

**RANCANG BANGUN *FRONTEND* SISTEM INFORMASI CAPAIAN
PEMBELAJARAN LULUSAN (CPL) BERBASIS WEB MENGGUNAKAN
METODE AGILE KANBAN**

(Skripsi)

Oleh

**IVAN SIMANGUNSONG
NPM 2015061060**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2026**

**RANCANG BANGUN *FRONTEND* SISTEM INFORMASI CAPAIAN
PEMBELAJARAN LULUSAN (CPL) BERBASIS WEB MENGGUNAKAN
METODE AGILE KANBAN**

**Oleh
IVAN SIMANGUNSONG**

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai
Gelar Sarjana Teknik**

**Pada
Program Studi Teknik Informatika
Fakultas Teknik Universitas lampung**



**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG**

2026

ABSTRAK

RANCANG BANGUN *FRONTEND* SISTEM INFORMASI CAPAIAN PEMBELAJARAN LULUSAN (CPL) BERBASIS WEB MENGGUNAKAN METODE AGILE KANBAN

Oleh

Ivan Simangunsong

Pendidikan tinggi memiliki peran penting dalam menghasilkan lulusan yang berkualitas dan kompetitif, sehingga setiap program studi perlu memastikan capaian pembelajaran lulusannya (CPL) sesuai standar yang ditetapkan. Saat ini, pengolahan dan penyajian data CPL di Universitas Lampung belum dilakukan secara sistematis, sehingga diperlukan sistem yang mampu mendukung pencatatan, perhitungan, dan penyajian informasi CPL secara terstruktur. Penelitian ini bertujuan merancang dan membangun *frontend* Sistem Informasi CPL berbasis *web* menggunakan *React.js* dengan metode *Agile Kanban*. Tahapan penelitian meliputi *studi literatur*, analisis kebutuhan *fungsional* dan *nonfungsional*, perancangan arsitektur *frontend*, serta pengembangan menggunakan papan *Kanban* yang terdiri atas kolom *Backlog/To Do*, *In Progress*, *Testing*, dan *Done*. Hasil penelitian menunjukkan bahwa *frontend* sistem berhasil dibangun dengan struktur *modular*, *layer* tampilan terpisah, *hooks*, *service*, dan *API* yang mendukung pengelolaan alur data secara efektif. Pengujian fungsional menggunakan metode *black box testing* menunjukkan tingkat keberhasilan sebesar 78,34% yang menandakan sebagian besar *test case* berjalan sesuai ekspektasi dan fungsi utama *frontend* dapat beroperasi dengan baik. Penelitian ini menghasilkan artefak teknis berupa *frontend* Sistem Informasi CPL yang dapat dijadikan referensi atau *prototype* untuk pengembangan sistem lebih lanjut.

Kata kunci: Capaian Pembelajaran Lulusan, *Frontend*, *React.js*, *Agile Kanban*, *Black Box Testing*.

ABSTRACT

Design and Development of a Web-Based Frontend for Graduate Learning Outcomes Information System (CPL) Using Agile Kanban Method

By

Ivan Simangunsong

Higher education plays an important role in producing qualified and competitive graduates, making it essential for each study program to ensure that its graduates' learning outcomes (Capaian Pembelajaran Lulusan / CPL) align with established standards. Currently, the processing and presentation of CPL data at the University of Lampung are not carried out systematically, highlighting the need for a system that can support structured recording, calculation, and presentation of CPL information. This study aims to design and develop a web-based frontend for the CPL Information System using React.js and the Agile Kanban method. The research stages include a literature review, analysis of functional and non-functional requirements, frontend architecture design, and development using a Kanban board consisting of Backlog/To Do, In Progress, Testing, and Done columns. The results show that the frontend system was successfully built with a modular structure, separate view layers, hooks, services, and APIs, which effectively support data flow management. Functional testing using the black-box testing method yielded a success rate of 78.34%, indicating that the majority of test cases ran as expected and the main frontend functions operated properly. This study produces a technical artifact in the form of a CPL information system frontend that can serve as a reference or prototype for further system development.

Keywords: Graduate Learning Outcomes, Frontend, React.js, Agile Kanban, Black Box Testing.

Judul Skripsi : RANCANG BANGUN *FRONTEND*
SISTEM INFORMASI CAPAIAN
PEMBELAJARAN LULUSAN (CPL)
BERBASIS WEB MENGGUNAKAN
METODE AGILE KANBAN

Nama Mahasiswa : Ivan Simangunsong

Nomor Pokok Mahasiswa : 2015061060

Program Studi : Teknik Informatika

Fakultas : Teknik

MENYETUJUI

1. Komisi Pembimbing



Wahyu Eko S, S.T., M.Sc.
NIP. 19741201200112001



Puput Budi Wintoro, S.Kom., M.T.I.
NIP. 19841031 201903 1 004

2. Mengetahui

Ketua Jurusan
Teknik Elektro



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

Ketua Program Studi
Teknik Informatika



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

MENGESAHKAN

1. Tim Penguji

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



Sekretaris : Puput Budi Wintoro, S.Kom., M.T.I.



Penguji : Yessi Mulyani, S.T., M.T



2. Dekan Fakultas Teknik



Dr. H. Ahmad Herison, S.T., M.T
ID 196910302000031001

Tanggal Lulus Ujian Skripsi : 30 Januari 2026

SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa dalam skripsi ini tidak terdapat karya yang pernah dilakukan oleh orang lain dan sepanjang sepengetahuan saya tidak terdapat atas diterbitkannya oleh orang lain, kecuali secara tertulis diacu dalam naskah ini sebagaimana yang disebutkan dalam Daftar Pustaka. Selain itu, saya menyatakan pula bahwa skripsi ini dibuat oleh saya sendiri.

Apabila pernyataan saya tidak benar, maka saya bersedia dikenai sanksi akademik sesuai dengan hukum berlaku.

Bandar Lampung, 23 Februari 2026




Ivan Simangunsong
NPM. 2015061060

RIWAYAT HIDUP



Ivan Simangunsong dilahirkan di Siantar, Sumatera Utara, pada tanggal 17 Februari 2002. Penulis merupakan anak kedua dari tiga bersaudara dari pasangan Bapak Mukson Simangunsong dan Ibu Ridawati Lumban Tobing. Penulis menempuh pendidikan sekolah dasar di SD RK Budi Mulia 3 dan lulus pada tahun 2014. Pendidikan selanjutnya dilanjutkan ke SMP Negeri 2 Siantar dan berhasil diselesaikan pada tahun 2017. Pada jenjang pendidikan menengah atas, penulis menempuh pendidikan di SMA Negeri 1 Siantar dan lulus pada tahun 2020. Setelah menyelesaikan pendidikan menengah, penulis melanjutkan pendidikan ke jenjang perguruan tinggi dan aktif mengembangkan kompetensi akademik maupun keterampilan penunjang di bidang teknologi informasi.

Selama masa studi, penulis mengikuti berbagai program pengembangan kompetensi melalui skema Merdeka Belajar Kampus Merdeka (MBKM). Pada tahun 2022, penulis mengikuti Program Studi Independen *Fresh Graduate Academy* dengan bidang *AI for Junior Developer*. Selanjutnya, pada tahun 2023 penulis mengikuti *Hactive Teknologi Program* dengan bidang *Web Developer Program*. Selain itu, penulis juga mengikuti berbagai pelatihan profesional, di antaranya *Alibaba Cloud Certification* pada bidang *Cloud Services* pada tahun 2024, serta Program *Fresh Graduate Academy* pada bidang *IT Fundamental* pada tahun yang sama. Sebagai bagian dari pemenuhan pengalaman kerja akademik, penulis melaksanakan kerja praktik di PT Queen Network Nusantara pada tahun 2023. Dalam kegiatan tersebut, penulis menyusun laporan kerja praktik dengan judul “Pengembangan *Front-End Dashboard* APJII di PT Queen Network Nusantara Berbasis *Web* Menggunakan *ReactJS*”.

MOTTO

*“Do not be anxious about anything, but in every situation, by prayer and petition,
with thanksgiving, present your requests to God.”*

- *Philippians 4:6* -

“Just Keep Swimming”

- *Finding Nemo, 2003* -

PERSEMBAHAN

Dengan penuh rasa syukur

Karya ini saya persembahkan kepada:

Orang Tua Tercinta

**Bapak Mukson Simangunsong
&
Ibu Ridawati Lumban Tobing**

Serta Abang dan Adik Saya

**Arie Setiawan Simangunsong
&
Cintya Margareth Simangunsong**

Terima kasih untuk semua doa dan dukungan selama ini.

Sehingga aku dapat menyelesaikan hasil karyaku ini.

UCAPAN TERIMA KASIH

Segala puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas kasih, penyertaan, dan berkat-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Rancang Bangun *Frontend* Sistem Informasi Capaian Pembelajaran Lulusan (CPL) Berbasis Web Menggunakan Metode *Agile Kanban*” dengan baik. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung.

Penyusunan skripsi ini tidak terlepas dari bantuan, dukungan, serta bimbingan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Kedua orang tua penulis, Ayah Mukson Simangunsong dan Ibu Ridawati Lumban Tobing, serta saudara kandung penulis Arie Setiawan Simangunsong dan Cintya Margareth Simangunsong, yang senantiasa memberikan doa, kasih sayang, dukungan moral, serta kepercayaan kepada penulis dalam setiap proses kehidupan dan penyelesaian studi ini.
2. Keluarga besar Oppung Brielle Simangunsong, yang telah memberikan perhatian, bimbingan, serta dukungan moral kepada penulis selama menempuh pendidikan dan proses penyusunan skripsi ini.
3. Bapak Dr. Ahmad Herison, S.T., M.T. selaku Dekan Fakultas Teknik Universitas Lampung.
4. Ibu Herlinawati, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung.
5. Ibu Yessi Mulyani, S.T., M.T. selaku Kepala Program Studi Teknik Informatika Fakultas Teknik Universitas Lampung.

6. Bapak Wahyu Eko S., S.T., M.Sc. selaku Dosen Pembimbing Utama, yang telah meluangkan waktu, memberikan arahan, bimbingan, serta masukan yang sangat berarti selama proses penyusunan skripsi ini.
7. Bapak Puput Budi Wintoro, S.Kom., M.T.I. selaku Dosen Pembimbing Pendamping, yang senantiasa memberikan saran, kritik, dan dukungan dalam penyempurnaan penelitian ini.
8. Ibu Yessi Mulyani, S.T., M.T. selaku Dosen Penguji, yang telah memberikan masukan dan saran yang membangun demi kesempurnaan skripsi ini.
9. Segenap dosen dan staf akademik Jurusan Teknik Elektro serta Fakultas Teknik Universitas Lampung, yang telah memberikan ilmu, bimbingan, pelayanan akademik, serta pengalaman berharga selama penulis menempuh pendidikan.
10. Teman-teman perjuangan (Syarif dan Faniel) yang telah mendukung dan membantu penulis selama masa perkuliahan.

