

**RANCANG BANGUN SISTEM INFORMASI PELATIHAN  
PENGEMBANGAN KOMPETENSI DOSEN MENGGUNAKAN METODE  
AGILE (Studi Kasus: Pelatihan AA dan Pekerti di lingkungan Universitas  
Lampung)**

**(Skripsi**

**Oleh:**

**MAHEZ PRADANA**

**2015061070**



**JURUSAN TEKNIK ELEKTRO**

**FAKULTAS TEKNIK**

**UNIVERSITAS LAMPUNG**

**2024**

## ABSTRAK

### **RANCANG BANGUN SISTEM INFORMASI PELATIHAN PENGEMBANGAN KOMPETENSI DOSEN MENGGUNAKAN METODE AGILE KANBAN (Studi Kasus: Pelatihan AA dan Pekerti di lingkungan Universitas Lampung)**

Oleh

**Mahez Pradana**

Peningkatan kualitas dosen di perguruan tinggi merupakan aspek krusial dalam pendidikan Indonesia. Kompetensi menurut Undang-Undang Nomor 14 Tahun 2005 tentang Guru dan Dosen adalah pengetahuan, keterampilan, dan perilaku yang harus dimiliki oleh guru atau dosen dalam melaksanakan tugas profesional. Universitas Lampung telah melaksanakan pelatihan PEKERTI dan Applied Approach (AA) sebagai upaya meningkatkan kompetensi dosen. Saat ini, proses pendaftaran manual dan integrasi kelas virtual yang kurang efisien menjadi kendala. Berdasarkan permasalahan tersebut, dirancang dan dikembangkanlah sistem informasi pelatihan dosen berbasis web dengan menerapkan metodologi Agile Kanban dengan 60 task yang dikerjakan selama 5 bulan dan dikembangkan menggunakan PHP, Laravel 10, dan PostgreSQL. Sistem ini memiliki dua peran utama, yaitu admin dengan fitur manajemen pelatihan, manajemen modul, manajemen materi, manajemen tugas, manajemen kuis, manajemen sertifikat, serta dasbor; dan dosen dengan fitur pendaftaran pelatihan, pengerjaan tugas, akses materi, pengerjaan kuis, dan melihat sertifikat. Kemudian dilanjutkan dengan pengujian *blackbox* dilakukan terhadap 103 kasus, yang seluruhnya menunjukkan hasil dengan status "valid" serta dilakukan pengujian *system usability scale* yang menghasilkan respons positif dari lima responden, dengan skor rata-rata sebesar 73 dan mendapat peringkat B-.

**Kata kunci:** Kompetensi Dosen, Sistem Informasi Pelatihan Dosen, *Laravel*, PEKERTI, *Applied Approach*, Universitas Lampung, *Agile Kanban*, *Blackbox Testing*, *System Usability Scale*.

## **ABSTRACT**

### ***BUILDING INFORMATION SYSTEM FOR COMPETENCY DEVELOPMENT TRAINING FOR DOSEN USING KANBAN'S AGILE METHOD (Case Study: AA and Pekerti Training at the University of Lampung)***

**By**

**Mahez Pradana**

*Improving the quality of lecturers in higher education is a crucial aspect of Indonesian education. Competence according to Law Number 14 of 2005 concerning Teachers and Lecturers is the knowledge, skills, and behaviors that must be possessed by teachers or lecturers in carrying out professional duties. Lampung University has carried out PEKERTI and Applied Approach (AA) training as an effort to improve lecturer competence. Currently, the manual registration process and inefficient integration of virtual classes are obstacles. Based on these problems, a web-based lecturer training information system was designed and developed by applying the Agile Kanban methodology with 60 tasks that were carried out for 5 months and developed using PHP, Laravel 10, and PostgreSQL. This system has two main roles, namely admin with features of training management, module management, material management, assignment management, quiz management, certificate management, and dashboard; and lecturers with features of training registration, assignment work, material access, quiz work, and viewing certificates. Then continued with blackbox testing carried out on 103 cases, all of which showed results with an "valid" status and system usability scale testing was carried out which resulted in a positive response from five respondents, with an average score of 73 and received a B- rating.*

**Keywords:** *Lecturer Competency, Lecturer Training Information System, Laravel, PEKERTI, Applied Approach, University of Lampung, Agile Kanban, Blackbox Testing, System Usability Scale.*

**RANCANG BANGUN SISTEM INFORMASI PELATIHAN  
PENGEMBANGAN KOMPETENSI DOSEN MENGGUNAKAN METODE  
AGILE KANBAN (Studi Kasus: Pelatihan AA dan Pekerti di lingkungan  
Universitas Lampung)**

**Oleh**

**MAHEZ PRADANA**

**Skripsi**

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

**Pada**

**Program Studi S1 Teknik Informatika**



**JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG  
2024**

Judul Skripsi

**: RANCANG BANGUN SISTEM INFORMASI  
PELATIHAN PENGEMBANGAN  
KOMPETENSI DOSEN MENGGUNAKAN  
METODE AGILE KANBAN (Studi Kasus:  
Pelatihan AA dan Pekerti di lingkungan  
Universitas Lampung)**

Nama Mahasiswa

**: Mahez Pradana**

Nomor Pokok Mahasiswa

**: 2015061070**

Program Studi

**: Teknik Informatika**

Jurusan

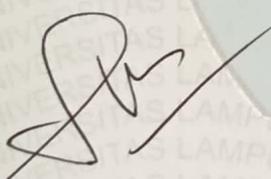
**: Teknik Elektro**

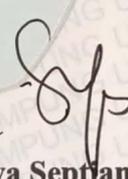
Fakultas

**: Teknik**

**MENYETUJUI**

**1. Komisi Pembimbing**

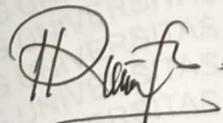
  
**Wahyu Eko Sulistiono, S.T., M.Sc**  
NIP. 197412012001121001

  
**Ir. Trisya Septiana, S. T., M.T.I.P.M**  
NIP. 199009212019032025

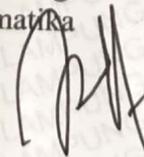
**2. Mengetahui**

Ketua Jurusan Teknik Elektro

Ketua Program Studi Teknik  
Informatika



**Herlinawati, S.T., M.T.**  
NIP. 1917103141999032001

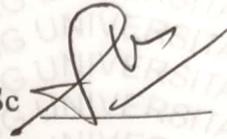


**Yessi Mulyani, S.T., M.T.**  
NIP.197312262000122001

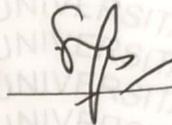
**MENGESAHKAN**

1. Tim Penguji

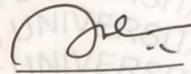
Ketua : Wahyu Eko Sulistiono, S.T., M.Sc



