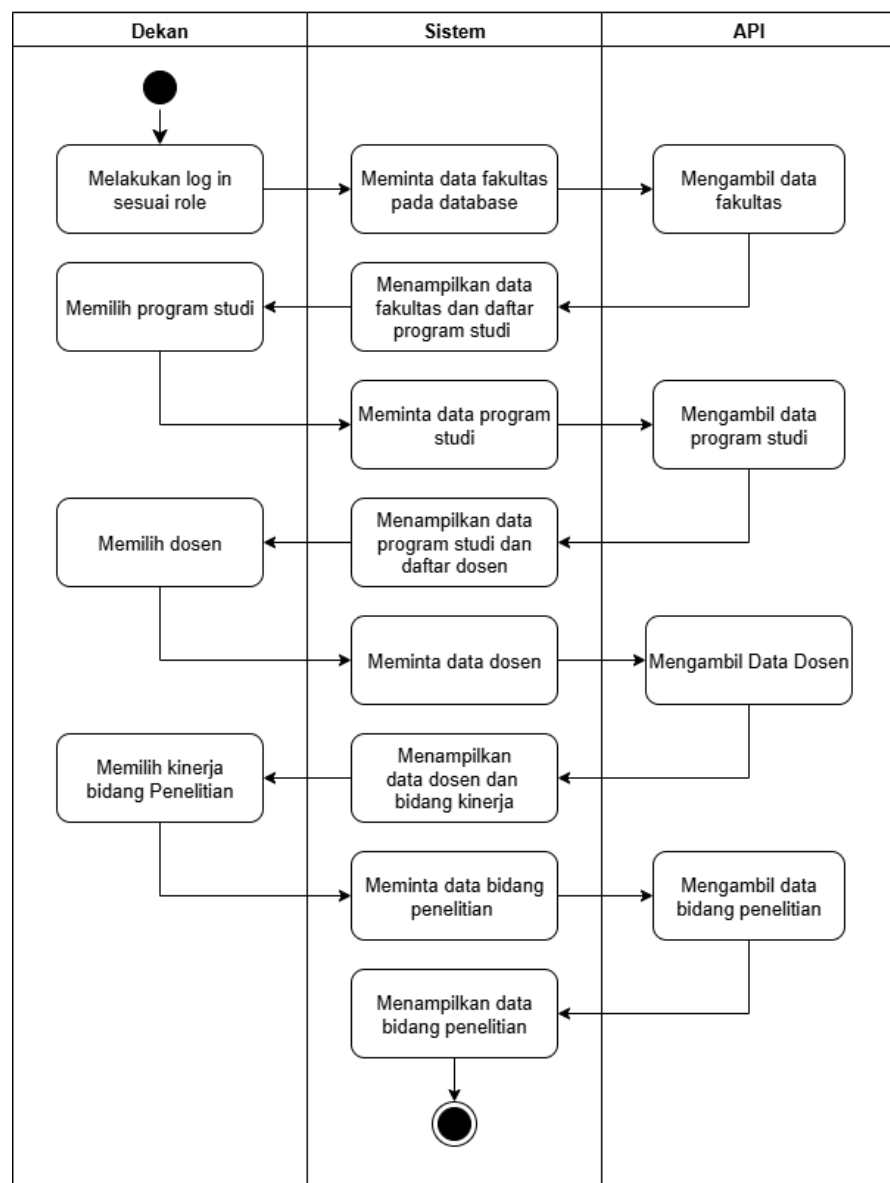
Activity diagram berikut merupakan alur dari kinerja bidang pendidikan di tingkat fakultas. Dimana dekan melakukan login yang telah disediakan oleh sistem, kemudian meminta data program studi dari database dan memilih program studi yang ingin dilihat. Selanjutnya, dekan memilih data dosen dan meminta data bidang kinerja dari database. Kemudian, sistem menampilkan data bidang pendidikan.



Gambar 7. Activity Diagram Melihat Kinerja Bidang Pendidikan Tingkat Fakultas

4) *Activity Diagram* Melihat Kinerja Bidang Penelitian Tingkat Fakultas (UC-04)

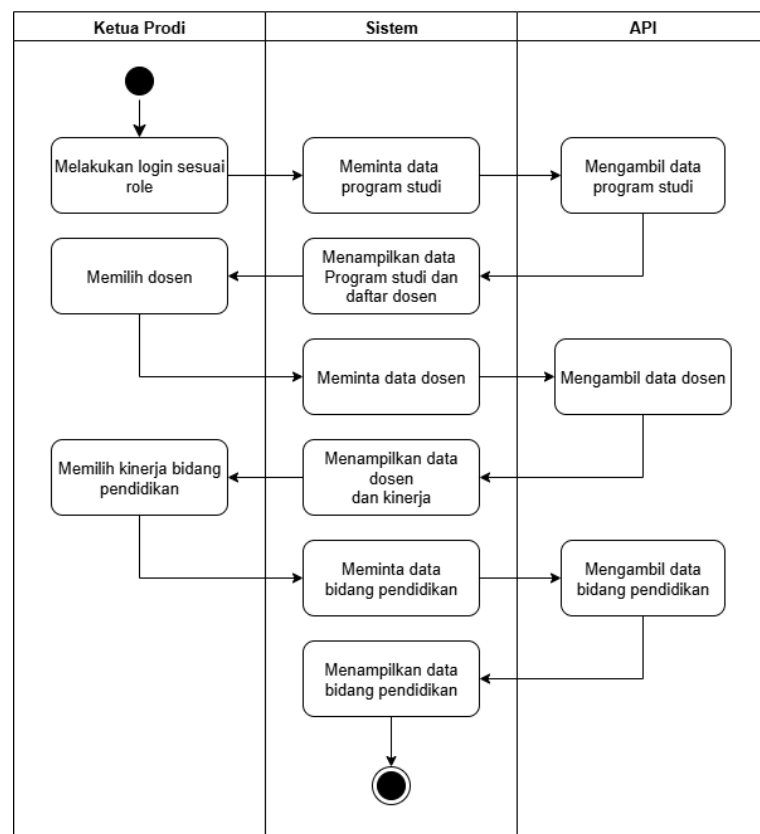
Activity diagram berikut merupakan alur dari kinerja bidang penelitian di tingkat fakultas. Dimana dekan melakukan *login* yang telah disediakan oleh sistem, kemudian meminta data program studi dari database dan memilih program studi yang ingin dilihat. Selanjutnya, dekan memilih data dosen dan meminta data bidang kinerja dari database. Kemudian, sistem menampilkan data bidang penelitian.



Gambar 8. Activity Diagram Melihat Kinerja Bidang Penelitian Tingkat Fakultas

5) Activity Diagram Melihat Kinerja Bidang Pendidikan Tingkat Program Studi (UC-05)

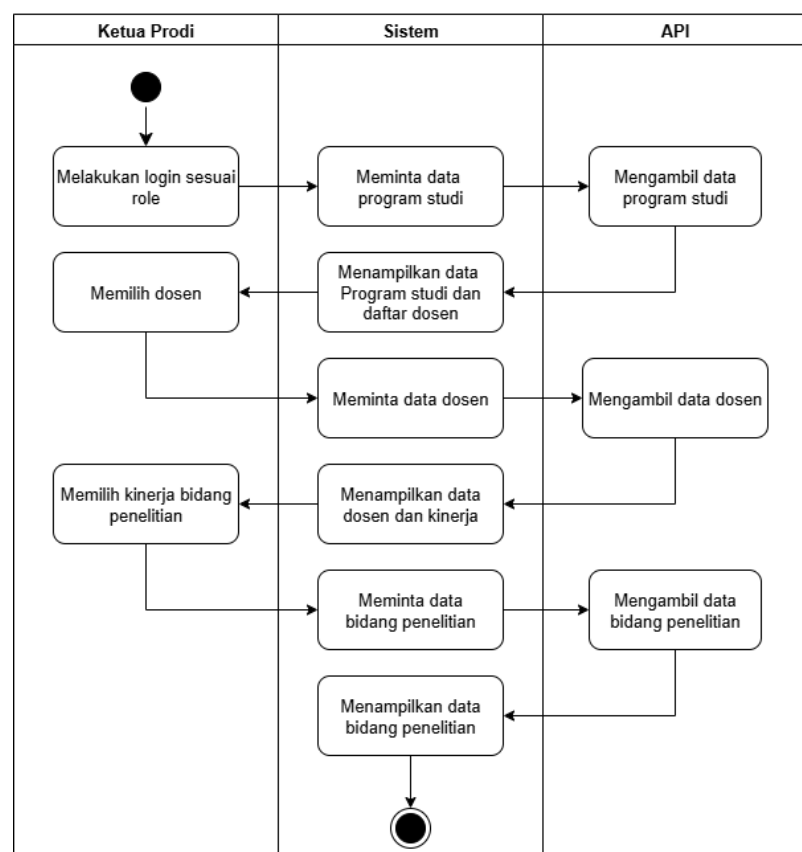
Activity diagram berikut menggambarkan alur aktivitas untuk melihat kinerja bidang pendidikan tingkat program studi. Untuk melihat kinerja bidang pendidikan tingkat program studi, ketua prodi harus melakukan *login* yang disediakan oleh sistem. Selanjutnya, sistem mengambil data dosen dari database dan menampilkan daftar dosen. kemudian, ketua prodi dapat memilih data dosen dan bidang kinerja yang ingin dilihat. Sistem mengambil data bidang pendidikan dari database dan menampilkannya.