Penulis menyadari bahwa skripsi ini masih memiliki keterbatasan. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun. Semoga skripsi ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan, khususnya di bidang teknologi informasi.

Bandar Lampung, 23 Februari 2026

Penulis,

Ivan Simangunsong

DAFTAR ISI

	Halaman
ABSTRAK	iii
ABSTRACT	iv
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvi
DAFTAR TABEL	20
I. PENDAHULUAN.....	22
1.1 Latar Belakang	22
1.2 Rumusan Masalah	23
1.3 Tujuan Penelitian.....	24
1.4 Manfaat Penelitian.....	24
1.5 Batasan Masalah.....	24
1.6 Tim Penelitian	25
1.7 Sistematika Penulisan.....	26
II. TINJAUAN PUSTAKA	27
2.1 Capaian Pembelajaran Lulusan (CPL)	27
2.1.1 Alur Perhitungan CPL.....	27
2.2 <i>Software Development Life Cycle (SDLC)</i>	28
2.3 Agile	29
2.4 UML (<i>Unified Modeling Language</i>)	32
2.4.1 <i>Use Case Diagram</i>	33
2.4.2 <i>Activity Diagram</i>	34
2.4.3 <i>Sequence Diagram</i>	36
2.5 <i>Blackbox Testing</i>	37
2.6 <i>React Js</i>	38
2.7 <i>Tailwind</i>	39
2.8 Penelitian Terdahulu.....	40
III. METODOLOGI PENELITIAN	44

3.1	Waktu dan Tempat	44
3.1.1	Waktu	44
3.1.2	Tempat.....	44
3.2	Alat dan Bahan Penelitian	45
3.2.1	Alat Penelitian	45
3.3	Tahapan Penelitian	46
3.3.1	Studi Literatur	46
3.3.2	Analisis Kebutuhan Sistem	47
3.3.3	Perancangan Sistem	47
3.3.4	<i>Kanban Board</i>	47
3.3.5	<i>Development</i>	48
IV.	HASIL DAN PEMBAHASAN	49
4.1	Studi Literatur.....	49
4.2	Analisis Kebutuhan Sistem	50
4.2.1	Kebutuhan <i>Fungsional</i>	50
4.2.2	Kebutuhan <i>Non Fungsional</i>	51
4.3	Perancangan Sistem.....	52
4.3.1	<i>Use Case Diagram</i>	52
4.3.2	<i>Activity Diagram</i>	54
4.3.3	Sequence Diagram Pengelolaan Data	75
4.3.4	Arsitektur sistem website	76
4.4	Perancangan <i>Kanban Board</i>	77
4.5	<i>Development</i>	78
4.5.1	Authentikasi	78
4.5.2	Admin Universitas	82
4.5.3	Admin Program Studi	89
4.5.4	User Dosen	115
4.5.5	<i>User Kepala Program Studi</i>	119
4.5.6	Testing.....	124
V.	PENUTUP	160
5.1	Kesimpulan.....	160
5.2	Saran	162
	DAFTAR PUSTAKA	163

LAMPIRAN.....	165
Lampiran A. Dokumentasi API yang Telah Terimplementasi.....	165
Lampiran B. Dokumentasi API yang Belum Terimplementasi	172
Lampiran C. Saran Rencana Pengembangan API Sistem.....	176

DAFTAR GAMBAR

	Halaman
Gambar 1. Tim Pengembang.....	25
Gambar 2. Rumus Perhitungn Nilai CPL.....	28
Gambar 3. <i>Software Development Life Cycle</i>	29
Gambar 4. Contoh Papan Kanban[7]	30
Gambar 5. <i>Class diagram of a Package owning a set of UseCases, Actors, and a Subsystem</i> [9]	33
Gambar 6. Rumus pengujian <i>Test Case Pass</i> [10].....	38
Gambar 7. Metodologi Penelitian	46
Gambar 8. Use Case Diagram	52
Gambar 9. <i>Activity Diagram Login User</i>	54
Gambar 10. <i>Activity Diagram Logout User</i>	55
Gambar 11. <i>Activity Diagram</i> Kelola Akun oleh Admin Universitas	56
Gambar 12. <i>Activity Diagram</i> Kelola Data Program Studi oleh Admin Universitas	57
Gambar 13. <i>Avtivity</i> Melihat Hasil Perhitungan oleh Admin Universitas	58
Gambar 14. <i>Activity Diagram</i> Kelola Akun oleh Admin Program Studi	59
Gambar 15. <i>Activity Diagram</i> Kelola Data Mahasiswa oleh Admin Program Studi	60
Gambar 16. <i>Activity Diagram</i> Kelola Matakuliah Oleh Admin Program Studi ...	61
Gambar 17. <i>Activity Diagram</i> Kelola Kelas Matakuliah oleh Admin Program Studi	62
Gambar 18. <i>Activity Diagram</i> Kelola Kelas Mahasiswa oleh Admin Program Studi	63
Gambar 19. <i>Activity Diagram</i> Kelola Data CPMK oleh Admin Program Studi ..	64
Gambar 20. <i>Activity Diagram</i> Kelola Data CPL oleh Admin Program Studi	65
Gambar 21. <i>Activity Diagram</i> Kelola Pemetaan oleh Admin Program Studi.....	66
Gambar 22. <i>Activity Diagram</i> Kelola Penilaian oleh Admin Program Studi	67

Gambar 23. <i>Activity Diagram</i> Melihat Hasil Perhitungan oleh Admin Program Studi	68
Gambar 24. <i>Activity Diagram</i> Melihat Data CPL dan CPMK oleh Kepala Program Studi	69
Gambar 25. <i>Activity Diagram</i> Melihat Pemetaan oleh Kepala Program Studi.....	70
Gambar 26. <i>Activity Diagram</i> Melihat Hasil Perhitungan oleh Kepala Program Studi	71
Gambar 27. <i>Activity Diagram</i> Kelola Atur Penilaian oleh Dosen	72
Gambar 28. <i>Activity Diagram</i> Kelola Penilaian oleh Dosen	73
Gambar 29. <i>Activity Diagram</i> Melihat Hasil Perhitungan Dosen	74
Gambar 30. <i>Sequence Diagram</i> Pengelolaan Data	75
Gambar 31. Arsitektur sistem website	76
Gambar 32. <i>Kanban Board</i>	77
Gambar 33. Halaman login	79
Gambar 34. <i>Hook: useLogin</i>	79
Gambar 35. <i>Service: authService</i>	80
Gambar 36. <i>Context: AuthContext</i>	80
Gambar 37. <i>PrivateRoutes (Role-Based Access)</i>	81
Gambar 38. <i>Set Data Logout</i>	82
Gambar 39. Halaman Dashboard	82
Gambar 40. Halaman Data Akun	83
Gambar 41. <i>Form modal</i> tambah akun	83
Gambar 42. <i>Form modal</i> edit akun	84
Gambar 43. <i>Modal</i> hapus akun	85
Gambar 44. Halaman Data Program Studi.....	85
Gambar 45. <i>Modal</i> Tambah data program studi	86
Gambar 46. <i>Modal Edit</i> data program studi.....	87
Gambar 47. <i>Modal</i> Hapus data program studi	87
Gambar 48. Halaman Hasil Perhitungan.....	88
Gambar 49. Halaman <i>Dashboard</i> Admin Prodi.....	89
Gambar 50. Tampilan distribusi mata kuliah.....	90
Gambar 51. Tampilan detail capaian kelas	90

Gambar 52. Tampilan detail capaian mahasiswa	91
Gambar 53. Halaman Data Akun	92
Gambar 54. <i>Form modal</i> tambah akun	92
Gambar 55. <i>Form modal edit</i> akun	93
Gambar 56. <i>Form modal</i> hapus akun	93
Gambar 57. Tampilan data mahasiswa	94
Gambar 58. <i>Form modal</i> tambah mahasiswa	95
Gambar 59. <i>Form modal</i> edit mahasiswa	95
Gambar 60. <i>Form</i> hapus data mahasiswa	96
Gambar 61. Tampilan data mata kuliah	97
Gambar 62. <i>Form modal</i> tambah data mata kuliah	97
Gambar 63. <i>Form modal edit</i> data mata kuliah	98
Gambar 64. <i>Form modal</i> hapus data mata kuliah	98
Gambar 65. Tampilan data CPMK	99
Gambar 66. <i>Form modal</i> tambah data CPMK	99
Gambar 67. <i>Form modal edit</i> data CPMK	100
Gambar 68. <i>Form modal</i> hapus data CPMK	101
Gambar 69. Tampilan Data Kelas	101
Gambar 70. <i>Form modal</i> tambah data Kelas	102
Gambar 71. <i>Form modal</i> tambah data Kelas	102
Gambar 72. <i>Form modal</i> tambah data Kelas	103
Gambar 73. Halaman Data CPL	104
Gambar 74. <i>Form modal</i> tambah data CPL	104
Gambar 75. <i>Form modal edit</i> data CPL	105
Gambar 76. <i>Modal</i> hapus data CPL	106
Gambar 77. Halaman Pemetaan	106
Gambar 78. Tampilan pemetaan data CPL	107
Gambar 79. <i>Modal</i> tambah pemetaan data CPL	107
Gambar 80. <i>Modal edit</i> pemetaan data CPL	108
Gambar 81. <i>Modal</i> hapus pemetaan data CPL	108
Gambar 82. Tampilan pemetaan data CPMK	109
Gambar 83. <i>Modal</i> tambah pemetaan data CPMK	110

Gambar 84. <i>Modal edit</i> pemetaan data CPMK	110
Gambar 85. <i>Modal</i> hapus pemetaan data CPMK.....	111
Gambar 86. Halaman Atur Penilaian	112
Gambar 87. Tampilan Atur Penilaian	112
Gambar 88. Tampilan Atur Penilaian	113
Gambar 89. Halaman Hasil Perhitungan.....	114
Gambar 90. Halaman <i>Dashboard</i>	115
Gambar 91. Halaman Detail Kelas.....	115
Gambar 92. Detail Capaian Mahasiswa	116
Gambar 93. Halaman Atur Penilaian	117
Gambar 94. Halaman Penilaian.....	117
Gambar 95. Halaman Hasil Perhitungan.....	118
Gambar 96. Halaman <i>Dashboard</i>	119
Gambar 97. Tampilan distribusi mata kuliah.....	119
Gambar 98. Tampilan detail capaian kelas	120
Gambar 99. Tampilan detail capaian mahasiswa	121
Gambar 100. Halaman Data CPL.....	121
Gambar 101. Halaman Data CPMK.....	122
Gambar 102. Halaman Pemetaan	122
Gambar 103 Halaman Pemetaan	123