Sekretaris : Ir. Trisya Septiana, S. T., M.T.I.P.M



Penguji : Ir.M. Komarudin, S.T., M.T



2. Dekan Fakultas Teknik



**Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc. )**

**NIP. 197509282001121002**



## SURAT PERNYATAAN

dengan ini saya menyatakan bahwa dalam skripsi ini tidak ada karya orang lain dan sepanjang pengetahuan saya tidak ada karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali secara tertulis diacu dalam naskah ini sebagaimana yang disebutkan dalam daftar pustaka. Selain itu, saya juga ingin mengklarifikasi bahwa tesis ini ditulis oleh saya sendiri. Apabila pernyataan saya tidak benar, maka saya bersedia dikenai sanksi sesuai dengan hukum yang berlaku.

Bandar Lampung, 30 September 2024

Penulis,



Mahez Pradana

2015061070

## RIWAYAT HIDUP



Penulis bernama Mahez Pradana yang dilahirkan di Pajaresuk pada 25 Juli 2001. Yang merupakan putra pertama dari dua bersaudara dari pasangan Maryono dan Erika Agustiani. Pendidikan formal yang pernah diselesaikan oleh penulis adalah SDN 1 Pajaresuk pada 2014, SMPN 1 Pringsewu pada 2017 dan SMAN 1 Pringsewu pada 2020. Penulis merupakan salah satu mahasiswa dari Universitas Lampung dengan program pendidikan S1 Program Studi Teknik Informatika, Jurusan Teknik Elektro, dan Fakultas Teknik. Selama menjadi mahasiswa, penulis aktif dalam beberapa kegiatan sebagai berikut:

1. Peserta program KMMI (Kredensial Mikro Mahasiswa Indonesia) dengan kursus Konsep, Strategi, dan Implementasi Internet of Thing (IOT) pada 2021.
2. Mengikuti program Studi Independen Kampus Merdeka dari Kementerian Riset Pendidikan dan Budaya dengan mengambil Full Stack Web Developer di PT. GITS Indonesia pada tahun 2023.
3. Mengikuti program beasiswa Indosat Ooredoo Hutchison (IDCamp) 2023 di alur belajar *Back-End Developer*.
4. Mengikuti program Magang Bersertifikat Kampus Merdeka dari Kementerian Riset & Pendidikan dan Budaya sebagai Full-stack Developer di PT. Trafoindo Prima Perkasa pada tahun 2023.

## **MOTTO**

“Allah tidak membebani seseorang melainkan  
sesuai dengan kesanggupannya”

**(Q.S. Al-Baqarah 2:286)**

"Mengalirlah seperti sungai, beradaptasi dengan tikungan dan rintangan, tetapi  
jangan pernah kehilangan arah atau membiarkan arus membawa Anda ke tempat  
yang tidak Anda inginkan."

**-Anonim-**

## SANWACANA

Segala puji syukur kehadiran Allah SWT yang Maha Pengasih lagi Maha Penyayang, atas limpahan rahmat, serta nikmat-Nya, penulis dapat menyelesaikan penulisan skripsi ini dengan judul **“RANCANG BANGUN SISTEM INFORMASI PELATIHAN PENGEMBANGAN KOMPETENSI DOSEN MENGGUNAKAN METODE AGILE KANBAN (Studi Kasus: Pelatihan AA dan Pekerti di lingkungan Universitas Lampung)”**. Penelitian ini merupakan salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung.

Dalam pelaksanaan dan pembuatan Skripsi ini penulis mendapat banyak dukungan yang sangat berharga dari berbagai pihak. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada semua pihak yang telah membantu, khususnya kepada:

1. Allah SWT yang senantiasa memberikan kemudahan, kelancaran dan rezeki kepada penulis serta Rasulullah Muhammad SAW yang telah menjadi suri tauladan dalam berperilaku selama kegiatan penelitian skripsi berlangsung.
2. Keluarga penulis yang selalu memberikan dukungan, semangat dan doa dalam hal moral, material, dan spiritual.
3. Bapak Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc., selaku Dekan Fakultas Teknik di Universitas Lampung.
4. Ibu Herlinawati, S.T., M.T., selaku Ketua Jurusan Teknik Elektro di Universitas Lampung.
5. Ibu Yessi Mulyani, S.T. M.T., selaku Ketua Program Studi Teknik Informatika di Universitas Lampung.

6. Bapak Ir. Muhamad Komarudin, S.T. M.T. selaku Pembimbing Akademik dan Dosen Pembahas yang telah meluangkan waktunya untuk memberikan saran dan arahan kepada penulis selama perkuliahan dan skripsi.
7. Bapak Wahyu Eko Sulistiono, S.T., M.Sc., selaku Pembimbing utama yang telah banyak memberikan saran dan bimbingan kepada penulis dalam penyusunan skripsi ini hingga selesai.
8. Ibu Ir. Trisya Septiana, S. T., M.T.I.P.M selaku Pembimbing Pendamping yang telah meluangkan waktunya untuk memberikan bimbingan dan arahan kepada penulis dalam mengerjakan penyusunan skripsi ini hingga selesai.
9. Seluruh Dosen dari Program Studi Teknik Informatika yang telah memberikan ilmu yang bermanfaat kepada penulis.
10. Seluruh staf dari Program Studi Teknik Informatika atas bantuannya dalam menyelesaikan urusan administrasi di Teknik Informatika Universitas Lampung.
11. Seluruh teman-teman Program Studi Teknik Informatika atas dukungan dalam perkuliahan.
12. Ipul, Bob, dan Mansur selaku teman terbaik yang selalu membantu dalam perkuliahan maupun skripsi.