Gambar 9. Activity Diagram Melihat Kinerja Bidang Pendidikan Tingkat Prodi

6) Activity Diagram Melihat Kinerja Bidang Penelitian Tingkat Program Studi (UC-06)

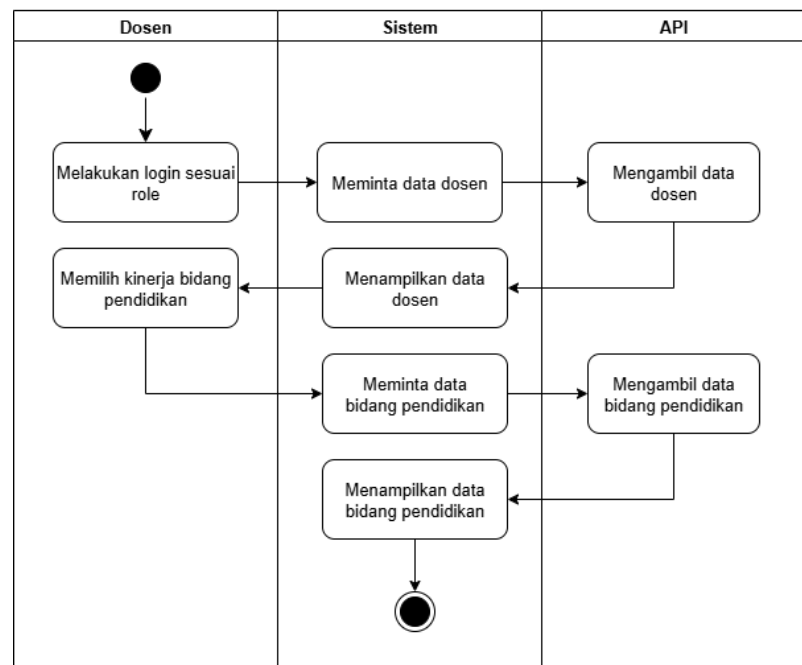
Activity diagram berikut menggambarkan alur aktivitas untuk melihat kinerja bidang penelitian tingkat program studi. Dimulai dengan melakukan *login* yang telah di sediakan pada sistem. Setelah itu, sistem akan mengambil data dosen dari database dan menampilkan daftar data dosen. Kemudian, pengguna dapat memilih data dosen dan bidang kinerja yang ingin mereka lihat. Selanjutnya, sistem akan mengambil data bidang penelitian dari database dan menampilkannya kepada pengguna.



Gambar 10. Activity Diagram Melihat Kinerja Bidang Penelitian Tingkat Prodi

7) *Activity Diagram* Melihat Kinerja Bidang Pendidikan (UC-07)

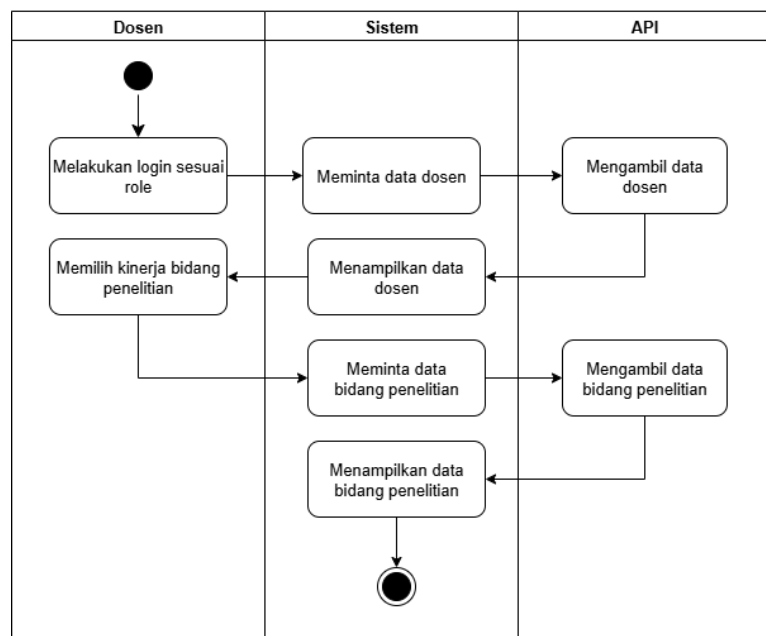
Pada *activity diagram* ini menggabungkan alur untuk melihat kinerja bidang pendidikan. Dimulai dengan dosen melakukan *login*. Kemudian, sistem meminta data bidang pendidikan dari database. Database kemudian mengambil dan mengirimkan data bidang pendidikan kembali ke sistem. Setelah data dikembalikan ke sistem, proses selesai.



Gambar 11. Activity Diagram Melihat Kinerja Bidang Pendidikan

8) *Activity Diagram* Melihat Kinerja Bidang Penelitian (UC-08)

Pada *activity diagram* ini menggabungkan alur untuk melihat kinerja bidang penelitian. Ini dimulai dengan dosen melakukan *login*. Kemudian, sistem meminta data bidang penelitian dari database. Database kemudian mengambil data tersebut dan menampilkannya pada sistem, yang kemudian ditunjukkan kepada Dosen.

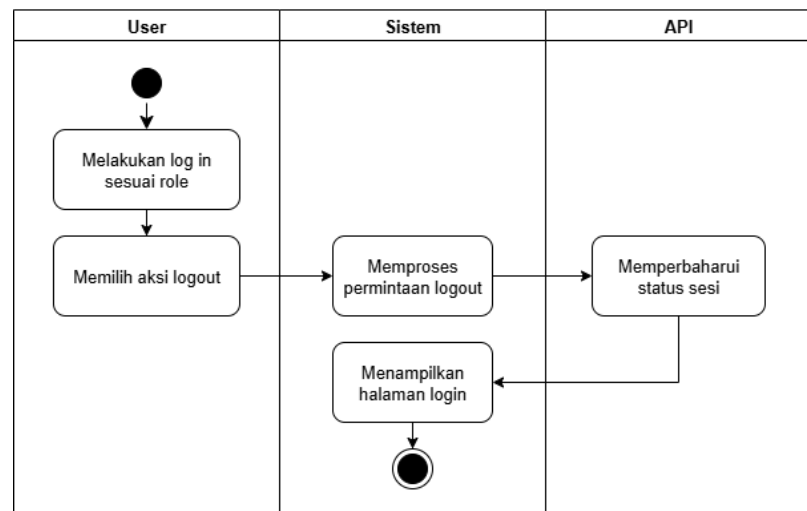


Gambar 12. *Activity Diagram* Melihat Kinerja Bidang Penelitian

9) *Activity Diagram* Melakukan Logout (UC-09)

Activity diagram proses *logout* menunjukkan alur yang terjadi ketika pengguna memilih untuk keluar dari sistem. Proses dimulai ketika pengguna menekan tombol *logout*, dan sistem menerima permintaan tersebut dan memprosesnya dengan menghapus sesi. Setelah itu, sistem mengarahkan

pengguna ke halaman *login* atau halaman utama aplikasi untuk memastikan bahwa mereka tidak dapat mengakses konten tanpa *login* kembali. Proses berakhir ketika pengguna berhasil keluar dan diarahkan ke halaman utama *website*.



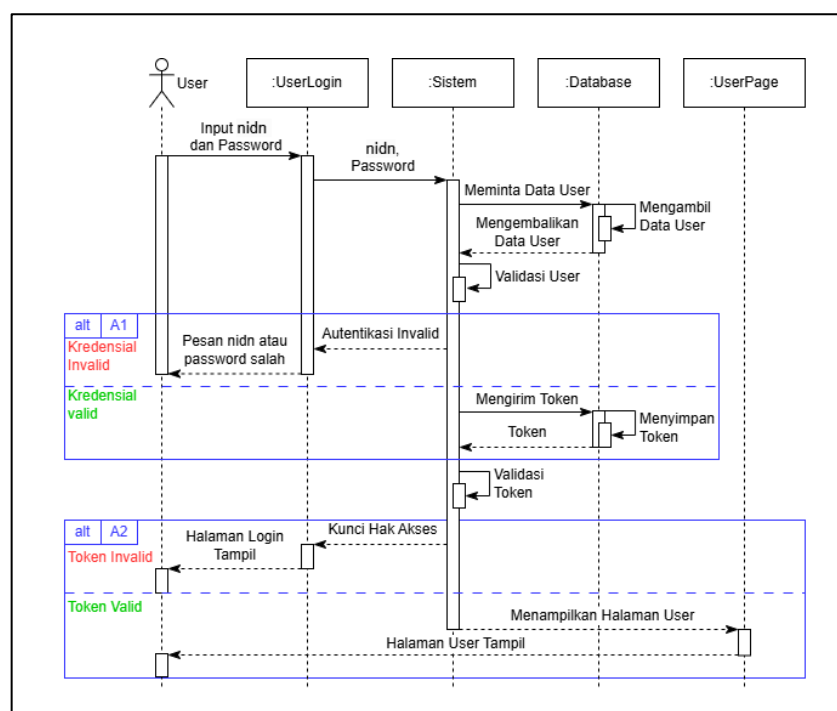
Gambar 13. *Activity Diagram Melakukan Logout*

c. *Sequence Diagram*

Sequence diagram adalah jenis diagram UML yang menunjukkan interaksi antar objek dalam sistem berdasarkan urutan waktu. Ini juga menunjukkan bagaimana pesan dikirim untuk menjalankan suatu skenario atau proses. Diagram ini membantu melihat alur komunikasi antar komponen karena terdiri dari lifeline (garis kehidupan objek), message (pesan yang dikirimkan), dan activation bar (blok eksekusi metode). Mereka membantu dalam pengembangan perangkat lunak, membantu dalam mendokumentasikan proses bisnis, memudahkan pemahaman sistem, dan menganalisis dan menemukan kesalahan dalam interaksi sistem sebelum diterapkan.