DAFTAR TABEL

	Halaman
Tabel 1. <i>Control Node</i> pada <i>Activity Diagram</i>	34
Tabel 2. <i>Graphic node sequence diagram</i>	36
Tabel 3. Waktu Pelaksanaan Penelitian	44
Tabel 4. Perangkat Keras	45
Tabel 5. Perangkat Lunak	45
Tabel 6. Kebutuhan Fungsional	50
Tabel 7. <i>Authentikasi</i> Testing.....	124
Tabel 8. Testing Halaman <i>Dashboard</i>	125
Tabel 9. Testing Halaman Hasil Perhitungan	126
Tabel 10. Testing Halaman Kelola Akun.....	127
Tabel 11. Testing Halaman Kelola Program Studi	129
Tabel 12. Testing Halaman <i>Dashboard</i> Admin Program Studi.....	131
Tabel 13. Testing Halaman Kelola Akun.....	131
Tabel 14. Testing Halaman Kelola Data Mahasiswa	134
Tabel 15. Testing Halaman Kelola Data Mahasiswa	136
Tabel 16. Testing Halaman Kelola Data CPMK.....	138
Tabel 17. Testing Halaman Kelola Data Kelas.....	140
Tabel 18. Testing Halaman Kelola Data Kelas.....	143
Tabel 19. Testing Halaman kelola Data CPL.....	145
Tabel 20. Testing Halaman Kelola Data Pemetaan	148
Tabel 21. Testing Halaman Kelola Atur Penilaian	150
Tabel 22. Testing Halaman Kelola Atur Pembobotan-Penilaian	151
Tabel 23. Testing Halaman Hasil Perhitungan	152
Tabel 24. Testing Halaman <i>Dashboard</i> Dosen	152
Tabel 25. Testing Halaman Atur Penilaian	153
Tabel 26. Testing Halaman Penilaian	153
Tabel 27. Testing Halaman Penilaian	154

Tabel 28. Testing Halaman Hasil Perhitungan	155
Tabel 29. Testing Halaman <i>Dashboard</i> Kepala Program Studi.....	155
Tabel 30. Testing Halaman Data CPL dan CPMK	156
Tabel 31. Testing Halaman Pemetaan.....	156
Tabel 32. Testing Halaman Hasil Perhitungan	157
Tabel 33. Tabel Total Pengujian	158
Tabel 34 Progres <i>Development</i> Kebutuhan <i>Fungsional</i>	159
Tabel 35. Dokumentasi API Autentikasi Terimplementasi	165
Tabel 36. Dokumentasi API Admin Universitas Terimplementasi	165
Tabel 37. Dokumentasi API Admin Program Studi Terimplementasi	167
Tabel 38. Dokumentasi API Admin Universitas yang Belum Terimplementasi	173
Tabel 39. Dokumentasi API Admin Program Studi yang Belum Terimplementasi	173
Tabel 40. Dokumentasi API Dosen yang Belum Terimplementasi	173
Tabel 41. Dokumentasi API Kaprodi yang Belum Terimplementasi	175
Tabel 42 Saran Rencana Pengembangan API Sistem	176

I. PENDAHULUAN

1.1 Latar Belakang

Pendidikan tinggi memiliki peranan penting dalam menghasilkan sumber daya manusia yang berkualitas dan berdaya saing. Untuk mencapai tujuan tersebut, setiap program studi perlu memastikan bahwa capaian pembelajaran lulusannya selaras dengan standar yang ditetapkan. Capaian pembelajaran mencakup aspek pengetahuan, keterampilan, dan sikap yang harus dimiliki mahasiswa setelah menyelesaikan suatu program pendidikan tertentu[1].

Dalam praktiknya, pengolahan dan penyajian data Capaian Pembelajaran Lulusan (CPL) masih belum dilakukan secara sistematis pada program studi di Universitas Lampung. Kondisi ini menyebabkan proses pemantauan CPL belum memiliki dukungan sistem yang memadai untuk melakukan pencatatan, perhitungan, maupun penyajian informasi secara terstruktur. Padahal, data CPL merupakan elemen penting dalam penyusunan laporan akademik serta dalam pemenuhan standar penjaminan mutu internal perguruan tinggi.

Standar Nasional Pendidikan Tinggi (SN-Dikti) menetapkan bahwa CPL menjadi salah satu dasar utama dalam pengembangan, pelaksanaan, dan evaluasi kurikulum. Selain itu, CPL juga merupakan komponen yang digunakan dalam proses akreditasi program studi oleh BAN-PT melalui pendekatan *outcome-based accreditation* yang berfokus pada ketercapaian hasil belajar lulusan[1]. Dengan demikian, diperlukan sistem yang mampu mengakomodasi pengolahan dan penyajian data CPL secara terstruktur.

Untuk memenuhi kebutuhan tersebut, penelitian ini berfokus pada perancangan dan pembangunan *frontend* sistem informasi CPL berbasis web menggunakan *React.js*.

React.js dipilih karena memiliki keunggulan dalam pengembangan antarmuka pengguna, seperti penggunaan *Virtual DOM* yang dapat meningkatkan performa, pendekatan berbasis komponen yang memudahkan pengelolaan elemen antarmuka, serta sintaks deklaratif yang mempermudah pembuatan tampilan interaktif[2]. Pendekatan ini mendukung pengembangan antarmuka yang, modular, dan mudah dipelihara.

Selain itu, metode *black-box testing* digunakan untuk memastikan bahwa fungsi-fungsi pada *frontend* berjalan sesuai dengan spesifikasi tanpa memerlukan pemahaman terhadap proses internal perangkat lunak. Pengujian ini berfokus pada validasi masukan dan keluaran untuk memastikan bahwa setiap bagian sistem bekerja sebagaimana mestinya [3].

Melalui pendekatan tersebut, penelitian ini menghasilkan *frontend* sistem informasi CPL yang dapat digunakan sebagai dasar bagi pengembangan sistem yang lebih lengkap di masa mendatang. Penelitian ini tidak berfokus pada evaluasi efektivitas sistem terhadap proses akademik atau akreditasi, melainkan pada penyediaan artefak teknis berupa rancangan antarmuka berbasis web yang sesuai dengan kebutuhan fungsional yang telah ditentukan.

1.2 Rumusan Masalah

Adapun rumusan masalah dari penelitian ini diantaranya adalah :

1. Bagaimana melakukan analisis kebutuhan fungsional dan nonfungsional serta merancang arsitektur *frontend* sistem informasi berbasis web untuk pengolahan dan penyajian data Capaian Pembelajaran Lulusan (CPL) sesuai dengan kebutuhan program studi?
2. Bagaimana menerapkan metode *Agile Kanban* dalam proses pembangunan *frontend* sistem informasi tersebut?

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah :

1. Menganalisis kebutuhan fungsional dan nonfungsional pada *frontend* Sistem Informasi Capaian Pembelajaran Lulusan (CPL).
2. Merancang arsitektur *frontend* dan struktur komponen sistem berbasis web berdasarkan desain antarmuka yang telah tersedia.
3. Membangun *frontend* sistem informasi CPL berbasis web dengan menerapkan metode *Agile Kanban*.

1.4 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah :

1. Penelitian ini memberikan kontribusi berupa rancangan dan implementasi *frontend* Sistem Informasi Capaian Pembelajaran Lulusan (CPL) yang dapat dijadikan sebagai referensi atau *prototype* dalam pengembangan sistem pada lingkungan program studi maupun institusi terkait.
2. Penelitian ini menghasilkan dokumentasi teknis yang dapat dimanfaatkan sebagai dasar evaluasi serta pengembangan lebih lanjut terhadap sistem informasi CPL di masa mendatang.

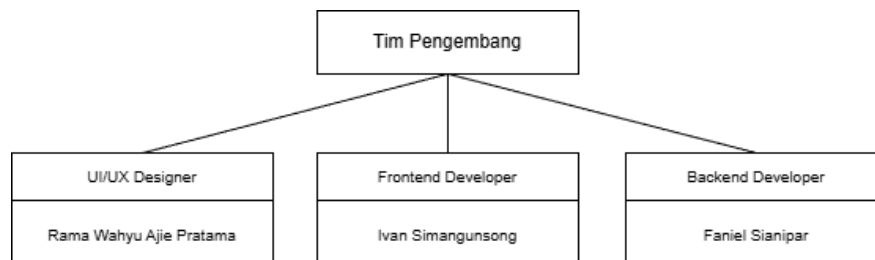
1.5 Batasan Masalah

Adapun batasan masalah pada penelitian ini di antaranya adalah :

1. Penelitian ini berfokus pada perancangan dan pembangunan *frontend* Sistem Informasi Capaian Pembelajaran Lulusan (CPL) berbasis web tanpa mencakup pengembangan *backend*, integrasi data nyata, maupun implementasi operasional pada tingkat program studi.
2. Data yang digunakan dalam proses perancangan dan pengujian merupakan data contoh (*dummy data*) dan tidak merepresentasikan data akademik resmi.

1.6 Tim Penelitian

Dalam proses pengembangan sistem informasi ini, peneliti tidak bekerja sendiri, melainkan melibatkan sebuah tim yang berperan dalam proses perancangan sistem. Tim penelitian terdiri dari beberapa anggota dengan pembagian tugas yang jelas sesuai dengan keahlian masing-masing, yaitu:



Gambar 1. Tim Pengembang

1. Rama Wahyu Ajie Pratama bertanggung jawab sebagai *UI/UX Designer*, yang berperan dalam merancang antarmuka pengguna serta memastikan pengalaman penggunaan aplikasi berjalan dengan baik dan sesuai dengan prinsip *usability*.
2. Ivan Simangunsong bertugas sebagai *Frontend Developer*, yang mengimplementasikan rancangan antarmuka ke dalam bentuk halaman web yang fungsional serta mengintegrasikannya dengan *API backend*.
3. Faniel Sianipar berperan sebagai *Backend Developer*, yang bertanggung jawab dalam pengembangan *API*, pengelolaan basis data, serta memastikan proses bisnis dapat berjalan dengan baik pada sisi server.

Dengan adanya pembagian tugas tersebut, proses pengembangan sistem dapat dilaksanakan secara lebih terstruktur, efisien, dan selaras dengan alur metodologi penelitian yang digunakan.