Akhir kata, semoga Allah SWT memberikan balasan atas tindakan baik dan jasa yang telah kalian berikan kepada penulis. Penulis juga berharap dapat memberikan manfaat melalui skripsi ini.

Bandar Lampung, 30 September 2024  
Penulis,

Mahez Pradana  
NPM. 2015061070

## DAFTAR ISI

	<b>Halaman</b>
<b>DAFTAR ISI</b> .....	<b>i</b>
<b>DAFTAR GAMBAR</b> .....	<b>iii</b>
<b>DAFTAR TABEL</b> .....	<b>vii</b>
<b>I. PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Tujuan Penelitian .....	2
1.3 Rumusan Masalah.....	3
1.4 Batasan Masalah .....	3
1.5 Sistematika Penulisan .....	3
<b>II. TINJAUAN PUSTAKA</b> .....	<b>5</b>
2.1 Kompetensi Dosen.....	5
2.2 Pelatihan PEKERTI .....	5
2.3 Pelatihan <i>Applied Approach</i> (AA).....	6
2.4 Sistem Informasi .....	6
2.5 UML ( <i>Unified Modelling Language</i> ) .....	7
2.6 <i>Entity Relationship Diagram</i> .....	8
2.7 <i>Wireframe</i> .....	8
2.8 Situs Web.....	9
2.9 PHP .....	9
2.10 <i>Laravel</i> .....	10
2.11 <i>PostgreSQL</i> .....	11
2.12 <i>Kanban Agile Development</i> .....	12
2.13 <i>User Acceptance Testing</i> (Blackbox).....	14
2.14 <i>System Usability Scale</i> .....	16

2.15	Penelitian Terkait .....	19
<b>III.</b>	<b>METODOLOGI PENELITIAN.....</b>	<b>23</b>
3.1	Waktu dan Tempat.....	23
3.2	Alat.....	24
3.3	Tahapan Penelitian.....	25
3.4	Analisis Kebutuhan Sistem.....	26
3.4.1	Kebutuhan Fungsional .....	27
3.4.2	Kebutuhan Non Fungsional.....	27
3.5	Membangun UML .....	28
3.5.1	Use case diagram.....	29
3.5.2	<i>Activity Diagram</i> .....	32
3.5.3	Class Diagram .....	70
3.6	Entity Relationship Diagram .....	73
3.7	Wireframe .....	75
3.8	Kanban .....	90
3.9	Pengujian Blackbox .....	90
3.10	<i>System Usability Scale</i> .....	91
<b>IV.</b>	<b>HASIL DAN PEMBAHASAN.....</b>	<b>94</b>
4.1	Pengembangan .....	94
4.2	Hasil Pengembangan.....	98
4.2.1	Halaman Admin .....	98
4.2.2	Halaman Dosen .....	104
4.3	Pengujian <i>Blackbox</i> .....	108
4.4	<i>System usability scale</i> .....	130
<b>V.</b>	<b>KESIMPULAN DAN SARAN.....</b>	<b>134</b>
5.1	Kesimpulan .....	134
5.2	Saran .....	135
	<b>DAFTAR PUSTAKA .....</b>	<b>136</b>

## DAFTAR GAMBAR

	<b>Halaman</b>
Gambar 1 Contoh Papan <i>Kanban</i> .....	12
Gambar 2 Sauro Lewis CGS .....	18
Gambar 3 Alur Penelitian.....	25
Gambar 4 Usecase admin.....	29
Gambar 5 Usecase Admin.....	30
Gambar 6 <i>Use Case</i> Diagram Dosen .....	31
Gambar 7 Mendaftar .....	32
Gambar 8 <i>Login</i> .....	33
Gambar 9 Melihat Materi.....	34
Gambar 10 Melihat Tugas.....	35
Gambar 11 Mengerjakan tugas .....	36
Gambar 12 Melihat Ujian.....	37
Gambar 13 Mengunduh sertifikat .....	38
Gambar 14 Membuat Tugas.....	39
Gambar 15 Mengedit Materi .....	40
Gambar 16 Mengedit Tugas.....	41
Gambar 17 Membuat Materi.....	42
Gambar 18 Membuat Ujian.....	43
Gambar 19 Mengedit Ujian.....	45
Gambar 20 Menghapus Materi.....	46
Gambar 21 Menghapus Ujian .....	47
Gambar 22 Menghapus Tugas .....	48
Gambar 23 Melihat Ujian.....	49
Gambar 24 Membuat Akun.....	50
Gambar 25 Mengubah Akun.....	51

Gambar 26 Menghapus Akun .....	52
Gambar 27 Reset <i>Password</i> .....	53
Gambar 28 Upload Sertifikat .....	54
Gambar 29 Menilai Ujian atau Tugas .....	55
Gambar 30 Membuat Pelatihan.....	56
Gambar 31 Membuat Periode .....	57
Gambar 32 Edit Pelatihan .....	58
Gambar 33 Edit Periode .....	59
Gambar 34 Menghapus Pelatihan .....	60
Gambar 35 Menghapus Periode .....	61
Gambar 36 Melihat Pelatihan.....	62
Gambar 37 Mendaftar Pelatihan .....	62
Gambar 38 Terima Pendaftar .....	63
Gambar 39 Menghapus Pengumuman .....	64
Gambar 40 Membuat Pengumuman.....	65
Gambar 41 Edit Pengumuman .....	66
Gambar 42 Mengedit Modul.....	67
Gambar 43 Menghapus Modul.....	68
Gambar 44 Membuat Modul.....	69
Gambar 45 Class Diagram Panel admin .....	70
Gambar 46 Class Diagram Panel Dosen .....	71
Gambar 47 ERD.....	73
Gambar 48 Halaman Register .....	75
Gambar 49 Halaman <i>Login</i> .....	75
Gambar 50 Halaman lupa <i>password</i> .....	76
Gambar 51 Halaman <i>Password</i> Baru.....	76
Gambar 52 Halaman Dasbor Admin.....	77
Gambar 53 Halaman Admin Pelatihan .....	77
Gambar 54 Halaman Admin Form Pelatihan.....	78
Gambar 55 Halaman Admin Detail Pelatihan.....	79
Gambar 56 Halaman Admin Edit Pelatihan.....	80
Gambar 57 Halaman Admin Modul.....	81
Gambar 58 Halaman Admin Pendaftar .....	81