1) *Sequence Diagram Melakukan Login (UC-01)*

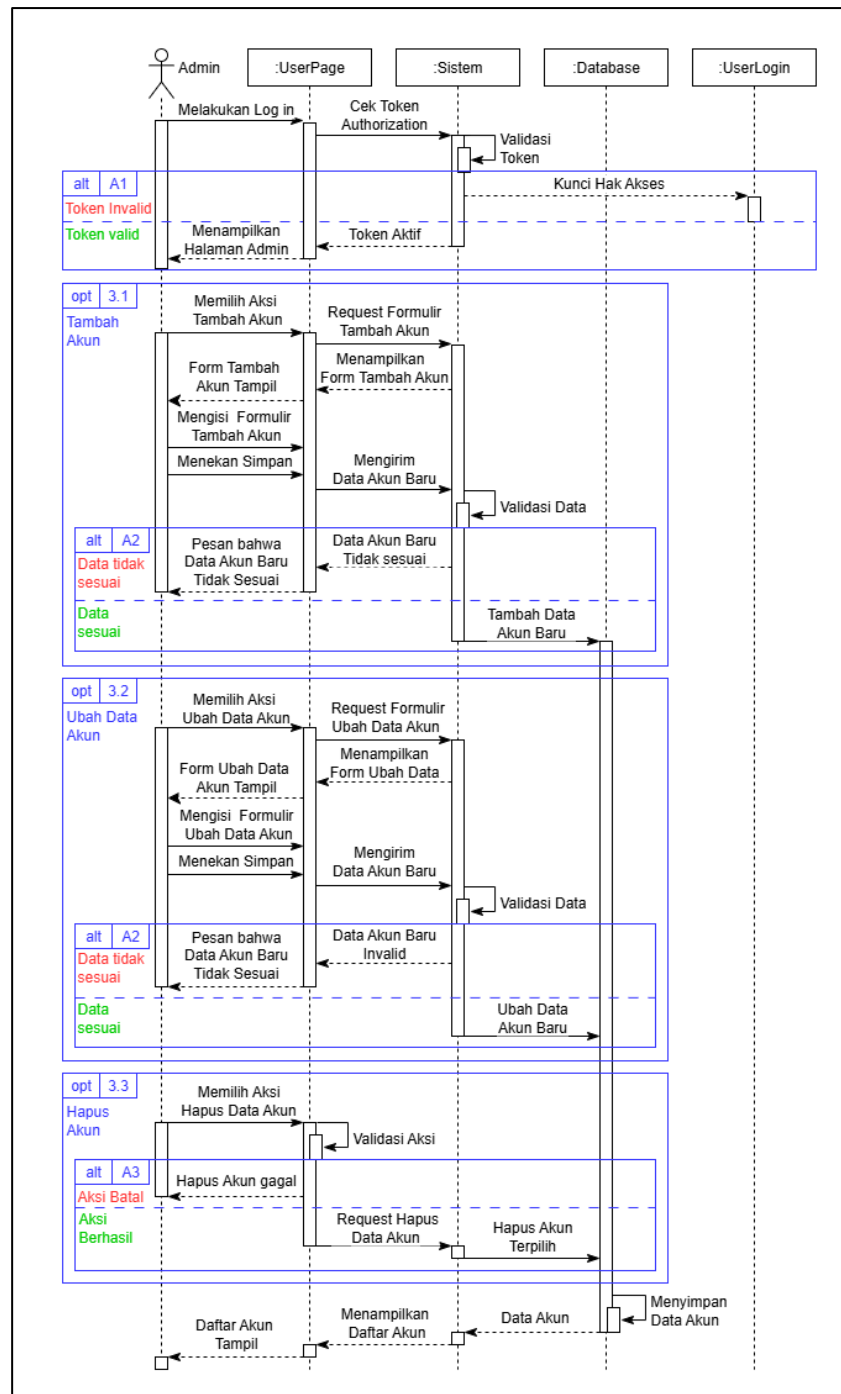
Sequence diagram menunjukkan alur kerja proses *login*. Pengguna pertama kali memasukkan username dan password mereka, dan kemudian sistem akan memverifikasi apakah mereka benar. Setelah itu, mereka akan dibawa ke halaman utama.



Gambar 14. *Sequence Diagram Melakukan Login*

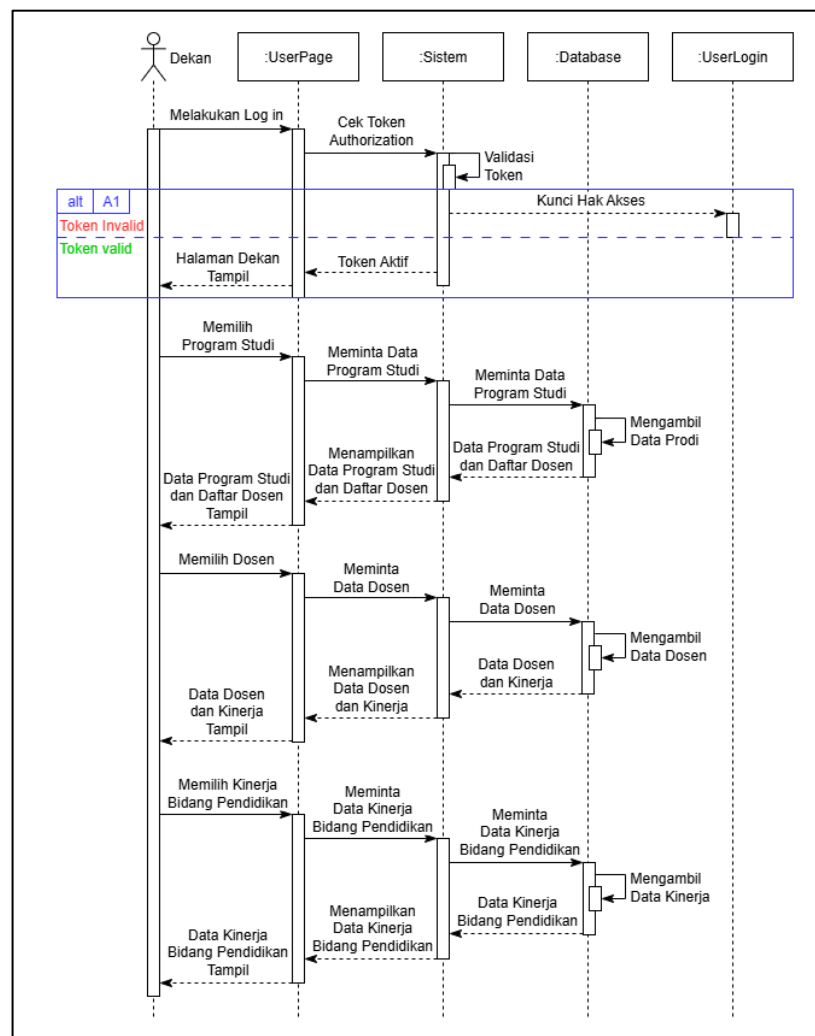
2) *Sequence Diagram Melakukan Mengelola Akun (UC-02)*

Alur pengelolaan akun yang ditunjukkan dalam *sequence diagram* ini memungkinkan pengguna untuk menambahkan, mengubah, atau menghapus akun. Sistem memverifikasi kebenaran data dan melakukan perubahan sesuai permintaan pengguna.

Gambar 15. *Sequence Diagram* Melakukan Mengelola Akun

3) *Sequence Diagram* Melihat Kinerja Bidang Pendidikan Tingkat Fakultas (UC-03)

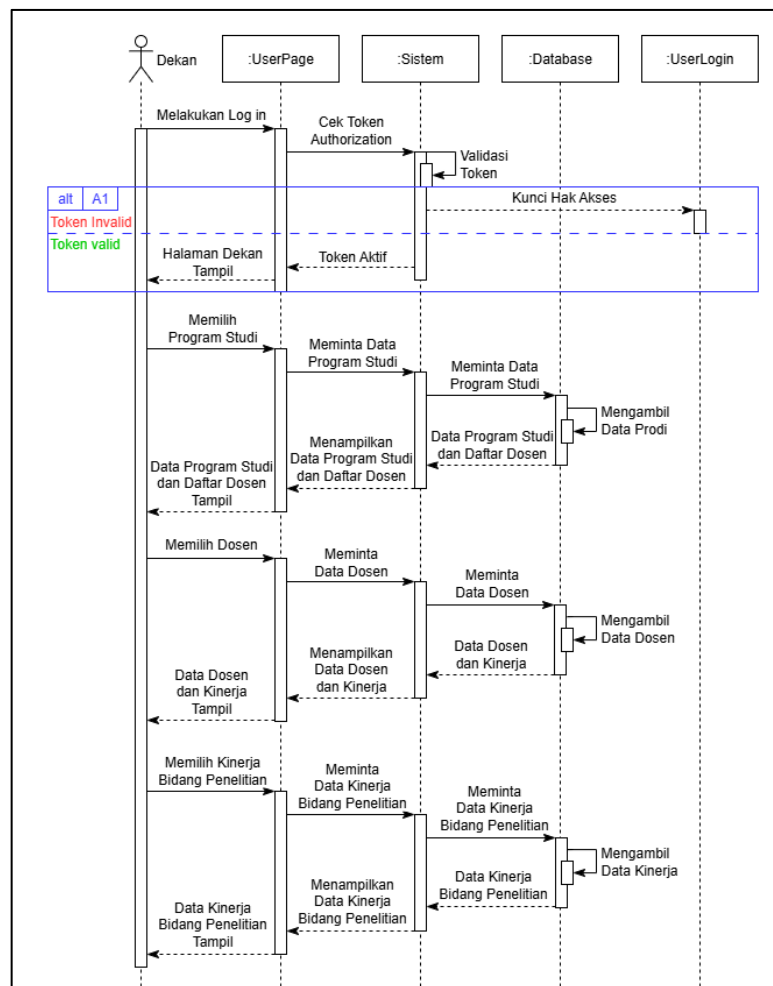
Proses melihat kinerja bidang pendidikan di tingkat fakultas digambarkan dalam *sequence diagram* ini. Dekan memiliki akses ke data kinerja ini. Setelah menerima permintaan, sistem akan mengumpulkan data tentang program studi, dosen, dan kinerja bidang pendidikan. Setelah itu, informasi akan ditampilkan sesuai dengan kebutuhan pengguna.