1.7 Sistematika Penulisan

Sistematika penulisan skripsi ini terdiri dari 5 bab sebagai berikut :

- BAB I : PENDAHULUAN**
 Bab ini berisi latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, tim penelitian dan sistematika penulisan.
- BAB II : TINJAUAN PUSTAKA**
 Bab ini berisi dasar-dasar teori yang digunakan sebagai landasan dalam penelitian ini.
- BAB III : METODOLOGI PENELITIAN**
 Bab ini berisi tentang waktu dan tempat penelitian, alat dan bahan penelitian, serta tahapan penelitian.
- BAB IV : HASIL DAN PEMBAHASAN**
 Bab ini berisi pembahasan mengenai perancangan dari website yang dibuat.
- BAB V : KESIMPULAN DAN SARAN**
 Bab ini memuat kesimpulan berdasarkan hasil pembahasan yang dilakukan dalam penelitian ini. Serta saran yang diharapkan dapat meningkatkan wawasan serta kemajuan bagi kita Bersama.
- DAFTAR PUSTAKA : Bab ini memuat daftar sumber kutipan teori-teori yang dijadikan acuan dalam menulis laporan.**

II. TINJAUAN PUSTAKA

2.1 Capaian Pembelajaran Lulusan (CPL)

Capaian Pembelajaran Lulusan (CPL) merupakan pernyataan mengenai kemampuan yang diharapkan dimiliki oleh lulusan setelah menyelesaikan seluruh proses pembelajaran pada suatu program studi. CPL mencerminkan hasil belajar akhir yang mencakup aspek pengetahuan, keterampilan, serta sikap dan tata nilai, yang dirumuskan sebagai tujuan pendidikan dan menjadi acuan utama dalam penyelenggaraan proses pembelajaran di perguruan tinggi[4].

Dalam implementasinya, CPL dijabarkan ke dalam beberapa Capaian Pembelajaran Mata Kuliah (CPMK) yang melekat pada mata kuliah yang ditempuh oleh mahasiswa. CPMK berfungsi sebagai indikator ketercapaian CPL, sehingga pencapaian CPL dapat diukur secara tidak langsung melalui hasil penilaian CPMK pada setiap mata kuliah. Oleh karena itu, hubungan antara CPL dan CPMK bersifat hierarkis dan saling terintegrasi, di mana ketercapaian CPL sangat bergantung pada pencapaian CPMK yang relevan[5].

Untuk menunjang proses asesmen CPL dan CPMK, diperlukan indikator performansi yang terukur serta metode penilaian yang tepat. Indikator tersebut digunakan untuk memastikan bahwa proses evaluasi pembelajaran dilakukan secara objektif, konsisten, dan dapat dipertanggungjawabkan. Penilaian yang akurat terhadap indikator performansi menjadi faktor penting dalam menghasilkan data capaian pembelajaran yang valid dan reliabel, sehingga dapat dimanfaatkan sebagai dasar evaluasi mutu pembelajaran dan pengambilan keputusan akademik.

2.1.1 Alur Perhitungan CPL

Perhitungan Capaian Pembelajaran Lulusan (CPL) didasarkan pada pencapaian Capaian Pembelajaran Mata Kuliah (CPMK) dari mata kuliah yang diambil oleh

mahasiswa. Nilai CPL diperoleh melalui proses pengolahan data hasil penilaian mahasiswa yang mencakup nilai tugas, Ujian Tengah Semester (UTS), dan Ujian Akhir Semester (UAS). Secara umum, alur perhitungan CPL dilakukan melalui beberapa tahapan sebagai berikut [4]:

1. Menentukan representasi bobot nilai CPMK terhadap CPL pada setiap mata kuliah. Bobot ini mencerminkan tingkat kontribusi masing-masing CPMK terhadap keseluruhan capaian pembelajaran lulusan.
2. Mengumpulkan data nilai mahasiswa dari setiap mata kuliah yang memiliki keterkaitan dengan CPL.
3. Melakukan perhitungan nilai CPMK berdasarkan data nilai mahasiswa yang diperoleh.
4. Menghitung nilai CPL dengan cara mengakumulasikan nilai CPMK yang telah dikalikan dengan bobot masing-masing.

Secara matematis, perhitungan nilai CPL dapat dirumuskan sebagai berikut[4]:

$$\text{Nilai CPL} = \frac{(\text{Nilai CPMK} \times \text{bobot}) + (...) + (...)}{\text{Jumlah keseluruhan bobot}}$$

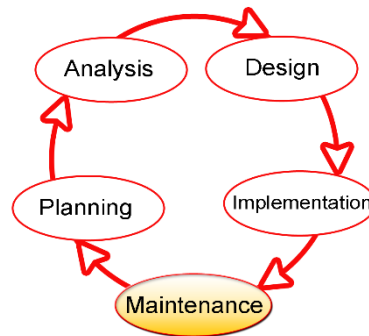
Gambar 2. Rumus Perhitungan Nilai CPL

Rumus capaian pembelajaran lulusan (CPL) dihitung dengan menggunakan rata-rata tertimbang dari capaian pembelajaran mata kuliah (CPMK). Nilai CPMK dikalikan dengan bobot masing-masing, kemudian dijumlahkan dan dibagi dengan total bobot. Dengan cara ini, CPMK yang memiliki bobot lebih besar akan lebih berpengaruh terhadap nilai CPL dibandingkan CPMK dengan bobot kecil.

2.2 *Software Development Life Cycle (SDLC)*

Software Development Life Cycle (SDLC) merupakan serangkaian tahapan yang terstruktur dan sistematis dalam proses pengembangan maupun modifikasi sistem perangkat lunak. Proses ini melibatkan penggunaan berbagai model dan metodologi yang dirancang untuk membantu pengembang dalam merancang, membangun, menguji, serta menerapkan perangkat lunak secara efisien dan efektif. Setiap

tahapan dalam SDLC dirancang untuk memastikan bahwa perangkat lunak yang dihasilkan mampu memenuhi kebutuhan pengguna dan berfungsi sesuai dengan spesifikasi yang telah ditetapkan[6].



Gambar 3. *Software Development Life Cycle*

2.3 Agile

Agile merupakan sekumpulan metode dan pendekatan yang dirancang untuk meningkatkan efektivitas berpikir, efisiensi kerja, serta kualitas pengambilan keputusan. Metode dan pendekatan ini mencakup berbagai aspek dalam rekayasa perangkat lunak, seperti manajemen proyek, desain dan arsitektur perangkat lunak, serta peningkatan proses pengembangan. Beberapa varian populer dari pendekatan *Agile* meliputi *Scrum*, *Kanban*, *Extreme Programming (XP)*, *Agile Model*, serta berbagai metode lainnya[7].

Kanban merupakan metode perbaikan proses yang digunakan oleh tim-tim yang mengadopsi pendekatan *Agile*. Tim yang menerapkan *Kanban* memulai dengan memahami proses yang digunakan dalam pengembangan perangkat lunak, kemudian melakukan perbaikan secara berkelanjutan seiring waktu. Seperti halnya *Scrum* dan *Extreme Programming (XP)*, *Kanban* memerlukan pola pikir tertentu, khususnya yang berbasis *lean*. Tim yang menggunakan *Kanban* menerapkan pola pikir *lean* dalam aktivitas kerja mereka, sehingga memberikan landasan yang kuat bagi upaya perbaikan proses.

Melalui metode *Kanban*, tim berfokus pada upaya mengidentifikasi dan menghilangkan pemborosan dalam alur kerja. Meskipun beberapa tim telah menerapkan pola pikir *lean* dalam pengembangan perangkat lunak tanpa

menggunakan *Kanban*, metode ini tetap menjadi salah satu pendekatan yang paling umum dan efektif dalam mengintegrasikan pola pikir *lean* ke dalam organisasi[7].

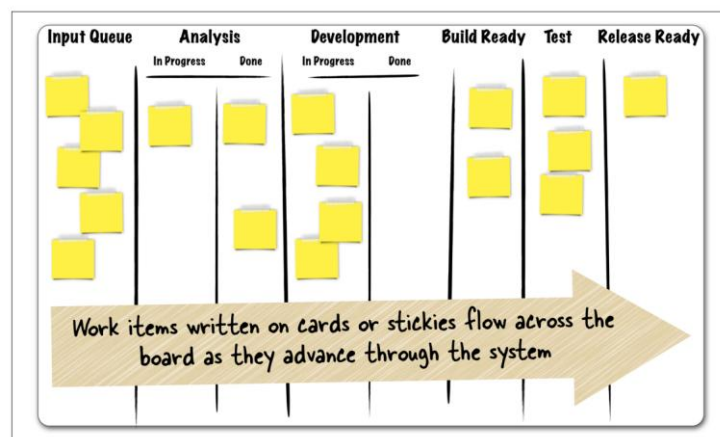
Proses pengembangan perangkat lunak dengan menggunakan metodologi *Kanban* melibatkan serangkaian langkah dan praktik yang dirancang untuk meningkatkan efisiensi, mengurangi pemborosan, serta memastikan aliran kerja yang berjalan secara optimal. Berikut merupakan langkah-langkah yang dapat diterapkan dalam metodologi *Kanban*:

1. Evaluasi Proses Saat Ini

Langkah awal dalam penerapan metodologi *Kanban* adalah melakukan peninjauan terhadap cara pengelolaan tugas dan proyek yang berjalan saat ini. Pada tahap ini dilakukan identifikasi terhadap setiap tahapan kerja, mulai dari awal hingga akhir proyek, serta permasalahan atau hambatan yang sering muncul dalam proses pengembangan..

2. Visualisasikan Alur Kerja

Tahap selanjutnya adalah memvisualisasikan alur kerja melalui penggunaan papan *Kanban*. Papan *Kanban* umumnya dibagi ke dalam beberapa kolom yang merepresentasikan tahapan dalam proses pengembangan, seperti *To Do*, *In Progress*, dan *Done*. Setiap tugas atau pekerjaan direpresentasikan dalam bentuk kartu yang dipindahkan dari satu kolom ke kolom lainnya seiring dengan perkembangan pengerjaan.



Gambar 4. Contoh Papan Kanban[7]

3. Batasi Pekerjaan dalam Proses (*Work in Progress/WIP*)

Pada tahap ini ditetapkan batasan jumlah pekerjaan yang dapat berada pada kolom *In Progress* dalam satu waktu. Sebagai contoh, jumlah *WIP* dapat dibatasi menjadi satu atau dua tugas. Pembatasan ini bertujuan untuk menjaga fokus tim dalam menyelesaikan pekerjaan yang sedang berjalan sebelum memulai tugas baru, sehingga dapat mengurangi penumpukan pekerjaan dan meningkatkan efisiensi penyelesaian tugas.

4. Kelola Aliran Kerja (*Manage Flow*)

Pengelolaan aliran kerja dilakukan dengan memantau pergerakan tugas pada papan *Kanban* untuk memastikan bahwa setiap pekerjaan dapat berpindah secara lancar dari satu tahap ke tahap berikutnya. Apabila terdapat tugas yang tertahan terlalu lama pada kolom *In Progress*, maka perlu dilakukan identifikasi terhadap penyebab hambatan tersebut dan dilakukan tindakan perbaikan yang sesuai.