Gambar 59 Halaman Admin Peserta.....	82
Gambar 60 Halaman Detail Materi tugas.....	83
Gambar 61 Halaman Admin Penilaian .....	84
Gambar 62 Halaman Admin Daftar Materi .....	84
Gambar 63 Halaman Admin Penilaian Ujian.....	85
Gambar 64 Halaman Admin Periode .....	85
Gambar 65 Halaman Admin Sertifikat .....	86
Gambar 66 Halaman Dosen Dasbor.....	86
Gambar 67 Halaman Dosen Detail Pelatihan.....	87
Gambar 68 Halaman Dosen Detail Pelatihan.....	88
Gambar 69 Halaman Dosen Modul Pelatihan.....	89
Gambar 70 Halaman Dosen Pengerjaan Ujian.....	89
Gambar 71 Papan Kanban.....	90
Gambar 72 <i>Sauro-Lewis curved grading scale (CGS)</i> .....	93
Gambar 73 folder proyek .....	94
Gambar 74 Kanban .....	95
Gambar 75 Grafik Durasi <i>Task</i> .....	97
Gambar 76 Dasbor Admin .....	98
Gambar 77 Navigasi Pelatihan.....	99
Gambar 78 Halaman Detail Pelatihan.....	99
Gambar 79 Halaman Penentuan Kelulusan.....	100
Gambar 80 Halaman Detail modul .....	100
Gambar 81 Halaman Materi Modul .....	101
Gambar 82 Halaman Tugas Modul .....	101
Gambar 83 Halaman Kuis Modul .....	102
Gambar 84 Halaman Bank Soal .....	102
Gambar 85 Halaman Pengumuman .....	103
Gambar 86 Halaman Periode .....	103
Gambar 87 Halaman User.....	104
Gambar 88 Dasbor Dosen .....	104
Gambar 89 Halaman Kalender.....	105
Gambar 90 Halaman Pelatihan.....	105
Gambar 91 Halaman Sertifikat.....	106

Gambar 92 Halaman Detail Pelatihan.....	106
Gambar 93 Halaman Modul.....	107
Gambar 94 Halaman Tugas.....	107
Gambar 95 Halaman Kuis.....	108
Gambar 96 Halaman Pengerjaan Kuis .....	108
Gambar 97 Rata - rata skor per pertanyaan.....	132

## DAFTAR TABEL

	<b>Halaman</b>
Tabel 1 Pertanyaan SUS.....	17
Tabel 2 Penelitian Terkait .....	19
Tabel 3 Jadwal kegiatan penelitian .....	23
Tabel 4 Alat.....	24
Tabel 5 Daftar Pengujian .....	90
Tabel 6 Tabel Pernyataan SUS .....	92
Tabel 7 Task Depedencies .....	96
Tabel 8 Uji Registrasi Akun.....	109
Tabel 9 Uji Masuk.....	109
Tabel 10 Uji Lupa Kata Sandi.....	110
Tabel 11 Uji Edit Akun .....	110
Tabel 12 Uji <i>Logout</i> .....	111
Tabel 13 Uji Membuat Periode .....	111
Tabel 14 Uji Mengedit Periode .....	112
Tabel 15 Uji Membuat Pelatihan .....	112
Tabel 16 Uji Mengedit Pelatihan .....	113
Tabel 17 Uji Membuat Modul.....	114
Tabel 18 Uji Mengedit Modul.....	115
Tabel 19 Uji Membuat Materi.....	116
Tabel 20 Uji Mengedit Materi.....	117
Tabel 21 Menghapus Materi .....	118
Tabel 22 Uji Membuat Tugas .....	118
Tabel 23 Uji Mengedit Tugas .....	119
Tabel 24 Uji Mengerjakan Tugas.....	120

Tabel 25 Uji Menilai Tugas .....	121
Tabel 26 Uji Membuat Soal .....	121
Tabel 27 Uji Mengedit Soal .....	122
Tabel 28 Uji Menghapus Soal.....	123
Tabel 29 Uji Membuat Kuis.....	123
Tabel 30 Uji Menambah Pertanyaan Kuis .....	124
Tabel 31 Uji Menghapus Pertanyaan Kuis.....	125
Tabel 32 Uji Mengerjakan Kuis .....	126
Tabel 33 Uji Menilai Kuis.....	126
Tabel 34 Uji Membuat Pengumuman .....	127
Tabel 35 Uji Mengedit Pengumuman .....	127
Tabel 36 Uji Menghapus Pengumuman.....	128
Tabel 37 Uji Menghapus Tugas .....	128
Tabel 38 Uji Menghapus Kuis .....	128
Tabel 39 Uji Menghapus Modul .....	129
Tabel 40 Uji Menghapus Pelatihan .....	129
Tabel 41 Uji Menghapus Periode.....	130

## I. PENDAHULUAN

### 1.1 Latar Belakang

Masalah peningkatan kualitas dosen di lingkungan perguruan tinggi merupakan isu yang signifikan dalam dunia pendidikan di Indonesia saat ini. Salah satu faktor penentu kualitas dosen adalah kompetensi dosen dalam pembelajaran. Undang - Undang No. 20 tahun 2003 pasal 39 ayat 2 menegaskan peran penting pendidik sebagai tenaga profesional yang melibatkan perencanaan dan pelaksanaan proses pembelajaran, penilaian hasil pembelajaran, pembimbingan dan pelatihan, serta penelitian dan pengabdian kepada masyarakat, terutama di perguruan tinggi.