Gambar 16. *Sequence Diagram* Melihat Kinerja Bidang Pendidikan Tingkat Fakultas

4) *Sequence Diagram* Melihat Kinerja Bidang Penelitian Tingkat Fakultas (UC-04)

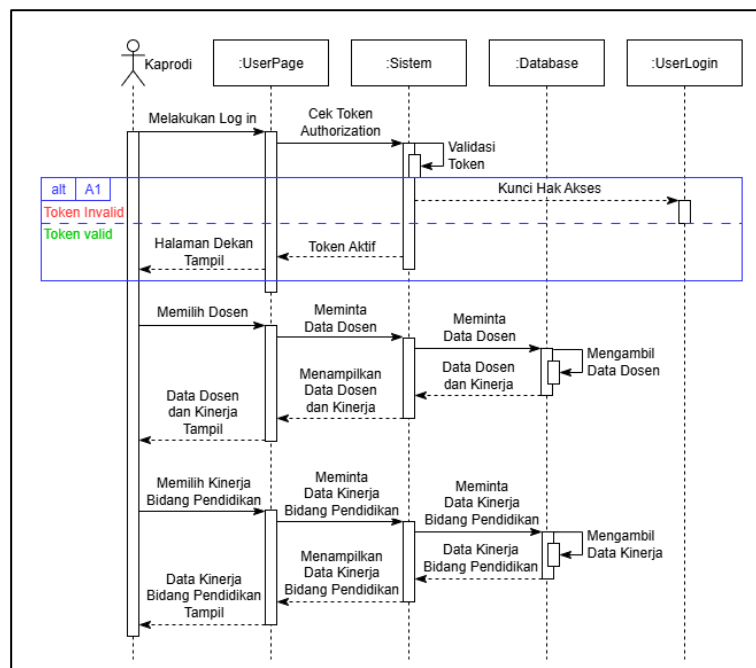
Proses melihat kinerja bidang penelitian di tingkat fakultas digambarkan dalam *sequence diagram* ini. Dekan memiliki akses ke data kinerja. Setelah permintaan diterima, sistem akan mengambil data program studi, dosen dan kinerja bidang penelitian. Kemudian, informasi akan ditampilkan sesuai dengan kebutuhan pengguna.



Gambar 17. *Sequence Diagram* Melihat Kinerja Bidang Penelitian Tingkat Fakultas

5) *Sequence Diagram* Melihat Kinerja Bidang Pendidikan Tingkat Prodi (UC-05)

Sequence Diagram ini menunjukkan proses melihat kinerja bidang pendidikan di tingkat program studi, yang memungkinkan ketua program studi untuk mendapatkan data. Setelah diminta, sistem akan mengumpulkan data tentang kinerja bidang pendidikan dan dosen, dan kemudian menampilkan informasi sesuai kebutuhan pengguna.

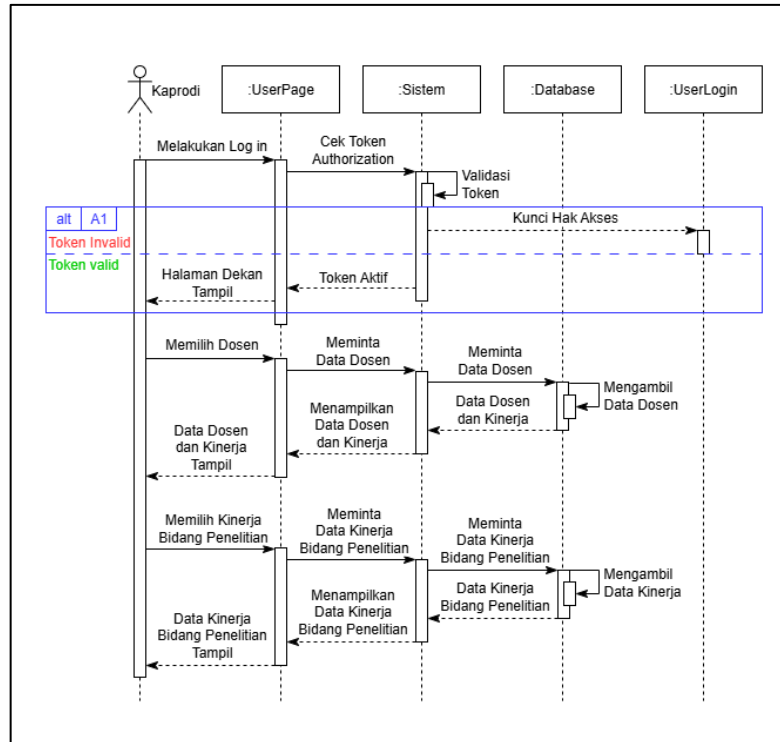


Gambar 18. *Sequence Diagram* Melihat Kinerja Bidang Pendidikan Tingkat Prodi

6) *Sequence Diagram* Melihat Kinerja Bidang Penelitian Tingkat Prodi (UC-06)

Sequence Diagram ini menunjukkan bagaimana data kinerja bidang penelitian di tingkat program studi dapat diakses oleh ketua program studi. Setelah menerima permintaan, sistem

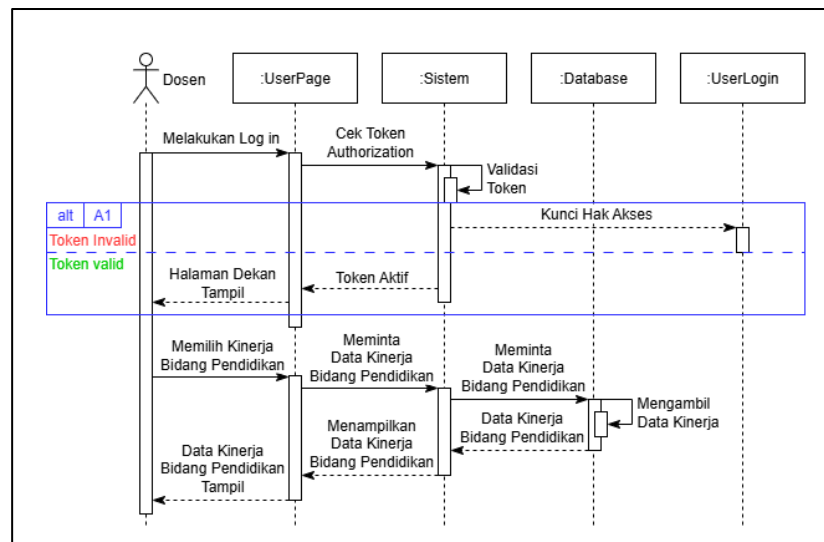
akan mengumpulkan data dosen dan kinerja bidang penelitian, dan kemudian menampilkan informasi sesuai kebutuhan pengguna.



Gambar 19. *Sequence Diagram* Melihat Kinerja Bidang Penelitian Tingkat Prodi

7) *Sequence Diagram* Melihat Kinerja Bidang Pendidikan (UC-07)

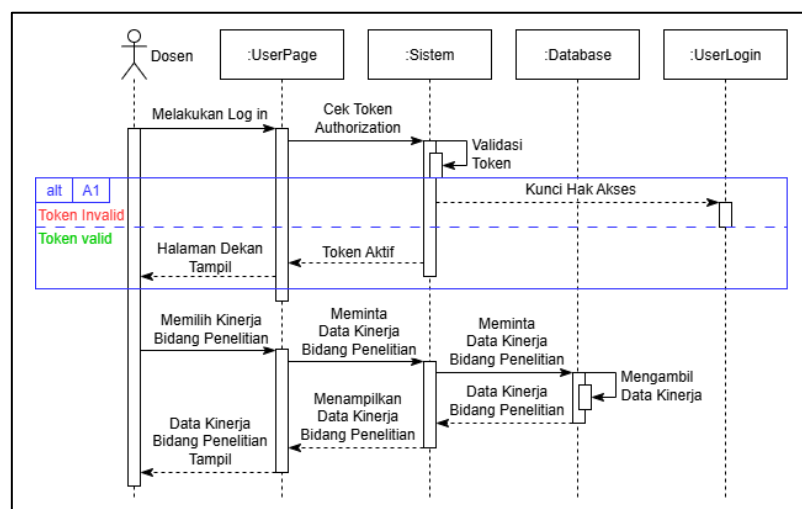
Sequence Diagram ini menunjukkan proses melihat kinerja bidang pendidikan, di mana dosen dapat mengakses data kinerja. Setelah permintaan diterima, sistem akan mengambil data dan menampilkan informasi sesuai kebutuhan pengguna.