5. Buat Kebijakan Proses Eksplisit

Pada tahap ini, kebijakan dan aturan kerja didokumentasikan secara jelas untuk setiap tahapan proses. Sebagai contoh, ditetapkan kriteria yang harus dipenuhi agar suatu tugas dapat dipindahkan dari kolom *In Progress* ke *Done*. Kebijakan proses yang eksplisit ini berfungsi untuk menjaga konsistensi pelaksanaan pekerjaan serta memastikan bahwa seluruh tahapan penting telah dilalui dengan benar.

6. Terapkan Umpan Balik Berkala

Umpan balik dilakukan secara rutin melalui evaluasi berkala terhadap kemajuan pekerjaan. Misalnya, pada akhir setiap minggu dilakukan peninjauan terhadap tugas yang telah diselesaikan, kendala yang dihadapi, serta aspek-aspek yang masih dapat ditingkatkan. Penerapan umpan balik secara berkelanjutan memungkinkan proses kerja untuk terus diperbaiki dari waktu ke waktu.

7. Perbaiki Secara Kolaboratif, Berkembang secara Eksperimental

Tahap terakhir adalah melakukan perbaikan proses secara kolaboratif dengan seluruh anggota tim. Apabila produktivitas dinilai belum optimal, tim dapat mencoba metode kerja atau alat pendukung baru, kemudian mengevaluasi hasil penerapannya. Pendekatan ini dilakukan secara eksperimental dengan menetapkan hipotesis, melakukan pengujian, serta mengevaluasi hasilnya sebagai dasar untuk peningkatan proses kerja selanjutnya.

Dalam kerangka kerja *Kanban* yang berbasis *Agile*, tidak terdapat tahapan formal seperti pada metodologi *Agile* lainnya, misalnya *Scrum*. Meskipun demikian, terdapat aliran kerja umum yang lazim digunakan dalam penerapan *Kanban*. Berikut adalah aliran kerja yang umum dalam *Kanban Agile* [8]:

1. *Backlog*

Pada tahap *backlog*, seluruh tugas diidentifikasi dan diuraikan menjadi pekerjaan yang lebih terperinci, kemudian ditentukan prioritas masing-masing tugas. Seluruh tugas yang berkaitan dengan pengembangan sistem dikumpulkan pada tahap ini dan diurutkan berdasarkan tingkat prioritas pekerjaan.

2. *To do*

Setelah seluruh ide dan tugas terkumpul dalam *backlog*, tugas-tugas tersebut dipindahkan ke tahap *to do* untuk menandai pekerjaan yang siap dilaksanakan.

3. *In progress*

Tugas yang telah dipilih untuk dikerjakan dipindahkan dari tahap *to do* ke tahap *in progress*. Pada tahap ini, pekerjaan sedang dilakukan sesuai dengan prioritas yang telah ditentukan. Untuk meningkatkan efisiensi dan mengoptimalkan alur kerja, terutama ketika jumlah tugas cukup banyak, perlu ditetapkan batasan pekerjaan yang sedang berjalan (*work in progress*), sehingga tim dapat fokus menyelesaikan tugas secara sistematis.

4. *Feedback*

Pada tahap *feedback*, dosen memberikan umpan balik berupa masukan, koreksi, atau identifikasi permasalahan terhadap tugas-tugas yang telah dikerjakan dan diuji. Umpan balik ini menjadi dasar untuk melakukan perbaikan sebelum tugas dinyatakan selesai.

5. *Done*

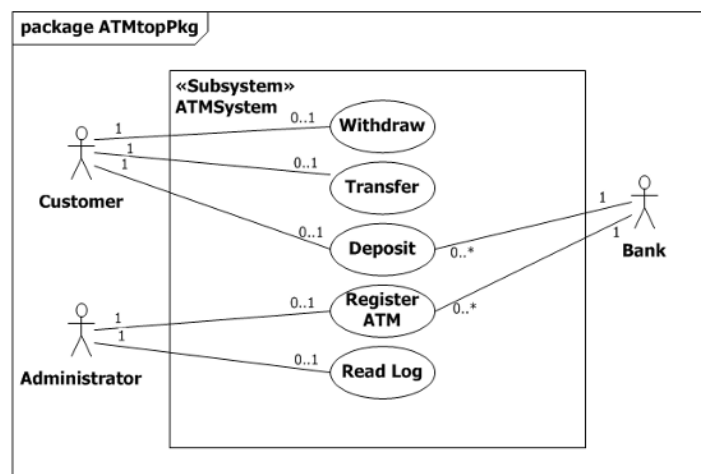
Suatu tugas dinyatakan selesai dan dipindahkan ke tahap *done* apabila hasil pekerjaan telah memenuhi persyaratan dan standar pengembangan yang ditetapkan, pengujian fungsional telah berhasil dilakukan, serta tidak ditemukan kesalahan pada tugas tersebut.

2.4 UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) merupakan salah satu standar bahasa pemodelan yang banyak digunakan di industri untuk mendefinisikan kebutuhan sistem, melakukan analisis dan desain, serta menggambarkan arsitektur sistem dalam pengembangan perangkat lunak berbasis pemrograman berorientasi objek. UML dikembangkan sebagai respons terhadap kebutuhan akan pemodelan visual yang mampu digunakan untuk menspesifikasikan, menggambarkan, membangun, serta mendokumentasikan sistem perangkat lunak secara terstruktur dan sistematis[6].

2.4.1 Use Case Diagram

Use Case merupakan salah satu cara untuk menangkap kebutuhan sistem dengan mendefinisikan apa saja yang seharusnya dilakukan oleh sistem. Dalam konsep *Use Case* terdapat tiga elemen utama, yaitu aktor, *use case*, dan subjek. Subjek merepresentasikan sistem yang sedang dikaji, sedangkan aktor merupakan pengguna atau sistem lain yang berinteraksi dengan subjek tersebut. *Use case* dipahami sebagai spesifikasi perilaku yang menggambarkan layanan atau fungsi yang disediakan sistem bagi aktor. Setiap kejadian nyata dari perilaku tersebut disebut sebagai *instance of use case*, yang umumnya dijelaskan melalui rangkaian interaksi antara aktor dan sistem.



Gambar 5. Class diagram of a Package owning a set of UseCases, Actors, and a Subsystem[9]

Diagram *Use Case* ATM menggambarkan sistem *ATM System* sebagai subjek yang menyediakan layanan kepada aktor eksternal, yaitu *Customer*, *Administrator*, dan *Bank*. *Customer* berinteraksi dengan sistem melalui *use case* *Withdraw*, *Transfer*, dan *Deposit*, sementara *Administrator* berhubungan dengan *use case* *Register ATM* dan *Read Log*. *Bank* berperan dalam *use case* *Deposit* dan *Register ATM*, dengan kemungkinan berpartisipasi dalam banyak kejadian secara bersamaan.

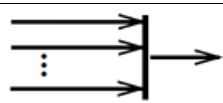
Garis penghubung antara aktor dan *use case* menunjukkan adanya asosiasi, yaitu relasi yang menandakan interaksi atau partisipasi aktor terhadap layanan yang disediakan oleh sistem. Dengan demikian, diagram ini menekankan bagaimana aktor-aktor eksternal memicu perilaku sistem ATM sesuai dengan kebutuhan dan peran masing-masing[9].

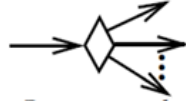
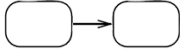
2.4.2 Activity Diagram

Activity Diagram merupakan diagram yang digunakan untuk memodelkan aktivitas atau alur kerja (*workflow*) dalam suatu sistem atau organisasi. Diagram ini menggambarkan rangkaian aktivitas yang terjadi dalam suatu proses, baik yang berasal dari dalam sistem maupun dari luar sistem. *Activity Diagram* banyak digunakan dalam rekayasa proses bisnis dan *workflow*, serta dalam pemodelan sistem informasi untuk menjelaskan proses-proses pada tingkat sistem secara terstruktur dan sistematis.

Dalam perancangan *Activity Diagram* terdapat elemen yang disebut *control node*. *Control node* berfungsi sebagai simpul pengendali dalam *Activity Diagram* yang menentukan bagaimana aliran proses berpindah dari satu aktivitas ke aktivitas lainnya. Terdapat beberapa jenis *control node* dengan fungsi yang berbeda, antara lain *Initial Node* sebagai titik awal proses, *Final Node* sebagai titik akhir proses, *Fork Node* untuk memecah alur menjadi proses paralel, *Join Node* untuk menggabungkan kembali alur paralel, *Merge Node* untuk menggabungkan percabangan alternatif, serta *Decision Node* yang digunakan untuk menentukan percabangan berdasarkan kondisi tertentu[9]. Rincian *control node* yang terdapat pada *activity diagram* disajikan dalam bentuk tabel 1 berikut:

Tabel 1. *Control Node* pada *Activity Diagram*

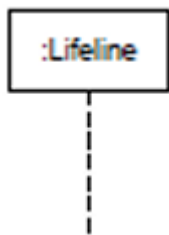
<i>Initial Note</i>	<i>InitialNode</i> adalah sebuah <i>ControlNode</i> yang berfungsi sebagai titik awal untuk menjalankan sebuah <i>Activity</i> . Satu <i>Activity</i> dapat memiliki lebih dari satu <i>InitialNode</i> .	
<i>Final Nodes</i>	<i>FinalNode</i> adalah sebuah <i>ControlNode</i> tempat aliran dalam sebuah <i>Activity</i> berhenti.	
<i>Fork Nodes</i>	<i>ForkNode</i> adalah sebuah <i>ControlNode</i> yang membagi satu aliran menjadi beberapa aliran yang berjalan secara bersamaan (<i>konkuren</i>).	
<i>Join Nodes</i>	<i>JoinNode</i> adalah sebuah <i>ControlNode</i> yang menyinkronkan beberapa aliran.	
<i>Merge Nodes</i>	<i>MergeNode</i> adalah sebuah <i>control node</i> yang menggabungkan beberapa aliran tanpa melakukan sinkronisasi.	

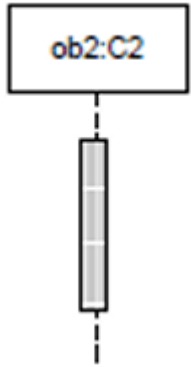
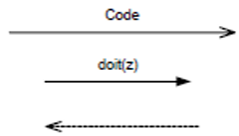
<i>Decision Nodes</i>	<i>DecisionNode</i> adalah sebuah <i>control node</i> yang memilih di antara aliran keluar.	
<i>Control Flow</i>	Suatu <i>Control flow</i> dinotasikan dengan garis berpanah yang menghubungkan dua aksi.	