Universitas Lampung telah menyelenggarakan pelatihan PEKERTI sejak tahun 1996, program ini sangat bermanfaat bagi pengembangan profesionalisme dosen karena kurikulumnya ditetapkan oleh DIKTI sesuai dengan UU No. 14 tahun 2005 tentang Guru dan Dosen, yang menyebutkan bahwa dosen memiliki beban kerja meliputi perencanaan, pelaksanaan proses, penilaian hasil pembelajaran, bimbingan, pelatihan, penelitian, dan pengabdian kepada masyarakat.[1]

Pelatihan *Applied Approach* (AA) dirancang sebagai lanjutan dari PEKERTI dan bertujuan untuk meningkatkan kompetensi pedagogis dosen. Kurikulum AA, yang juga ditetapkan oleh Direktorat Jenderal Pendidikan Tinggi, sejalan dengan amanat UU No. 14 tahun 2005. Program ini mencakup berbagai materi seperti kebijakan pengembangan profesionalisme dosen, etika moral dalam pembelajaran, manajemen mutu terpadu, konstruktivisme, rekonstruksi mata kuliah, penulisan bahan ajar, pengembangan kurikulum, model pembelajaran aktif dan inovatif, evaluasi PBM, *assessment* alternatif, dan aplikasi penelitian tindakan kelas (PTK).

Untuk mendukung keterampilan pembelajaran, dosen memerlukan latihan berkelanjutan dengan menggunakan berbagai fasilitas seperti komputer, LCD, video pembelajaran, internet, dan buku. Pembuatan bahan ajar dan buku teks juga merupakan keterampilan penting yang perlu dilatihkan. Selain itu, dosen harus terus mengembangkan kemampuan dalam strategi, metode, model pembelajaran, dan penggunaan alat bantu mengajar.

Salah satu tantangan yang dihadapi oleh Universitas Lampung saat ini adalah proses pendaftaran yang masih bersifat manual dan kelas virtual yang tergabung dengan mahasiswa. Pendaftaran dilakukan melalui formulir daring, dan peserta harus menunggu pembuatan akun oleh penyelenggara untuk menggunakan platform kelas virtual untuk mahasiswa yang sudah.

Berdasarkan masalah tersebut, Universitas Lampung berupaya memfasilitasi pelatihan dengan membangun sistem informasi baru yang mempermudah peserta pelatihan dalam mendaftar dan mengakses pelatihan dan mempermudah pengelola pelatihan untuk melacak perkembangan masing-masing peserta. Dengan demikian, diharapkan lulusan Universitas Lampung akan memiliki kompetensi yang kuat dan relevan, serta siap untuk menghadapi tantangan dunia kerja yang terus berkembang.

## **1.2 Tujuan Penelitian**

Tujuan umum dari penelitian ini adalah merancang dan membangun sistem informasi pelatihan pengembangan kompetensi dosen menggunakan metode Agile Kanban dalam studi kasus pelatihan PEKERTI dan AA di Universitas Lampung.

Adapun tujuan khusus dari penelitian ini meliputi:

1. Mengidentifikasi kebutuhan pengguna untuk pengembangan kompetensi dosen dalam pelatihan PEKERTI dan AA,
2. Mengetahui teknologi dan *framework* yang digunakan dalam pengembangan sistem informasi pelatihan dosen,
3. Merancang sistem informasi pelatihan dosen dengan pendekatan Agile Kanban, dengan task yang diatur berdasarkan kebutuhan pengguna,

4. Mengevaluasi hasil pengujian sistem secara fungsional menggunakan metode pengujian *blackbox*, serta mengukur tingkat kepuasan pengguna terhadap sistem melalui pengujian *System Usability Scale (SUS)*.

### 1.3 Rumusan Masalah

Rumusan masalah pada penelitian ini adalah “Bagaimana membangun sistem informasi pengembangan kompetensi dosen menggunakan *framework Laravel* dan metode *Agile Kanban* dalam studi kasus pelatihan pekerti dan AA di Universitas Lampung”.

### 1.4 Batasan Masalah

Batasan masalah dari penelitian ini adalah:

1. Dikembangkan menggunakan bahasa PHP dengan *framework laravel 10* dan *postgreSQL*.
2. Sistem berfokus pada fitur-fitur seperti *login*, *register*, *reset password*, manajemen pelatihan, manajemen modul beserta materi dan tugas, pengerjaan dan penilaian tugas, pemberian sertifikat, dasbor admin, dasbor dosen, dan manajemen pengumuman yang berbasis *website* dan dioptimalkan dalam mode desktop.

### 1.5 Sistematika Penulisan

Isi dari laporan Tugas Akhir ini terdiri dari beberapa bagian bab dengan substansi yang beragam antara lain:

Laporan ini dibagi menjadi beberapa bab untuk memudahkan dalam penguraian antara lain:

BAB I	:	PENDAHULUAN
		Bab ini berisi latar belakang, tujuan, rumusan masalah, batasan masalah, dan sistematika laporan.
BAB II	:	TINJAUAN PUSTAKA
		Pada bab ini berisi tentang teori yang membahas mengenai PHP, <i>laravel</i> , <i>postgreSQL</i> , website, <i>Kanban agile</i> , pengujian <i>blackbox</i> , dan <i>system usability scale</i> .
BAB III	:	METODE PENELITIAN
		Bab ini berisi tentang teori dari <i>agile kanban</i> , <i>use case</i> , desain <i>database</i> , dan <i>wireframe</i> .
BAB IV	:	HASIL DAN PEMBAHASAN
		Bab ini berisi tentang kesimpulan dan saran yang dikumpulkan dari hasil yang telah diperoleh.
BAB V	:	KESIMPULAN DAN SARAN
		Bab ini berisi tentang kesimpulan dan saran yang dikumpulkan dari hasil yang telah diperoleh.
DAFTAR PUSTAKA	:	Bab ini memuat daftar sumber kutipan teori-teori yang dijadikan acuan penulis dalam menulis laporan.