Gambar 20. *Sequence Diagram* Melihat Kinerja Bidang Pendidikan

8) *Sequence Diagram* Melihat Kinerja Bidang Penelitian (UC-08)

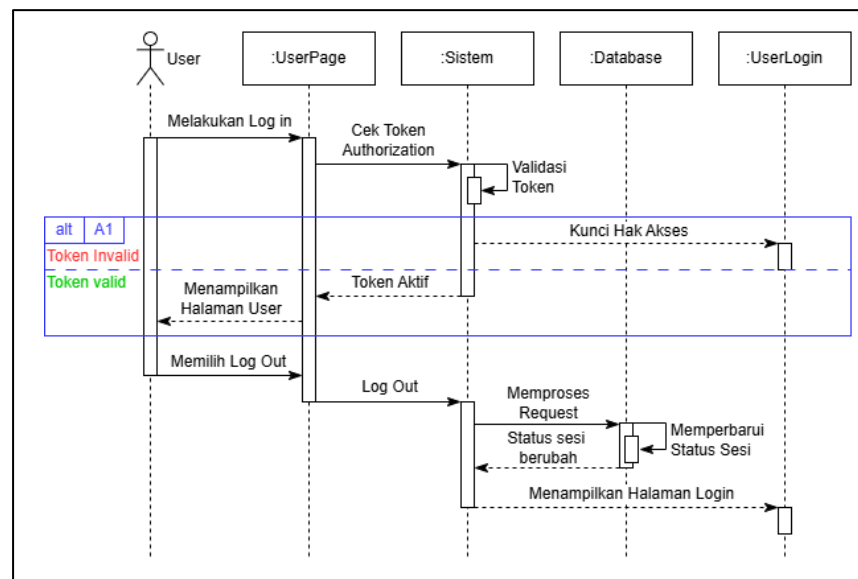
Sequence Diagram ini menunjukkan proses melihat kinerja bidang penelitian, di mana dosen dapat mengakses data kinerja. Setelah data kinerja bidang penelitian diterima, sistem akan menampilkan informasi sesuai kebutuhan pengguna.



Gambar 21. *Sequence Diagram* Melihat Kinerja Bidang Penelitian

9) Sequence Diagram Melakukan Logout

Alur proses *logout* digambarkan dalam *sequence diagram* ini. Pengguna memilih opsi *logout* untuk keluar dari system. Setelah permintaan diterima, sistem akan mengakhiri sesi pengguna, menghapus data sesi yang tersimpan, dan mengarahkan pengguna ke halaman *login*.



Gambar 22. Sequence Diagram Melakukan Logout

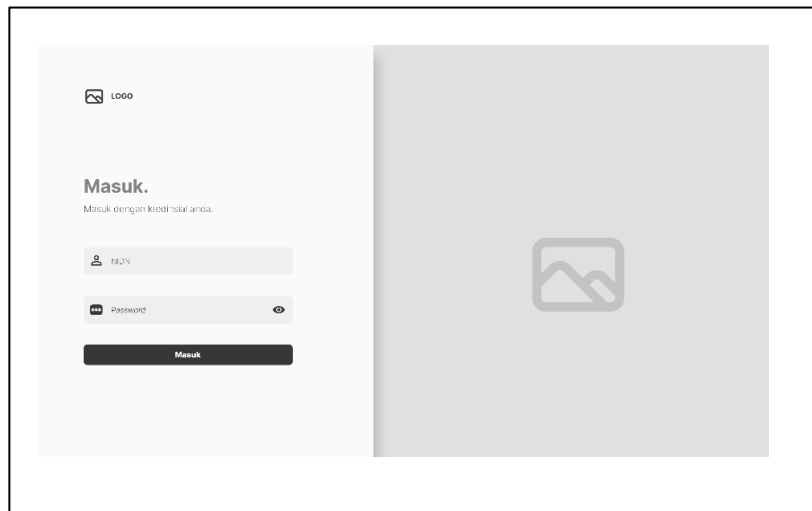
d. Wireframe

Berdasarkan *use case diagram* dan *activity diagram* yang telah dibuat sebelumnya, rancangan desain dibuat sebagai pedoman untuk membuat *website*. Dengan rancangan ini, pengembangan situs *web* dapat berjalan lebih efisien.

1) Rancangan Tampilan Login

Halaman ini digunakan untuk masuk pada *website* remunerasi dosen, setelah memiliki akun pengguna dapat

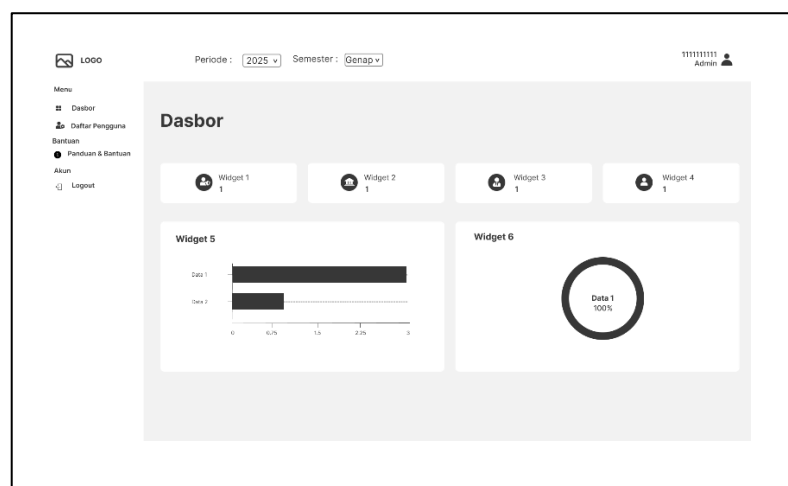
masuk kedalam *website* dengan menggunakan NIDN dan *password* yang disediakan oleh admin.



Gambar 23. Tampilan *Login*

2) Rancangan Tampilan Dasbor

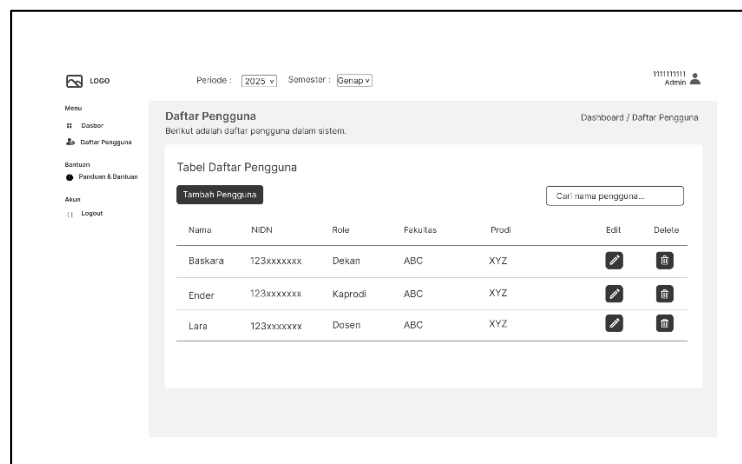
Halaman ini digunakan sebagai *home page* atau *default page* setelah proses *login*. Dasbor menampilkan Kumpulan *widget* dan informasi yang relevan dengan peran pengguna yang sedang *login*.



Gambar 24. Rancangan Tampilan Dasbor

3) Rancangan Tampilan Mengelola Akun

Tampilan ini dirancang untuk admin dapat mengelola akun pengguna, serta memungkinkan admin untuk mengatur hak akses seperti menambahkan akun baru, mengedit data pengguna, dan menghapus akun yang tidak diperlukan. Fitur ini dirancang untuk memastikan bahwa pengelolaan data pengguna berjalan dengan baik dan terorganisir.



Gambar 25. Tampilan Mengelola Akun

4) Rancangan Tampilan Tambah Akun

Halaman ini menampilkan form input yang digunakan untuk memudahkan admin untuk menambahkan akun baru pada sistem kinerja. Serta terdapat icon simpan yang digunakan untuk menyimpan data dan batal digunakan untuk membatalkan proses penambahan akun.

Gambar 26. Tampilan Tambah Akun

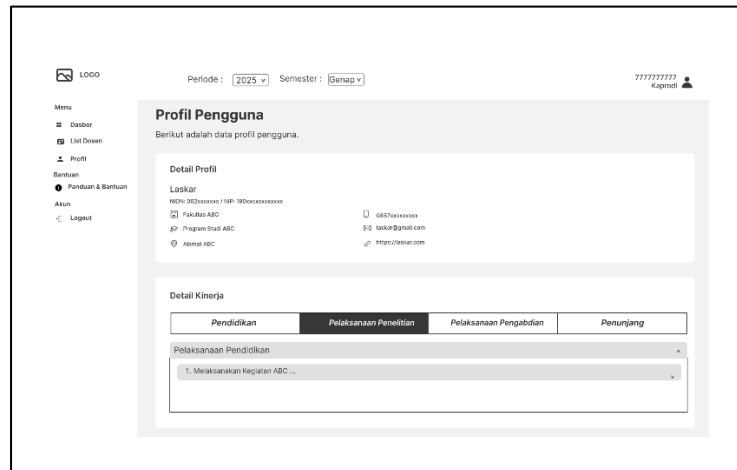
5) Rancangan Tampilan Edit Akun

Halaman ini menampilkan form yang berisi data akun yang dapat diedit sesuai dengan kebutuhan. Data awal ditampilkan secara otomatis dalam form sebagai nilai default, sehingga dapat memudahkan admin untuk memperbarui data tertentu tanpa mengisi ulang seluruh data.