2.4.3 Sequence Diagram

Sequence Diagram merupakan jenis *interaction diagram* yang paling umum digunakan dalam UML. Diagram ini berfokus pada pertukaran pesan antar objek atau *lifeline* dengan menekankan urutan pesan yang terjadi. Melalui *Sequence Diagram*, interaksi antar komponen sistem dapat digambarkan secara jelas, termasuk spesifikasi kejadian (*occurrence specifications*) yang muncul pada setiap *lifeline*. Interaksi yang ditunjukkan oleh *Sequence Diagram* menjadi dasar penting untuk memahami makna dan semantik dari kelas-kelas yang terdapat dalam paket *Interactions*. Tabel 2 berikut menunjukkan berbagai *graphic node* yang dapat digunakan dalam *Sequence Diagram*[9].

Tabel 2. *Graphic node sequence diagram*

<i>Lifeline</i>	<i>lifeline</i> adalah simbol dalam <i>sequence diagram</i> yang merepresentasikan keberadaan (<i>lifetime</i>) suatu partisipan atau objek selama interaksi berlangsung.	
-----------------	---	---

<i>Execution Specification</i>	<i>Execution Specification</i> adalah blok eksekusi pada <i>lifeline</i> yang menandai kapan sebuah aksi berlangsung, dimulai dari penerimaan pesan hingga berakhir dengan pengiriman balasan atau hasil.	
<i>Message</i>	<i>Message</i> dalam <i>sequence diagram</i> adalah garis komunikasi yang menghubungkan pengirim (<i>sender MessageEnd</i>) dengan penerima (<i>receiver MessageEnd</i>).	

2.5 Blackbox Testing

Blackbox Testing merupakan metode pengujian perangkat lunak yang berfokus pada fungsi eksternal sistem tanpa memperhatikan kode program. Penguji hanya menggunakan spesifikasi fungsional untuk menentukan input dan output yang diharapkan. Tujuan dari pengujian ini adalah memastikan bahwa sistem memberikan hasil yang sesuai dengan kebutuhan pengguna. Dengan demikian, *Blackbox Testing* efektif digunakan untuk memvalidasi apakah sistem telah memenuhi persyaratan yang ditentukan.

Salah satu teknik dalam *Blackbox Testing* adalah *Equivalence Partitioning* (EP). Teknik ini membagi domain input ke dalam kelas-kelas ekuivalen, yaitu kelompok data yang diasumsikan diproses dengan cara yang sama oleh sistem. Dari setiap kelas, cukup diuji satu data perwakilan untuk mewakili keseluruhan kelas. Pendekatan ini membantu mengurangi jumlah *test case* tanpa mengurangi cakupan pengujian. Dengan menggunakan EP, penguji dapat bekerja lebih efisien dalam menemukan kesalahan, terutama pada input valid dan tidak valid yang sering menjadi sumber kesalahan[3].

Analisis hasil pengujian dilakukan dengan menghitung tingkat keberhasilan pengujian menggunakan metrik standar pengujian perangkat lunak, yaitu persentase *test case* yang berhasil (*test case pass rate*). Metrik ini digunakan untuk memberikan gambaran kuantitatif mengenai tingkat kesesuaian fungsional sistem terhadap kebutuhan pengguna. Perhitungan persentase keberhasilan pengujian dilakukan menggunakan rumus pada gambar 6 berikut:

$$Test\ Case\ Pass = \left(\frac{Test\ Case\ Passed}{Total\ Test\ Case} \right) \times 100\%$$

Gambar 6. Rumus pengujian *Test Case Pass*[10]

Nilai persentase yang dihasilkan menunjukkan tingkat keberhasilan sistem dalam memenuhi kebutuhan fungsional yang diuji. Semakin tinggi persentase *test case* yang berhasil, maka semakin baik tingkat kesesuaian fungsi sistem terhadap spesifikasi yang telah ditetapkan. Sebaliknya, nilai persentase yang rendah mengindikasikan masih adanya fungsionalitas sistem yang belum berjalan sesuai dengan harapan dan memerlukan perbaikan lebih lanjut. Dengan pendekatan ini, hasil pengujian dapat disajikan secara sistematis dan objektif sebagai dasar evaluasi kualitas *fungsional* sistem.

2.6 *React Js*

React.js adalah *library* JavaScript yang digunakan untuk membangun antarmuka pengguna (*user interface/UI*) yang interaktif dan efisien. React memungkinkan pengembang untuk membuat tampilan secara deklaratif, sehingga setiap perubahan data pada aplikasi dapat dirender ulang secara otomatis tanpa harus memperbarui seluruh elemen antarmuka. Dengan pendekatan ini, kode menjadi lebih terprediksi dan mudah untuk dilakukan *debugging*, sehingga mempermudah proses pengembangan.

Selain itu, React menerapkan konsep berbasis komponen, di mana antarmuka aplikasi dibangun dari komponen-komponen kecil yang terenkapsulasi dan mampu mengelola *state* masing-masing. Komponen-komponen tersebut dapat digunakan

kembali serta dikombinasikan untuk membentuk antarmuka yang lebih kompleks. Karena logika pada komponen React ditulis menggunakan JavaScript tanpa menggunakan *template* khusus, pengelolaan data dalam aplikasi menjadi lebih fleksibel dan tidak bergantung pada manipulasi langsung terhadap *Document Object Model (DOM)*.

React juga memiliki prinsip “*Learn Once, Write Anywhere*”, yang berarti React dapat digunakan pada berbagai lingkungan pengembangan tanpa harus menulis ulang kode yang telah dibuat. Selain dijalankan pada *browser* untuk membangun aplikasi web, React juga mendukung *server-side rendering* menggunakan Node.js serta pengembangan aplikasi seluler melalui React Native. Dengan tingkat fleksibilitas tersebut, React menjadi salah satu pilihan yang banyak digunakan dalam pengembangan aplikasi modern[2].

2.7 Tailwind

Tailwind CSS adalah sebuah *utility-first CSS framework* yang dikembangkan untuk mempermudah dan mempercepat proses pembuatan antarmuka web modern. *Framework* ini menyediakan kumpulan kelas utilitas berukuran kecil yang dapat langsung digunakan di dalam HTML, seperti *flex*, *pt-4*, *text-center*, atau *bg-blue-500*, sehingga pengembang tidak perlu menulis kode CSS secara manual. Pendekatan ini memberikan fleksibilitas yang tinggi karena desain dapat dibangun dengan mengombinasikan kelas-kelas utilitas sesuai kebutuhan, tanpa bergantung pada komponen siap pakai.

Tailwind CSS juga mendukung desain responsif melalui sistem *breakpoints* yang sederhana, sehingga tampilan antarmuka dapat menyesuaikan berbagai ukuran layar perangkat. Selain itu, *framework* ini menyediakan fitur kustomisasi penuh melalui berkas konfigurasi, mendukung *dark mode*, serta melakukan optimisasi dengan menghapus kelas yang tidak digunakan pada saat proses produksi. Dengan demikian, ukuran berkas CSS yang dihasilkan menjadi lebih kecil dan efisien. Dengan filosofi yang menekankan kebebasan, efisiensi, dan konsistensi, Tailwind CSS menjadi salah satu pilihan populer dalam pengembangan sistem berbasis web[11].

2.8 Penelitian Terdahulu

Dalam rangka pelaksanaan penelitian ini, dilakukan studi literatur terhadap beberapa penelitian sebelumnya yang berkaitan dengan topik yang sama. Studi literatur ini bertujuan untuk mendukung pelaksanaan penelitian serta mencegah terjadinya plagiarisme. Adapun beberapa penelitian terdahulu yang relevan dengan topik penelitian ini disajikan sebagai berikut:

1. Analisa dan Perancangan Aplikasi Evaluasi Capaian Pembelajaran Lulusan

Penelitian ini dilakukan oleh Dony Tanu Wijaya, Istiqomah Sumadikarta, dan Bosar Panjaitan pada tahun 2023 dengan judul *Analisa dan Perancangan Aplikasi Evaluasi Capaian Pembelajaran Lulusan*. Penelitian ini membahas pengembangan sistem penilaian terkomputerisasi untuk Capaian Pembelajaran Lulusan (CPL) pada Program Studi Teknik Informatika Universitas Satya Negara Indonesia. Pada kondisi sebelumnya, belum tersedia sistem yang mampu merepresentasikan nilai CPL mahasiswa secara terkomputerisasi. Oleh karena itu, penelitian ini bertujuan untuk membangun aplikasi berbasis web yang dapat mengevaluasi hasil belajar mahasiswa dengan menggunakan bahasa pemrograman PHP dan *framework* CodeIgniter.

Pengembangan sistem pada penelitian ini menggunakan metode *waterfall*, sedangkan pengujian dilakukan menggunakan *test case* dan *User Acceptance Testing* (UAT). Umpan balik serta kesalahan yang ditemukan selama proses pengujian kemudian dianalisis dan diperbaiki guna meningkatkan kualitas sistem. Berdasarkan hasil pengujian, baik dosen maupun mahasiswa menyatakan bahwa sistem yang dikembangkan telah sesuai dengan tujuan penelitian, yaitu mampu merepresentasikan dan mengevaluasi capaian pembelajaran mahasiswa secara efektif[12].

2. Pengembangan Sistem Informasi Rencana Pembelajaran Semester dan Evaluasi Capaian Pembelajaran Lulusan Berbasis *Progressive Web APP*

Penelitian ini dilakukan oleh Nada S. Adilah, Lillyan Hadjaratie, dan Rampi Yusuf pada tahun 2022 dengan judul *Pengembangan Sistem Informasi Rencana Pembelajaran Semester dan Evaluasi Capaian Pembelajaran Lulusan Berbasis Progressive Web App*. Penelitian ini dilatarbelakangi oleh permasalahan dalam penyusunan Rencana Pembelajaran Semester (RPS) dan evaluasi Capaian Pembelajaran Lulusan (CPL) di Jurusan Teknik Informatika Universitas Negeri Gorontalo yang masih menggunakan aplikasi Microsoft Excel. Penggunaan Excel dinilai kurang optimal dan efisien, terutama ketika terjadi perubahan atau pembaruan yang disebabkan oleh perkembangan ilmu pengetahuan dan teknologi.

Untuk mengatasi permasalahan tersebut, penelitian ini mengembangkan sistem informasi menggunakan metode *prototype*. Sistem yang dikembangkan bertujuan untuk mempermudah proses penyusunan dan pengembangan RPS serta membantu Jurusan Teknik Informatika dalam memonitor ketercapaian dan kelayakan CPL. Selain itu, sistem ini memungkinkan mahasiswa untuk mengetahui rencana kegiatan pembelajaran pada setiap pertemuan mata kuliah, nilai tugas, nilai ujian tengah semester, nilai ujian akhir semester, serta informasi lain yang berkaitan dengan CPL yang dibebankan pada mata kuliah. Hasil penelitian menunjukkan bahwa sistem informasi yang dikembangkan mampu meningkatkan efisiensi dan efektivitas dalam penyusunan RPS serta evaluasi CPL[4].