## **II. TINJAUAN PUSTAKA**

### **2.1 Kompetensi Dosen**

Kompetensi menurut Undang-Undang Nomor 14 Tahun 2005 tentang Guru dan Dosen adalah pengetahuan, keterampilan, dan perilaku yang harus dimiliki oleh guru atau dosen dalam melaksanakan tugas profesional. Dalam PermenpanRB Nomor 17 Tahun 2013 dan Permendikbud Nomor 92 Tahun 2014, kompetensi diartikan sebagai kemampuan yang diperlukan dosen untuk melakukan kegiatan pendidikan, penelitian, dan pengabdian masyarakat, mencakup pengetahuan, keterampilan, dan sikap kerja yang relevan. Secara internasional, kompetensi adalah perilaku dan keterampilan yang dapat meningkatkan kinerja atau efisiensi dalam penugasan. Fokus utama adalah efisiensi dan peningkatan kinerja dalam pelaksanaan tugas.[2]

### **2.2 Pelatihan PEKERTI**

Pelatihan Peningkatan Keterampilan Dasar Teknik Instruksional (PEKERTI) bertujuan untuk menyelenggarakan proses pembelajaran yang sesuai dengan tuntutan masyarakat, meningkatkan kemampuan dalam pembelajaran, dan melakukan evaluasi hasil pembelajaran untuk peningkatan kualitas berkelanjutan. PEKERTI penting untuk pengembangan profesionalisme dosen karena kurikulumnya, yang ditetapkan oleh DIKTI, sejalan dengan amanat UU No. 14 tahun 2005 tentang Guru dan Dosen. UU ini menyebutkan bahwa beban kerja dosen mencakup perencanaan, pelaksanaan, penilaian hasil pembelajaran, pembimbingan dan pelatihan, serta penelitian dan pengabdian kepada masyarakat. Program

PEKERTI, dirancang oleh Direktorat Jenderal Pendidikan Tinggi, bertujuan untuk meningkatkan kompetensi pedagogis dosen.[3]

### **2.3 Pelatihan *Applied Approach* (AA)**

Pelatihan *Applied Approach* (AA) menjadi sangat penting dalam pengembangan profesionalisme dosen. AA merupakan program pelatihan yang dirancang oleh Direktorat Jenderal Pendidikan Tinggi untuk meningkatkan kompetensi pedagogi dosen. Program ini bertujuan untuk melatih dosen dalam menggunakan strategi, pendekatan, metode, model pembelajaran yang inovatif, serta menguasai penggunaan alat bantu mengajar dan penulisan bahan ajar.[2]

### **2.4 Sistem Informasi**

Sistem adalah sebuah kesatuan yang terdiri dari orang-orang yang bekerja sama sesuai dengan aturan yang terstruktur untuk mencapai tujuan tertentu. Sistem memiliki beberapa komponen penting, termasuk batasan sistem, lingkungan luar sistem, penghubung sistem, masukan sistem, keluaran sistem, pengolahan sistem, dan sasaran sistem. Sedangkan, informasi adalah data yang telah diolah menjadi lebih bermakna dan berguna bagi penerimanya, serta membantu dalam mengurangi ketidakpastian dalam pengambilan keputusan. Sistem informasi adalah kombinasi terstruktur dari orang-orang, perangkat keras, perangkat lunak, jaringan komunikasi, dan sumber daya data yang mengumpulkan, mengubah, dan menyebarkan informasi dalam suatu organisasi.[4]

Sistem informasi yaitu suatu sistem yang menyediakan informasi untuk manajemen dalam mengambil keputusan dan juga untuk menjalankan operasional perusahaan, di mana sistem tersebut merupakan kombinasi dari orang-orang, teknologi informasi dan prosedur-prosedur yang terorganisasi. Biasanya suatu perusahaan atau badan usaha menyediakan semacam informasi yang berguna bagi manajemen.[5]

## 2.5 UML (*Unified Modelling Language*)

UML (*Unified Modeling Language*) adalah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasi, menyesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis OO (*Object Oriented*). UML sendiri juga memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database* dan komponen-komponen yang diperlukan dalam *software*. [6]

Adapun diagram-diagram dari UML yaitu:

### 1. *Structural* diagram

- *Class* diagram, yaitu diagram yang menjelaskan hubungan antar *class* dalam sebuah sistem.
- *Object* diagram, yaitu diagram yang menjelaskan tentang objek-objek dalam sebuah sistem pada satu titik waktu.
- *Component* diagram, yaitu diagram yang menampilkan komponen dalam sistem dan menjelaskan hubungan antar mereka.
- *Deployment* diagram, yaitu diagram yang menunjukkan tata letak sebuah sistem secara fisik.

### 2. *Behavioral* diagram

- *Activity* diagram, yaitu diagram yang menggambarkan alur kerja dari berbagai aktivitas dosen atau sistem.
- *Use case* diagram, yaitu diagram yang menunjukkan peran dosen dan bagaimana peran dosen menggunakan sistem.
- *Sequence* diagram, yaitu diagram yang menggambarkan interaksi dosen dengan sistem secara sekuensial
- *State chart* diagram, yaitu diagram yang menggambarkan bagaimana sistem dapat bereaksi terhadap suatu kejadian dari dalam atau luar.

## 2.6 *Entity Relationship Diagram*

*Entity Relationship Diagram* (ERD) adalah pendekatan *top-down* dalam perancangan basis data yang dimulai dengan mengidentifikasi data-data terpenting yang disebut entitas, serta hubungan antara entitas-entitas tersebut. Hubungan ini kemudian digambarkan dalam suatu model.[7]

ERD terdiri dari tiga elemen dasar: entitas, atribut, dan relasi.[8]

### 1. Entitas

objek yang menjadi fokus dalam suatu basis data. Entitas bisa berupa manusia, tempat, benda, atau kondisi yang terkait dengan data yang diperlukan. Simbol untuk entitas adalah persegi panjang.

### 2. Atribut

informasi yang ada dalam entitas. Setiap entitas harus memiliki *primary key* sebagai identifikasi unik dan atribut deskriptif lainnya. Atribut biasanya ditempatkan dalam tabel entitas atau bisa juga terpisah. Simbol untuk atribut adalah elips.

### 3. Relasi

menunjukkan hubungan antara dua atau lebih entitas. Simbol untuk relasi adalah belah ketupat. Ada beberapa jenis relasi yang dapat digunakan dalam ERD, yaitu:

- a. *One to One*: Satu anggota entitas berhubungan dengan satu anggota entitas lainnya.
- b. *One to Many*: Satu anggota entitas berhubungan dengan beberapa anggota entitas lainnya.
- c. *Many to Many*: Beberapa anggota entitas berhubungan dengan beberapa anggota entitas lainnya.