Gambar 27. Tampilan Edit Akun

6) Rancangan Tampilan Profile

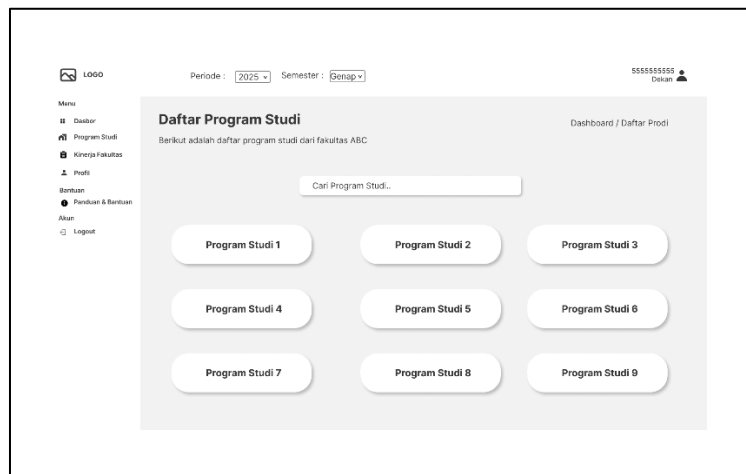
Halaman ini menampilkan informasi pengguna pada sistem. Informasi pengguna yang ditampilkan seperti detail profil yang berisi nama, NIDN/Nip, fakultas, jurusan, no hp, email dan lain-lain.



Gambar 28. Tampilan Profile

7) Rancangan Tampilan Daftar Program Studi

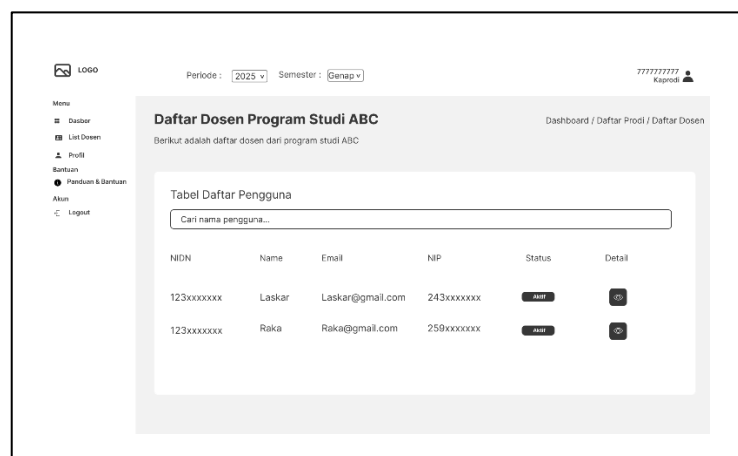
Halaman ini berisi tentang program studi yang berada di fakultas tersebut. Informasi yang disediakan pada halaman ini mencakup daftar program studi. Dengan mengklik salah satu program studi maka akan diarahkan ke halaman daftar dosen yang terdapat pada program studi tersebut.



Gambar 29. Tampilan Daftar Program Studi

8) Rancangan Tampilan Daftar Dosen

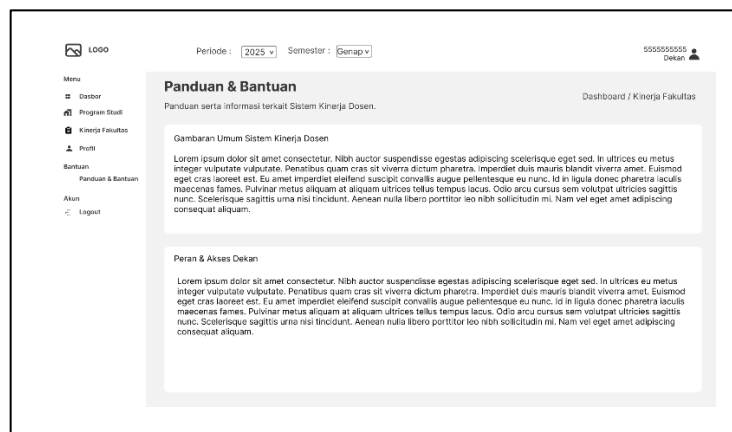
Halaman ini menampilkan daftar nama dosen yang terdapat dalam program studi. Dengan mengklik salah satu nama dosen maka akan diarahkan ke halaman berikutnya yang akan menampilkan informasi tentang kinerja dosen yang telah dilakukan. Halaman ini dirancang untuk mempermudah pengguna dalam menelusuri kinerja dosen.



Gambar 30. Tampilan Daftar Dosen

9) Rancangan Tampilan Panduan & Bantuan

Halaman ini menampilkan Gambaran umum sistem kinerja dosen serta peran dan akses dekan. Di dalam halaman ini juga terdapat fitur yang tersedia untuk mengetahui bagian yang bisa di akses oleh dekan.



Gambar 31. Tampilan Panduan & Bantuan Pada Sistem

3. *Sprint Planning*

Dalam *sprint planning*, *product owner* akan menjalankan *product backlog* kepada *development team* (seluruh tim). Dengan *sprint planning* tim melakukan kolaborasi atau bertukar pikiran untuk menentukan *product backlog* yang dipilih sesuai dengan *sprint goal*, setelah selesai pemilihan *product backlog* tersebut maka akan menghasilkan *sprint backlog*. Sprint tersebut harus diselesaikan oleh tiap development pada setiap *sprint*.

a. *Sprint ke-1*

Detail *Sprint ke-1* dilampirkan pada tabel berikut.

Tabel 19. *Sprint ke-1*

Task Product Backlog	Task	Kepentingan	Waktu	Penanggung Jawab
Modeling	- Membuat <i>use case</i> diagram - Membuat <i>activity diagram</i>	Tinggi	8 Hari	Tim
Pembuatan <i>Mockup</i>	- Membuat <i>wireframe</i> - Membuat <i>hi-fi</i>	Sedang	6 Hari	Tim

b. *Sprint ke-2*

Detail *Sprint ke-2* dilampirkan pada tabel berikut.

Tabel 20. *Sprint ke-2*

Task Product Backlog	Task	Kepentingan	Waktu	Penanggung Jawab
<i>Login dan logout</i>	- Membuat halaman <i>login dan logout</i> - Mengelola token autentikasi yang disediakan oleh <i>backend</i> - Membuat fungsi login dan <i>logout</i>	Tinggi	9 Hari	Amalia, Chandra
Dasbor	- Membuat halaman dasbor untuk semua peran	Sedang	5 Hari	Amalia, Chandra

<i>Task Product Backlog</i>	<i>Task</i>	Kepentingan	Waktu	Penanggung Jawab
	- Menarik data yang telah disediakan oleh <i>backend</i> melalui API			

c. *Sprint ke-3*

Detail *Sprint* ke-3 dilampirkan pada tabel berikut.

Tabel 21. *Sprint* ke-3

<i>Task Product Backlog</i>	<i>Task</i>	Kepentingan	Waktu	Penanggung Jawab
Mengelola akun	- Membuat halaman mengelola akun pengguna - Menarik data dari <i>backend</i> melalui API	Tinggi	7 Hari	Amalia, Chandra
Tambah akun	- Membuat Halaman untuk tambah akun pengguna - Sinkronisasi data ke <i>backend</i>	Sedang	4 Hari	Amalia, Chandra
Edit akun	- Membuat halaman edit akun pengguna - Sinkronisasi data ke <i>backend</i>	Sedang	3 Hari	Amalia, Chandra

d. *Sprint* ke-4

Detail *Sprint* ke-4 dilampirkan pada tabel berikut.

Tabel 22. *Sprint* ke-4

<i>Task Product Backlog</i>	<i>Task</i>	Kepentingan	Waktu	Penanggung Jawab
Hapus akun	- Menambahkan fungsi hapus akun - Sinkronisasi data ke <i>backend</i>	Sedang	3 Hari	Amalia, Chandra
Daftar program studi	- Membuat halaman daftar program studi - Menarik data dari <i>backend</i> melalui API	Sedang	4 Hari	Amalia, Chandra
Daftar dosen	- Membuat halaman yang menampilkan daftar dosen - Menarik data yang telah disediakan oleh <i>backend</i>	Sedang	7 Hari	Amalia, Chandra

e. *Sprint* ke-5

Detail *Sprint* ke-5 dilampirkan pada tabel berikut.

Tabel 23. *Sprint* ke-5

<i>Task Product Backlog</i>	<i>Task</i>	Kepentingan	Waktu	Penanggung Jawab
Profil	- Membuat halaman profil - Menarik data dari <i>backend</i> melalui API	Tinggi	7 Hari	Amalia, Chandra
Kinerja Fakultas	- Membuat halaman kinerja fakultas - Menarik data dari <i>backend</i> melalui API	Sedang	3 Hari	Amalia, Chandra
Kinerja Program Studi	- Membuat halaman kinerja program studi - Menarik data dari <i>backend</i> melalui API	Sedang	3 Hari	Amalia, Chandra
Panduan & Bantuan	- Membuat halaman panduan sistem	Rendah	1 Hari	Amalia, Chandra

4. *Sprint*

Setiap *sprint* memiliki tujuan yang jelas, dan tim berusaha menyelesaikan pekerjaan yang telah direncanakan agar fitur atau fungsi tertentu dapat diterapkan atau diuji pada akhir *sprint*. *Sprint* juga merupakan periode waktu terfokus di mana tim bekerja untuk menyelesaikan item dalam *backlog sprint*.

5. *Daily Scrum*

Pada tahap ini, pengembang mengadakan pertemuan harian, atau *daily scrum*, pada waktu yang telah ditetapkan untuk membahas progres tugas-tugas yang telah diselesaikan dari *backlog sprint*. Pertemuan harian ini memungkinkan setiap anggota tim untuk berbagi informasi, menemukan masalah, dan memastikan bahwa pengembang memiliki pemahaman yang sama tentang kemajuan dan kebutuhan *sprint*.

6. *Sprint Review*

Pada tahap ini, *product owner*, *Scrum Master*, dan *tim developer* berkumpul untuk membahas bagaimana meningkatkan hasil kerja tim untuk menyelesaikan *sprint*. Mereka juga melihat apa yang telah dicapai dari semua item yang ada di *backlog sprint* dan menilai pencapaian, atau peningkatan. Selain itu, jika diperlukan, perubahan pada *backlog* produk didiskusikan untuk memastikan bahwa kebutuhan dan prioritas tetap sesuai dengan tujuan produk.