3. Pengembangan Sistem Informasi Evaluasi Capaian Pembelajaran Lulusan Jurusan Informatika FTI UII

Penelitian ini dilakukan oleh Puspita Dewi Cahyawardani dan Hendrik dengan judul *Pengembangan Sistem Informasi Evaluasi Capaian Pembelajaran Lulusan Jurusan Informatika FTI UII*. Penelitian ini berfokus pada pengembangan sistem penilaian capaian pembelajaran mahasiswa pada Program Studi Informatika Universitas Islam Indonesia. Pada kondisi sebelumnya, Program Studi Informatika telah menerapkan evaluasi CPL, namun belum tersedia sistem yang mampu menghitung dan merekapitulasi nilai yang diperoleh mahasiswa secara otomatis.

Untuk mengatasi permasalahan tersebut, penelitian ini bertujuan membangun aplikasi berbasis web dengan menggunakan bahasa pemrograman PHP, *framework* Laravel, serta *plugin* jExcel. Aplikasi yang dikembangkan dirancang untuk membantu pihak program studi dalam memantau hasil nilai mahasiswa, memudahkan dosen dalam melakukan perhitungan penilaian CPL, serta memungkinkan mahasiswa untuk mengetahui hasil nilai yang diperoleh.

Metode yang digunakan dalam perhitungan nilai adalah metode CPL sebagai akumulasi nilai CPMK (Capaian Pembelajaran Mata Kuliah) pendukung CPL, yang diperoleh melalui berbagai aktivitas akademik seperti penugasan, proyek, kuis di kelas, presentasi lisan, ujian, dan kegiatan akademik lainnya. Proses perekapan nilai tersebut dikenal sebagai *direct assessment*. Model pengembangan sistem yang digunakan adalah model *Agile* dengan metode *Scrum*. Pengujian sistem dilakukan melalui skenario pengujian dan *User Acceptance Testing* (UAT). Hasil pengujian dimanfaatkan untuk mengevaluasi dan menyempurnakan sistem, khususnya dalam mengidentifikasi serta memperbaiki *bug* yang ditemukan. Penelitian ini diharapkan mampu meningkatkan efisiensi dan efektivitas dalam proses pemantauan serta penilaian capaian pembelajaran mahasiswa[13].

4. Perancangan Sistem Penilaian Praktikum Industri kreatif Berbasis *Outcome Based Education*(OBE)

Penelitian ini dilakukan oleh Ezra Driasetyawan dan Ir. Ratnanto Fitriadi, S.T., M.T. dengan judul *Perancangan Sistem Penilaian Praktikum Industri Kreatif Berbasis Outcome Based Education (OBE)*. Penelitian ini dilaksanakan di Laboratorium Teknik Industri Universitas Muhammadiyah Surakarta, yang menjadi tempat pelaksanaan praktikum bagi mahasiswa Program Studi Teknik Industri. Dalam proses praktikum, terdapat beberapa alur sistem yang harus dilalui oleh mahasiswa dengan pendampingan asisten praktikum yang bertugas mengarahkan kegiatan serta memberikan penilaian pada setiap tahapan proses.

Tujuan dari penelitian ini adalah mengembangkan sistem penilaian yang terstruktur melalui perancangan rubrik penilaian yang didasarkan pada Capaian Pembelajaran Lulusan (CPL) dan Capaian Pembelajaran Mata Kuliah (CPMK) dengan

menggunakan pendekatan *Outcome Based Education* (OBE). Permasalahan utama yang sering terjadi dalam proses penilaian adalah adanya perbedaan persepsi antar asisten praktikum dalam menilai kemampuan praktikan. Oleh karena itu, diperlukan rubrik penilaian yang jelas dan terstandar guna menghasilkan penilaian yang objektif dan konsisten.

Hasil dari penelitian ini berupa sistem penilaian praktikum berbasis OBE yang mencakup mekanisme konversi nilai dari periode sebelumnya untuk menghasilkan nilai yang konsisten. Output dari perancangan rubrik sistem penilaian praktikum ini disajikan dalam bentuk *chart output* yang memudahkan proses evaluasi Praktikum Industri Kreatif. Dengan demikian, sistem yang dirancang diharapkan mampu meningkatkan efisiensi dan efektivitas dalam proses penilaian praktikum[14].[_____](#)

III. METODOLOGI PENELITIAN

3.1 Waktu dan Tempat

3.1.1 Waktu

Tabel 3. Waktu Pelaksanaan Penelitian

No	Kegiatan Penelitian	Jun	Jul	Aug	Sept	Okt	Nov	Des	Jan
1	Studi Literatur								
2	Analysis Kebutuhan								
3	Membuat Diagram								
4	Merancang Papan Kanban								
5	Proses Kanban(Coding)								
6	Pengujian								
7	Penulisan Laporan								

3.1.2 Tempat

Penelitian ini dilaksanakan di lingkungan jurusan Teknik Elektro Universitas Lampung, Jl. Prof. Dr. Ir. Sumantri Brojonegoro No., Gedong Meneng, Kec. Rajabasa, Kota Bandar Lampung, Lampung.

3.2 Alat dan Bahan Penelitian

3.2.1 Alat Penelitian

Pada penelitian ini, digunakan beberapa alat yang digunakan untuk mendukung dan menunjang setiap tahapan yang sesuai dengan prosedur penelitian. Adapun alat pada penelitian ini terbagi atas dua jenis, yaitu perangkat keras dan perangkat lunak.

Tabel 4 dan tabel 5 di bawah ini merupakan tabel daftar alat-alat yang digunakan dalam penelitian ini:

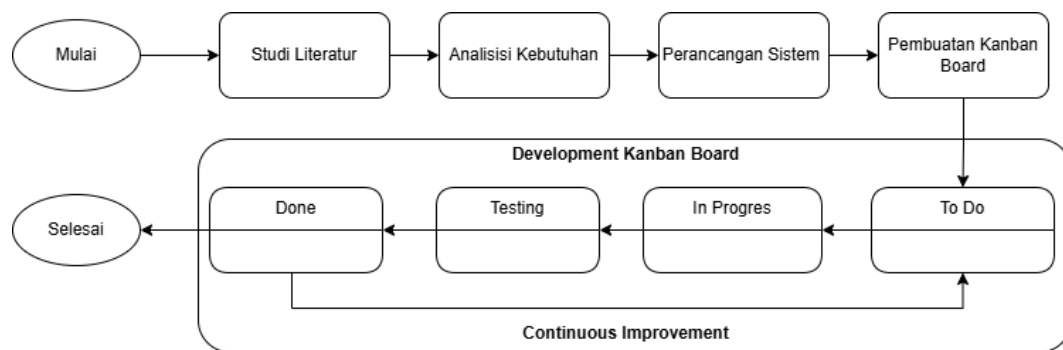
Tabel 4. Perangkat Keras

Perangkat Keras	Fungsi
Laptop Ryzen 3 3250 U	Membuat desain dan implementasi kode program.
<i>Mouse</i>	Menggerakkan kursor.
<i>Keyboard</i>	Mengetik kata atau kalimat.

Tabel 5. Perangkat Lunak

Perangkat Lunak	Fungsi
<i>Visual Studi Code</i>	Kode editor untuk mengolah kode program.
<i>Google Chrome (browser)</i>	Media implementasi website.
<i>React Js</i>	<i>Library</i> pengembangan website.
<i>Tailwind CSS</i>	<i>Framework</i> CSS untuk <i>design</i> antarmuka.
<i>Javascript</i>	Bahasa pemrograman pengembangan web.
<i>Draw Io</i>	<i>Tools</i> untuk membuat diagram.

3.3 Tahapan Penelitian



Gambar 7. Metodologi Penelitian

Gambar 7 menunjukkan tahapan penelitian yang dilaksanakan dalam penelitian ini. Proses penelitian diawali dengan tahap studi literatur, dilanjutkan dengan analisis kebutuhan sistem, perancangan sistem, serta perancangan papan Kanban. Selanjutnya, penelitian memasuki tahapan pengembangan menggunakan metode Kanban yang terdiri atas beberapa kolom alur kerja, yaitu *Backlog* atau *To Do* sebagai daftar tugas yang perlu diselesaikan, *In Progress* untuk tugas yang sedang dikerjakan, *Testing* untuk proses pengujian sistem, dan *Done* untuk tugas yang telah diselesaikan.

Pada papan Kanban, setelah suatu tugas diselesaikan dan melalui tahap pengujian, proses akan kembali ke tahapan *To Do* untuk melanjutkan pengerjaan tugas lainnya. Alur ini memungkinkan proses pengembangan berjalan secara berulang dan terkontrol sesuai dengan prinsip metode Kanban.

3.3.1 Studi Literatur

Tahapan ini bertujuan untuk memperoleh pemahaman yang mendalam mengenai topik yang diteliti. Peneliti melakukan penelusuran terhadap berbagai sumber ilmiah, seperti jurnal, laporan penelitian, dan dokumen kebijakan yang berkaitan dengan Capaian Pembelajaran Lulusan (CPL), serta implementasi sistem informasi dalam konteks pendidikan tinggi. Studi literatur juga mencakup kajian terhadap perangkat kebijakan nasional, seperti Standar Nasional Pendidikan Tinggi (SN-Dikti), yang digunakan sebagai dasar dalam pengembangan sistem. Tahapan ini menjadi fondasi konseptual dan metodologis yang sangat penting untuk

memastikan bahwa sistem yang dikembangkan selaras dengan kebutuhan akademik serta perkembangan teknologi terkini.

3.3.2 Analisis Kebutuhan Sistem

Setelah memperoleh landasan teori yang kuat, penelitian dilanjutkan ke tahap analisis kebutuhan sistem. Pada tahap ini dilakukan identifikasi terhadap kebutuhan pengguna sistem, baik dari sisi fungsional maupun nonfungsional. Kebutuhan fungsional mencakup fitur-fitur utama, seperti input nilai mahasiswa, pemetaan CPMK terhadap CPL, serta penghitungan capaian pembelajaran. Sementara itu, kebutuhan nonfungsional meliputi aspek kemudahan penggunaan antarmuka dan keandalan sistem. Analisis kebutuhan dilakukan melalui observasi terhadap proses bisnis yang berjalan, studi dokumen, serta wawancara informal dengan pihak-pihak yang berkepentingan, seperti dosen dan pengelola sistem akademik. Hasil dari tahap ini digunakan sebagai dasar dalam penyusunan model sistem menggunakan *Unified Modeling Language* (UML).

3.3.3 Perancangan Sistem

Tahap perancangan sistem merupakan proses transformasi dari hasil analisis kebutuhan menjadi desain teknis sistem yang akan diimplementasikan. Perancangan ini bertujuan untuk memastikan bahwa sistem informasi yang dibangun mampu memenuhi kebutuhan pengguna secara efisien, terstruktur, dan mudah dikembangkan di masa mendatang. Pada tahap ini, peneliti menyusun *use case diagram* untuk menggambarkan hubungan antara aktor dan fungsionalitas utama sistem, serta *activity diagram* untuk memvisualisasikan alur proses dari setiap aktivitas yang dilakukan dalam sistem, mulai dari proses input data hingga penghitungan capaian pembelajaran.

3.3.4 Kanban Board

Setelah desain sistem selesai, peneliti memasuki tahap pembuatan *Kanban board*. Kanban digunakan sebagai metode manajemen tugas berbasis visual yang memudahkan pengelolaan pekerjaan selama proses pengembangan sistem. *Kanban board* memetakan setiap tugas ke dalam status pengerjaan tertentu, sehingga

memungkinkan pengaturan prioritas, pelacakan progres, serta pencegahan penumpukan pekerjaan. Penerapan Kanban juga memungkinkan proses pengembangan dilakukan secara iteratif dan adaptif, sejalan dengan prinsip pengembangan perangkat lunak modern.

Pada *Kanban board* terdapat empat kolom utama, yaitu *To Do*, *In Progress*, *Testing*, dan *Done*. Setiap fitur atau modul yang akan dikembangkan pertama kali ditempatkan pada kolom *To Do*. Setelah proses pengerjaan dimulai, tugas dipindahkan ke kolom *In Progress*. Selanjutnya, setelah pengembangan selesai, dilakukan pengujian pada kolom *Testing* untuk memastikan setiap komponen sistem berjalan sesuai dengan spesifikasi dan bebas dari kesalahan.

Apabila hasil pengujian telah memenuhi kriteria yang ditetapkan, tugas dipindahkan ke kolom *Done* sebagai penanda bahwa fitur tersebut telah selesai dikembangkan. Proses ini berlangsung secara berulang dan memungkinkan diterapkannya konsep *continuous improvement*, yaitu peningkatan kualitas sistem secara berkelanjutan berdasarkan hasil evaluasi pada setiap iterasi pengembangan.

3.3.5 Development

Setelah *Kanban board* disusun sebagai alat manajemen tugas, tahap pengembangan (*development*) dilaksanakan secara terstruktur melalui alur kerja yang telah ditetapkan pada papan tersebut. Setiap *backlog* fitur yang telah diidentifikasi dipindahkan secara bertahap dari kolom *To Do* ke *In Progress* sesuai dengan prioritas pengerjaan.

Pada tahap *In Progress*, pengembang mulai membangun komponen *frontend*, menyiapkan struktur *API*, serta mengimplementasikan proses bisnis sesuai dengan kebutuhan sistem. Setelah suatu fitur selesai diimplementasikan, tugas dipindahkan ke kolom *Testing* untuk dilakukan pengujian fungsi, validasi data, serta pemeriksaan kesesuaian antarmuka. Apabila fitur telah lolos pengujian tanpa ditemukan kesalahan, tugas dipindahkan ke kolom *Done* sebagai penanda bahwa fungsi tersebut siap digunakan.

V. PENUTUP

5.1 Kesimpulan

1. Analisis kebutuhan *fungsiional* dan *nonfungsiional* pada *frontend* Sistem Informasi CPL telah berhasil dilakukan. Kebutuhan *fungsiional* mencakup fitur-fitur utama yang mendukung pengelolaan dan penyajian data CPL, sedangkan kebutuhan *nonfungsiional* meliputi aspek *usability*, *performance*, *security*, dan *compatibility*. Hasil analisis tersebut menjadi dasar dalam pengembangan *frontend* sistem.
2. Arsitektur frontend dan struktur komponen sistem berbasis web telah berhasil dirancang dengan menerapkan pemisahan *layer* tampilan, *hooks*, *service*, dan *API*. Perancangan ini dilakukan berdasarkan desain antarmuka yang telah tersedia sehingga menghasilkan struktur *frontend* yang terorganisasi, *modular*, dan mendukung pengelolaan alur data secara efektif.
3. *Frontend* Sistem Informasi CPL telah berhasil dibangun menggunakan pendekatan *Agile Kanban*. Pendekatan ini memungkinkan proses pengembangan dilakukan secara bertahap dan terkontrol, sehingga sistem yang dihasilkan dapat berjalan sesuai dengan kebutuhan yang telah dianalisis.
4. Berdasarkan hasil pengujian fungsional dengan metode *black box testing* serta perhitungan progres pengembangan kebutuhan fungsional, dapat disimpulkan bahwa sistem telah mencapai tingkat keberhasilan pengujian sebesar 78,34% dan

tingkat penyelesaian pengembangan sebesar 66,7%. Hal ini menunjukkan bahwa sebagian besar fungsionalitas utama sudah berjalan sesuai dengan kebutuhan, namun masih terdapat beberapa modul yang belum selesai sehingga perlu dilanjutkan pada tahap pengembangan berikutnya.

5. Meskipun sistem *frontend* telah berhasil dikembangkan dan diuji pada fitur-fitur utama, penelitian ini masih memiliki keterbatasan pada beberapa fitur pendukung yang belum dapat diimplementasikan secara penuh. Keterbatasan tersebut disebabkan oleh batasan ruang lingkup penelitian serta belum terintegrasinya sistem dengan layanan *backend* secara menyeluruh. Rincian fitur yang belum terimplementasi disajikan secara lebih lengkap pada Lampiran B.

5.2 Saran

1. Penelitian selanjutnya disarankan untuk melengkapi dan mengintegrasikan seluruh layanan *backend* atau *API* sehingga *frontend* Sistem Informasi Capaian Pembelajaran Lulusan (CPL) dapat menggunakan data nyata dan beroperasi secara penuh dalam lingkungan operasional program studi.
2. Penelitian berikutnya dapat memperluas cakupan pengujian dengan melibatkan pengguna secara langsung melalui *user acceptance testing* (UAT) untuk mengevaluasi kenyamanan dan kemudahan penggunaan antarmuka sistem.
3. Disarankan agar penelitian selanjutnya dapat melengkapi dan mengembangkan fitur-fitur sistem yang belum terimplementasi maupun fitur tambahan yang dinilai penting dalam mendukung operasional Sistem Informasi Capaian Pembelajaran Lulusan (CPL), termasuk implementasi fitur validasi data CPL secara otomatis guna meminimalkan kesalahan input dan meningkatkan akurasi data yang dikelola oleh sistem. Rekomendasi pengembangan fitur sistem secara lebih rinci disajikan pada Lampiran C sebagai acuan pengembangan sistem di masa mendatang.

DAFTAR PUSTAKA

- [1] Direktorat Pembelajaran dan Kemahasiswaan, *Panduan Penyusunan Kurikulum Pendidikan Tinggi Mendukung Merdeka Belajar-Kampus Merdeka Menuju Indonesia Emas*, Edisi V. Jakarta: Direktorat Jenderal Pendidikan Tinggi, Riset, dan Teknologi, Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi, 2024.
- [2] “React – Library JavaScript untuk membangun antarmuka pengguna.” Accessed: Mar. 24, 2025. [Online]. Available: <https://id.legacy.reactjs.org/>
- [3] I. Sommerville, *Software engineering*, 9th ed. Boston: Pearson, 2011.
- [4] N. S. Adilah, L. Hadjaratie, and R. Yusuf, “Pengembangan Sistem Informasi Rencana Pembelajaran Semester dan Evaluasi Capaian Pembelajaran Lulusan Berbasis Progressive Web App,” *Diffus. J. Syst. Inf. Technol.*, vol. 2, no. 2, Art. no. 2, Jul. 2022, doi: 10.37031/diffusion.v2i2.15571.
- [5] M. Akbar and D. Irianto, “Usulan Sistem Penilaian Capaian Pembelajaran Mata Kuliah (CPMK) dalam Menunjang Penilaian Capaian Pembelajaran Lulusan (CPL) Program Studi,” *Pros. Semin. Nas. Tek. Ind. SENASTI*, vol. 1, pp. 974–984, Oct. 2023.
- [6] F. N. Hasanah and R. S. Untari, *Buku Ajar Rekayasa Perangkat Lunak*. 2020. Accessed: Jul. 31, 2024. [Online]. Available: <https://press.umsida.ac.id/index.php/umsidapress/article/view/978-623-6833-89-6>
- [7] A. Stellman and J. Greene, *Learning agile: Understanding scrum, XP, lean, and kanban*, First edition. Beijing: O’Reilly, 2014.
- [8] A. G. Purwanto, R. Y. Wijaya, T. Timotius, and I. B. Trisno, “Website System Design Using Agile Kanban Based On QR Code,” *JISAJurnal Inform. Dan Sains*, vol. 5, no. 1, pp. 19–27, Jun. 2022, doi: 10.31326/jisa.v5i1.1066.
- [9] Object Management Group, “Unified Modeling Language (UML), Version 2.5.1,” Object Management Group (OMG), formal/2017-12-05, Dec. 2017. [Online]. Available: <https://www.omg.org/spec/UML/2.5.1/PDF>
- [10] I. G. S. Aryandana, A. E. Permanasari, and T. B. Adji, “Comparing method equivalence class partitioning and boundary value analysis with study case add medicine module,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 732, no. 1, p. 012072, Jan. 2020, doi: 10.1088/1757-899X/732/1/012072.
- [11] Tailwind Labs Inc., “Tailwind CSS: Rapidly build modern websites without ever leaving your HTML,” Tailwind CSS. Accessed: Mar. 24, 2025. [Online]. Available: <https://tailwindcss.com>

- [12] D. Tanu Wijaya, I. Sumadikarta, and B. Panjaitan, “Analisa Dan Perancangan Aplikasi Evaluasi Capaian Pembelajaran Lulusan,” *Pros. Semin. Nas. Inov. Teknol.*, vol. 4, pp. 137–147, Nov. 2023, doi: 10.59134/prosidng.v4i.558.
- [13] P. D. Cahyawardani, “Pengembangan Sistem Informasi Evaluasi Capaian Pembelajaran Lulusan Jurusan Informatika FTI UII,” *AUTOMATA*, vol. 1, no. 1, Art. no. 1, Jan. 2020, Accessed: Jul. 31, 2024. [Online]. Available: <https://journal.uui.ac.id/AUTOMATA/article/view/13873>
- [14] E. Driasetyawan and S. T. Ir. Ratnanto Fitriadi, “Perancangan Sistem Penilaian Praktikum Industri Kreatif Berbasis Outcome Based Education (OBE) (Studi Kasus: Praktikum Industri Kreatif Laboratorium Teknik Industri),” s1, Universitas Muhammadiyah Surakarta, 2023. Accessed: Jul. 31, 2024. [Online]. Available: <https://eprints.ums.ac.id/115253/>