7. *Sprint Retrospective*

Pada tahap ini, *Scrum Master* akan meninjau hasil kerja dari *sprint* pertama hingga *sprint* kedua dan merencanakan perbaikan untuk *sprint* berikutnya. *Scrum Master* mengevaluasi proses implementasi dan memperbarui *backlog sprint*, mengintegrasikan kebutuhan atau perubahan ke dalam perencanaan *sprint* yang baru. Diberikan kesempatan kepada setiap anggota tim untuk berbicara dan memberikan pendapat mereka, sehingga seluruh tim dapat menghasilkan gagasan dan rekomendasi untuk meningkatkan kinerja pada *sprint* berikutnya.

3.3.5 Pengujian

Pengujian yang dilakukan dalam penelitian ini menggunakan pengujian *Black-box*. Berikut ini adalah skenario untuk menguji sistem kinerja.

1. Pengujian fungsionalitas

- 1) Hasil pengujian fungsionalitas dengan menggunakan pengujian *black box* yang mencakup *positive case* sebagai berikut.

Tabel 24. Skenario Pengujian *Positive Case Black Box*

Kelas Uji	Kode Tes	Skenario Pengujian	Langkah Pengujian	Hasil Yang Diharapkan
Login dan logout	TCP-001	Mengakses halaman <i>login</i> sebelum melakukan <i>login</i>	Akses URL <i>login</i> melalui <i>browser</i>	Menampilkan halaman <i>login</i>
	TCP-002	Mengakses halaman <i>login</i> sesudah melakukan <i>login</i>	Akses URL <i>login</i> melalui <i>browser</i>	Mengarahkan pengguna ke halaman dasbor
	TCP-003	Melakukan <i>login</i> dengan data valid	Mengisi <i>field input</i> dengan data <i>valid</i>	Pengguna diarahkan ke halaman dasbor
	TCP-004	Melakukan <i>logout</i>	Tidak mengisi <i>field input</i>	Pengguna diarahkan ke halaman <i>login</i>
Mengelola akun	TCP-005	Mengakses halaman daftar pengguna	Memilih menu daftar pengguna	Menampilkan halaman daftar pengguna
Tambah akun	TCP-006	Mengakses halaman tambah pengguna	Mengklik tombol tambah pengguna	Menampilkan halaman tambah pengguna

Kelas Uji	Kode Tes	Skenario Pengujian	Langkah Pengujian	Hasil Yang Diharapkan
Edit akun	TCP-007	Menambah akun pengguna baru	Mengisi <i>field input</i> dengan benar	Muncul pesan berhasil membuat akun pengguna baru, mengarahkan pengguna ke halaman daftar pengguna
	TCP-008	Mengakses halaman edit pengguna	Mengklik tombol edit pengguna	Menampilkan halaman edit pengguna
	TCP-009	Mengedit akun pengguna	Mengisi <i>field input</i> dengan benar	Muncul pesan berhasil mengedit akun pengguna, mengarahkan pengguna ke halaman daftar pengguna
Hapus akun	TCP-010	Menghapus akun pengguna	Mengklik tombol hapus akun	Muncul pesan berhasil menghapus akun pengguna
Profil	TCP-011	Mengakses halaman profil	Memilih menu profil	Menampilkan halaman profil pengguna
	TCP-012	Mengakses data kinerja berdasarkan periode tertentu	Pengguna memilih menu periode yang ada pada <i>header</i>	Menampilkan halaman profil pengguna berisi data pribadi
	TCP-013	Mengakses data kinerja pribadi	Pengguna memilih menu profil dari <i>sidebar</i> dan mengklik tab bidang kinerja yang ingin dilihat.	Menampilkan data kinerja pribadi
Kinerja program studi	TCP-014	Mengakses data kinerja Tingkat program studi	Pengguna memilih menu kinerja program	Pengguna memilih menu kinerja program studi dari sidebar

Kelas Uji	Kode Tes	Skenario Pengujian	Langkah Pengujian	Hasil Yang Diharapkan
			studi dari sidebar	
	TCP-015	Mengakses data kinerja berdasarkan periode tertentu	Pengguna memilih periode yang ada pada <i>header</i>	Menampilkan data berdasarkan periode yang dipilih
	TCP-016	Meng export data kinerja tingkat program studi	Pengguna berada dalam halaman kinerja tingkat program studi dan mengklik tombol export pada bidang kinerja yang ingin diunduh	Mengunduh data kinerja tingkat program studi berdasarkan bidang terpilih
Daftar program studi	TCP-017	Mengakses halaman daftar program studi	Memilih menu daftar program studi	Menampilkan halaman daftar program studi
	TCP-018	Mengakses halaman daftar dosen	Mengklik tombol program studi	Menampilkan halaman daftar dosen dari program studi yang dipilih
Daftar dosen	TCP-019	Mengakses halaman daftar dosen	Memilih menu daftar dosen	Menampilkan halaman daftar dosen
	TCP-020	Mengakses halaman profil dosen	Mengklik tombol profil dosen	Menampilkan halaman profil dari dosen yang dipilih

- 2) Hasil pengujian fungsionalitas dengan menggunakan pengujian *black box* yang mencakup *negative case* sebagai berikut.

Tabel 25. Skenario Pengujian *Negative Case Black Box*

Kelas Uji	Kode Tes	Skenario Pengujian	Langkah Pengujian	Hasil Yang Diharapkan
Login dan logout	TCN-001	Melakukan <i>login</i> dengan data tidak valid	Mengisi field input dengan NIDN atau <i>password</i> salah	Muncul Pesan <i>Error</i> bahwa NIDN atau <i>password</i> salah
	TCN-002	Mengakses halaman login sesudah masuk sistem	Pengguna mengakses url halaman sistem kinerja dalam keadaan telah melakukan proses login	Menampilkan halaman dasbor pengguna
	TCN-003	Melakukan <i>login</i> dengan data kosong	Tidak mengisi <i>field input</i>	Muncul Pesan <i>Error</i> bahwa <i>field input</i> tidak boleh kosong
Tambah akun	TCN-004	Menambah akun pengguna baru dengan input yang tidak valid	Pengguna mengklik tombol tambah pengguna di halaman daftar pengguna lalu mengisi kolom tambah data pengguna yang tidak sesuai format serta mengklik tombol submit	Muncul pesan error sesuai dengan kesalahan input
	TCN-005	Menambah akun pengguna baru dengan input yang kosong	Pengguna mengklik tombol tambah pengguna di halaman daftar pengguna serta mengklik tombol submit	Muncul pesan error bahwa <i>field input</i> tidak boleh kosong

Kelas Uji	Kode Tes	Skenario Pengujian	Langkah Pengujian	Hasil Yang Diharapkan
Edit akun	TCN-006	Mengedit akun pengguna yang tidak valid	Pengguna mengklik tombol edit di halaman daftar pengguna. Mengisi kolom ubah data pengguna yang tidak sesuai format. Mengklik tombol submit	Muncul pesan error sesuai dengan kesalahan input
Kontrol akses & otoritas	TCN-007	Mengakses halaman di luar hak akses	Akses URL halaman di luar hak akses melalui <i>browser</i>	Pengguna diarahkan ke halaman <i>error</i>
	TCN-008	Mengakses halaman yang tidak ada di sistem	Akses URL halaman yang tidak ada di sistem melalui <i>browser</i>	Pengguna diarahkan ke halaman <i>error</i>
	TCN-009	Mengakses halaman internal sebelum login	Jika pengguna mengakses halaman internal sebelum melakukan proses login, maka pengguna akan diarahkan ke halaman login. Hal ini terjadi karena halaman internal hanya bisa diakses jika pengguna memiliki token JWT dari proses login	Mengarahkan pengguna ke halaman login

2. Pengujian *responsivitas*

Berikut ini pengujian *responsivitas* dengan menggunakan *pengujian black box*.

Tabel 26. Skenario Pengujian *Responsivitas Black Box*

Kelas Uji	Kode Tes	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan
Responsivitas UI	TC-001	Mengakses sistem di ukuran berbagai layar	Mengakses sistem di berbagai ukuran layer (<i>desktop</i> (1440px), <i>tablet</i> (768px), <i>mobile</i> (425px)) melalui simulasi dev tools	Tampilan menyesuaikan ukuran layer yanpa distorsi
	TC-002	Mengakses sistem di berbagai jenis <i>browser</i>	Mengakses sistem di berbagai jenis <i>browser</i> (Firefox, Chrome, Edge)	Tampilan dan fungsi tetap berjalan dengan baik

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Hasil penelitian dan implementasi menunjukkan bahwa antarmuka *front-end* sistem kinerja di bidang pendidikan dan penelitian telah dirancang dan dikembangkan dengan baik menggunakan React JS untuk memenuhi kebutuhan pengguna. Sistem ini dapat mengelola dan menampilkan data kinerja dosen dengan baik, mendukung integrasi dan sinkronisasi data dengan akurat melalui API yang disediakan oleh tim *backend*, serta memfasilitasi program studi dalam mengakses informasi kinerja untuk keperluan akreditasi. Sistem ini dinilai berhasil memenuhi tujuan dalam penelitian ini karena pengujian fungsional menggunakan metode *black box* menunjukkan bahwa semua fitur utama berjalan dengan baik, sementara pengujian performa menggunakan *Lighthouse* menunjukkan hasil yang sangat baik pada perangkat desktop dan cukup optimal pada perangkat mobile.

5.2 Saran

Berdasarkan hasil dari penelitian yang telah dilakukan terdapat saran yang dapat dijadikan acuan pengembang selanjutnya sebagai berikut.

- 1) Berdasarkan hasil pengujian menggunakan *Lighthouse*, performa sistem pada perangkat mobile masih perlu ditingkatkan, terutama pada metrik *Total Blocking Time* (TBT). Oleh karena itu, disarankan untuk melakukan optimasi lanjutan terhadap proses rendering dan pengelolaan resource agar lebih ringan dan efisien pada perangkat dengan spesifikasi rendah.

- 2) Disarankan agar sistem menyediakan fitur reset atau atur ulang password secara mandiri oleh pengguna. Hal ini bertujuan untuk meningkatkan kenyamanan dan keamanan akses, serta mengurangi ketergantungan pengguna pada admin dalam proses pemulihan akun.

DAFTAR PUSTAKA

- Adam, A. (2021). *Front End Adalah: Pengertian, Cara Kerja, dan Perbedaannya dengan Back End*. Accurate. <https://accurate.id/teknologi/front-end-adalah/>
- Aliyah Aliyah, Nahrin Hartono, & Asrul Azhari Muin. (2024). Penggunaan User Acceptance Testing (UAT) Pada Pengujian Sistem Informasi Pengelolaan Keuangan Dan Inventaris Barang. *Switch : Jurnal Sains Dan Teknologi Informasi*, 3(1), 84–100. <https://doi.org/10.62951/switch.v3i1.330>
- Ardiansyah, A., Yahya, F. Y., Irawati, A. R., & Yusman, M. (2021). Pengembangan Sistem Informasi Terpadu Fmipa Universitas Lampung (Simipa) Menggunakan Metode Scrum. *Jurnal Teknoinfo*, 15(2), 112. <https://doi.org/10.33365/jti.v15i2.1041>
- Bota, Y. T., & Setiyawati, N. (2022). Pengembangan Sistem Informasi Perantara Bisnis Menggunakan Framework Flask. *Journal of Information Technology Ampera*, 3(2), 79–93. <https://doi.org/10.51519/journalita.volume3.issue2.year2022.page79-93>
- Dwiyanto, B., & Putro, H. P. (2022). Pengembangan Front-End Sistem Manajemen Kuis dengan Service Oriented Architecture. *Jurnal AUTOMATA*, 3(2).
- Fariz, M., Lazuardy, S., & Anggraini, D. (2022). Modern Front End Web Architectures with React.Js and Next.Js. *International Research Journal of Advanced Engineering and Science*, 7(1), 132–141.
- Faruk Ulum, & Rinaldi Muchtar. (2018). Pengaruh E-Service Quality Terhadap E-Customer Satisfaction Website Start-Up Kaosyay. *Jurnal Tekno Kompak*, 12(2), 68–72.
- Fitriastuti, F., Putri, A. E., Sunardi, A. K., & Hidayat, R. A. (2024). Analisis Website Siakad Universitas Janabadra Menggunakan Metode UAT. *Jurnal Teknologi Sistem Informasi*, 5(1), 276–285. <https://doi.org/10.35957/jtsi.v5i1.6998>
- Hasanah, F. N. (2020). Buku Ajar Rekayasa Perangkat Lunak. In *Buku Ajar Rekayasa Perangkat Lunak*. <https://doi.org/10.21070/2020/978-623-6833-89-6>
- Informasi Umum Beban Kerja Dosen (BKD)*. (n.d.). <https://pusatinformasi.sister.kemdikbud.go.id/hc/id/articles/23215418102041-Informasi-Umum-Beban-Kerja-Dosen-BKD>

- Iswari, L., & Nasution. (2021). Penerapan React JS Pada Pengembangan FrontEnd. *Automata*, 2(2), 193–200.
- Kharisma Raharjana, I. (2015). Pembuatan Model Sequence Diagram Dengan Reverse Engineering Aplikasi Basis Data Pada Smartphone. *Jurnal Ilmiah Teknologi Informasi*, 133–142.
- Khuat, & Tung. (2018). Developing a frontend application using ReactJS and Redux. *Laurea University of Applied Sciences*, 61. https://www.theseus.fi/bitstream/handle/10024/150837/Tung_Khuat_1301747_Thesis.pdf
- Kinerja, P. P. (2022). *Pedoman Rubrik Remunerasi Universitas Lampung Tahun 2022 Bandar Lampung*. 2022.
- Levlin, M. (2020). *DOM Benchmark Comparison Of The Front-End JavaScript Frameworks React, Angular, Vue, and Svelte*. 95.
- Noviantoro, A., Silviana, A. B., Fitriani, R. R., & Permatasari, H. P. (2022). Rancangan Dan Implementasi Aplikasi Sewa Lapangan Badminton Wilayah Depok Berbasis Web. *Jurnal Teknik Dan Science*, 1(2), 88–103. <https://doi.org/10.56127/jts.v1i2.108>
- Nuraminudin, M., Dewi, M. M., Suharsono, S., Dahlan, A., & Lukman, L. (2023). Implementasi Reactjs Pada Pembuatan Sistem Informasi Digital Printing Berbasis Website. *Information System Journal*, 6(01), 25–32. <https://doi.org/10.24076/infosjournal.2023v6i01.1214>
- Nurlailah, E., & Nova Wardani, K. R. (2023). Perancangan Website Sebagai Media Informasi Dan Promosi Oleh-Oleh Khas Kota Pagaralam. *JIPi (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, 8(4), 1175–1185. <https://doi.org/10.29100/jipi.v8i4.4006>
- Nursaid, F. F., Hendra Brata, A., & Kharisma, A. P. (2020). Pengembangan Sistem Informasi Pengelolaan Persediaan Barang Dengan ReactJS Dan React Native Menggunakan Prototype (Studi Kasus : Toko Uda Fajri). *J-Ptiik.Ub.Ac.Id*, 4(1), 46–55. <http://j-ptiik.ub.ac.id>
- Rahmadhani, S., Wildana, D. W., Arumdanie, H. W., & Hakim, L. (2024). Penerapan React JS dan Axios untuk Pengembangan Front-end Aplikasi iCare. *Software Development Digital Business Intelligence and Computer Engineering*, 2(02), 40–46. <https://doi.org/10.57203/session.v2i02.2024.40-46>
- Regita, N. (2023). *Mengenal Tridharma Perguruan Tinggi*. Suteki Technology. <https://suteki.co.id/pengertian-tri-dharma-perguruan-tinggi-dan-cara-penerapannya/>
- Rizaldy, R., & Dirgahayu, R. T. (2020). Pengembangan Front-End Sistem Informasi Pendataan Pendar Foundation Yogyakarta. *Automata*, 1(2), 190–194. <https://journal.uui.ac.id/AUTOMATA/article/view/15591/10252>
- Samsudin, S. (2019). Optimalisasi Penerimaan Remunerasi Dosen Menggunakan

- Metode Rule Base Reasoning. *Klik - Kumpulan Jurnal Ilmu Komputer*, 6(3), 224. <https://doi.org/10.20527/klik.v6i3.185>
- Siska Narulita, Ahmad Nugroho, & M. Zakki Abdillah. (2024). Diagram Unified Modelling Language (UML) untuk Perancangan Sistem Informasi Manajemen Penelitian dan Pengabdian Masyarakat (SIMLITABMAS). *Bridge : Jurnal Publikasi Sistem Informasi Dan Telekomunikasi*, 2(3), 244–256. <https://doi.org/10.62951/bridge.v2i3.174>
- Sitorus, S. L. (2021). Qualitative Method (Case Study Research). *Journal Of Communication Education*, 15(1), 20–29. <https://doi.org/10.58217/joce-ip.v15i1.224>
- Srivastava, S., Shukla, H., Landge, N., Srivastava, A., & Jindal, D. (2024). A Comprehensive Review of Next.js Technology: Advancements, Features, and Applications. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.4831070>
- Subeni, B. (2023). *Mengenal 7 Tugas Frontend Developer*. Jatis Mobile. <https://jatismobile.com/id/blog/aplikasi-dan-platform-digital/tugas-frontend-developer/>
- Syaifudin, M., Fauziah, F., & Rahman, B. (2021). Point of Sale Framework-Based Code Igniter and Model View Controller Using Lighthouse Testing. *Journal of Computer Networks, Architecture and High Performance Computing*, 3(2), 202–212. <https://doi.org/10.47709/cnahpc.v3i2.1006>
- Teknologi, J., Jtsi, I., Novitasari, Y. S., Adrian, Q. J., Kurnia, W., Informasi, S., Teknik, F., & Indonesia, U. T. (2021). *Rancang Bangun Sistem Informasi Media Pembelajaran Berbasis Website (Studi Kasus : Bimbingan Belajar De Potlood)*. 2(3), 136–147.
- Wijaya, A., & Hendrastuty, N. (2022). Rancang Bangun Sistem Informasi Manajemen Kepegawaian (Simpeg) Berbasis Web (Studi Kasus : Pt Sembilan Hakim Nusantara). *Jurnal Teknologi Dan Sistem Informasi (JTSl)*, 3(2), 9–17. <http://jim.teknokrat.ac.id/index.php/JTSl>
- Wulandari, W., Nofiyani, N., & Hasugian, H. (2023). User Acceptance Testing (Uat) Pada Electronic Data Preprocessing Guna Mengetahui Kualitas Sistem. *Jurnal Mahasiswa Ilmu Komputer*, 4(1), 20–27. <https://doi.org/10.24127/ilmukomputer.v4i1.3383>
- Zareinia, K. (2024). *Benchmarking Front-End Frameworks: A Deep Dive Into The Performance And Scalability Of React, Angular, And Vue.js*. November.