## 2.7 *Wireframe*

*Wireframe* adalah sebuah kerangka untuk menata suatu item di halaman web atau aplikasi. *Wireframe* sendiri dapat diartikan dengan sederhana sebagai kerangka gambar. Merancang sebuah *wireframe* berarti merencanakan kerangka garis besar sebuah aplikasi.[9]

## 2.8 Situs Web

Situs Web adalah kumpulan halaman yang saling terhubung yang di dalamnya terdapat beberapa item seperti dokumen dan gambar yang tersimpan di dalam web server. Web adalah sebuah aplikasi yang berada dalam web server yang bisa pengguna akses melalui browser. Web biasanya menampilkan data *user* dan informasi dari server.[10]. Web bisa diakses melalui internet dengan menggunakan alamat unik yang disebut URL (*Uniform Resource Locator*). Web juga bisa berisi berbagai jenis informasi, seperti teks, gambar, video, dan audio, serta fitur interaktif, seperti *form* kontak, komentar, atau *chatting*.[11]

## 2.9 PHP

PHP adalah bahasa pemrograman yang sederhana namun kuat yang digunakan untuk membuat konten HTML[12]. Diciptakan oleh Rasmus Lerdord pada tahun 1994, PHP dapat digunakan dalam dua cara utama:

- Skrip sisi server (*server side script*)

PHP pada awalnya dirancang untuk membuat konten web dinamis, dan masih yang terbaik paling cocok untuk tugas itu. Untuk menghasilkan HTML, memerlukan parser PHP dan sebuah web server untuk mengirim *file* dokumen yang dikodekan. PHP juga telah menjadi populer untuk menghasilkan konten dinamis melalui koneksi *database*, dokumen XML XML, grafik, file PDF, dan masih banyak lagi.

- Skrip baris perintah (*command-line script*)

PHP dapat menjalankan skrip dari baris perintah, seperti halnya *Perl*, *awk*, atau *Unix shell*. PHP dapat menggunakan skrip baris perintah untuk tugas-tugas administrasi sistem, seperti pencadangan dan penguraian log; bahkan beberapa skrip tipe pekerjaan CRON dapat dilakukan dengan cara ini (sebagai tugas PHP non visual).

Halaman PHP umumnya adalah halaman HTML dengan perintah PHP yang tertanam di dalamnya. Hal ini berbeda dengan banyak solusi halaman web dinamis lainnya, yang merupakan skrip yang menghasilkan HTML. Server web memproses

perintah PHP dan mengirimkan keluarannya (dan HTML apa pun dari *file*) ke browser.

PHP mendukung semua sistem basis data yang populer, termasuk *MySQL*, *PostgreSQL*, *Oracle*, *Sybase*, *SQLite*, dan basis data yang sesuai dengan ODBC.

## 2.10 *Laravel*

*Laravel* merupakan *framework* PHP *open source* yang gratis dan berbasis web, yang diciptakan oleh Taylor Otwell untuk pengembangan aplikasi web dengan menggunakan *model-view-controller* (MVC) atau pola arsitektur lainnya. *Laravel* memiliki beberapa fitur menarik, seperti sistem modular yang mudah dikelola, metode akses *database* relasional yang beragam, alat bantu yang memudahkan penyebaran dan pemeliharaan aplikasi, dan banyak fungsi yang memfasilitasi interaksi dengan *database*, seperti mengambil, menyisipkan, memperbarui, dan menyaring data.[13]

*Laravel* adalah sebuah *framework* PHP yang populer dan *powerful* untuk membuat *website* yang dinamis, cepat, dan aman. *Laravel* menerapkan konsep MVC (*Model View Controller*) yang membagi kode program menjadi tiga bagian utama, yaitu:

*Model*, yaitu bagian yang berhubungan langsung dengan *database* dan mengelola data aplikasi. *Model* bisa melakukan operasi CRUD (*Create*, *Read*, *Update*, *Delete*) pada data dan mengembalikan hasilnya ke *controller*.

- *View*, yaitu bagian yang menampilkan data aplikasi ke dosen dalam bentuk halaman web. *View* bisa menggunakan *template engine Blade* yang memudahkan penulisan kode HTML dan CSS dengan sintak yang sederhana dan elegan.
- *Controller*, yaitu bagian yang mengatur alur logika aplikasi dan berkomunikasi dengan *model* dan *view*. *Controller* bisa menerima *input* dari dosen, memanggil *model* untuk mengakses data, dan mengirimkan data ke *view* untuk ditampilkan

## 2.11 PostgreSQL

*PostgreSQL* adalah sebuah sistem manajemen basis data relasional yang bersifat *open source*. Sistem ini menekankan pada *ekstensibilitas*, kreativitas, dan juga kompatibilitas. Sistem ini bersaing dengan vendor-vendor *database* relasional utama seperti *Oracle*, *MySQL*, *SQL server*, dan lainnya. Ini digunakan oleh berbagai sektor termasuk lembaga pemerintah dan sektor publik dan swasta. Ini adalah DBMS lintas platform, dan berjalan di sebagian besar sistem operasi modern termasuk Windows, MAC, dan Linux. Ini sesuai dengan SQL, dan sesuai dengan ACID[14].

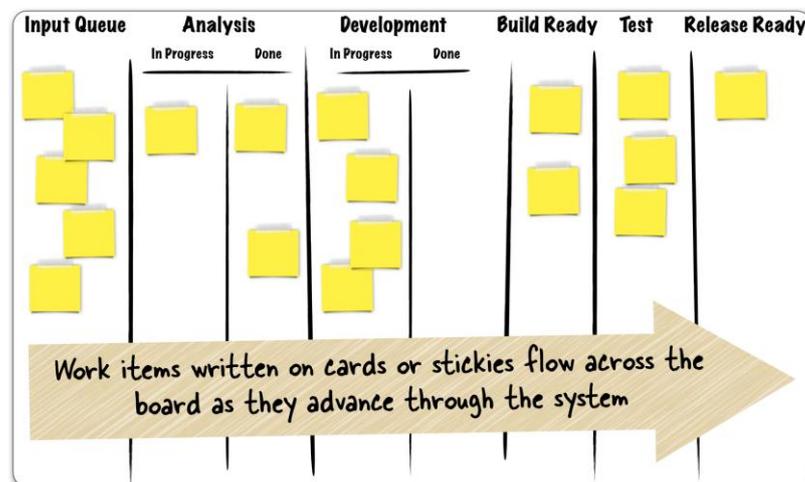
*PostgreSQL* sangat menarik bagi pengembang, administrator, dan arsitek karena memiliki beragam fitur yang kaya, memungkinkan para pengembang untuk menyelesaikan tugas-tugas dengan efisien. Berikut adalah beberapa fitur menarik *PostgreSQL* bagi pengembang:

- Rilis Rutin: Rilis baru hampir setiap tahun, memberikan inovasi terbaru dalam 23 rilis utama sejak Postgres95.
- Dokumentasi Komprehensif: Dokumentasi yang sangat baik dengan lebih dari 2.500 halaman memudahkan pengembang untuk menemukan solusi cepat.
- Ekstensi yang Kaya: Repositori ekstensi yang luas memungkinkan fokus pada logika bisnis dan adaptasi terhadap perubahan persyaratan.
- *Open Source*: Sumber terbuka, dapat dikustomisasi dan diperluas tanpa kesulitan.
- Alat Administratif yang Kuat: Klien dan alat administratif memudahkan tugas rutin seperti deskripsi objek *database*, impor/export data, dan pemulihan *database*.
- Administrasi yang Efisien: Administrasi *database* dapat diotomatisasi, menghemat waktu dan usaha.
- Fleksibilitas Integrasi: Integrasi mudah dengan sistem manajemen basis data lain memberikan fleksibilitas dalam mengimplementasikan desain perangkat lunak.

## 2.12 *Kanban Agile Development*

*Agile* adalah sebuah metode yang membahas semua bidang rekayasa perangkat lunak tradisional, termasuk manajemen proyek, desain dan arsitektur perangkat lunak dan peningkatan proses. Metodologi yang tersaji terdiri dari praktik yang disederhanakan dan dioptimalkan untuk membuatnya semudah mungkin untuk diadopsi[15].

*Kanban* adalah sistem manajemen proses visual yang diperkenalkan oleh Toyota pada 1947, berasal dari bahasa Jepang yang berarti 'papan tanda'. Ini mengadopsi pendekatan *Just In Time* (JIT) untuk mengelola pengetahuan dan pekerjaan dengan fokus pada penghapusan *bottleneck* dan limbah, serta mengurangi waktu tunggu untuk meningkatkan produksi. David J. Anderson, dianggap sebagai bapak *Kanban*, mengenalkannya sebagai metode pengembangan perangkat lunak pada 2004. *Kanban* membantu meningkatkan pemahaman, visibilitas, dan pengendalian alur kerja, serta mengidentifikasi *bottleneck* dalam pengembangan proyek perangkat lunak melalui sistem tarik untuk kolaborasi dan perbaikan terus-menerus.[16]



Gambar 1 Contoh Papan *Kanban*

*Kanban* adalah metode untuk mendefinisikan, mengelola, dan meningkatkan layanan yang memberikan pekerjaan pengetahuan, seperti layanan profesional, upaya kreatif, dan desain produk fisik dan perangkat lunak. *Kanban* adalah metode untuk secara bertahap meningkatkan apa pun yang dikerjakan, baik perangkat lunak pengembangan, IT/ operasi, penetapan staf, perekrutan, pemasaran dan penjualan atau pengadaan.

*Kanban* memiliki praktik-praktik yang memberikan cara untuk menstabilkan dan meningkatkan sistem dalam membangun perangkat lunak[15], yaitu:

Pertama, ikuti prinsip-prinsip dasar:

- Mulailah dengan apa yang Anda lakukan sekarang: Identifikasi dan pahami proses dan alur kerja yang sedang berlangsung saat ini sebelum melakukan perubahan.
- Setujui untuk mengejar perubahan inkremental dan evolusioner: Sepakati untuk melakukan perubahan secara bertahap dan berkelanjutan, dengan terbuka terhadap penyesuaian seiring waktu.
- Awalnya, hormati peran, tanggung jawab, & jabatan pekerjaan yang ada: Pertahankan struktur organisasi dan peran yang ada pada awalnya untuk memfasilitasi transisi yang lebih mulus.

Kemudian, adopsi praktik-praktik inti:

- Visualisasikan: Gambarkan alur kerja dan status tugas secara visual untuk meningkatkan pemahaman dan transparansi.
- Batasi WIP (*Work in Progress*): Tetapkan batasan untuk jumlah tugas yang sedang dikerjakan pada satu waktu untuk mencegah *overburdening* dan meningkatkan fokus.
- Kelola Aliran: Monitor dan atur aliran kerja secara aktif untuk memastikan tugas bergerak lancar dari satu tahap ke tahap berikutnya.
- Jadikan Kebijakan Proses Eksplisit: Definisikan secara jelas kebijakan dan aturan proses untuk memandu tindakan tim dan pengambilan keputusan.
- Terapkan *Loop* Umpan Balik: Sediakan mekanisme untuk umpan balik reguler dari dosen atau tim untuk membantu pengambilan keputusan dan perbaikan proses.
- Tingkatkan secara Kolaboratif, Evolusi secara Eksperimental (menggunakan model/metode ilmiah): Libatkan tim dalam proses perbaikan secara berkelanjutan dengan mengadopsi pendekatan ilmiah untuk eksperimen dan evolusi.

Dalam kerangka kerja *Kanban* yang berbasis *Agile*, tidak ada tahapan formal seperti dalam metodologi *Agile* lainnya seperti *Scrum*. Namun, ada aliran

kerja umum yang biasanya diikuti dalam penggunaan *Kanban*. Berikut adalah aliran kerja yang umum dalam *Kanban Agile*[17]:

1. *Backlog*

Dalam *backlog*, semua tugas akan dibagi menjadi tugas yang lebih terperinci dan menentukan prioritas masing-masing, yang akan dilaksanakan dan diurutkan berdasarkan tingkat prioritas kerja yang berbeda. Semua tugas terkait sistem ini akan dikumpulkan dalam *backlog*.

2. *Todo*

Setelah semua ide dikumpulkan dalam *backlog*, tugas-tugas akan dipindahkan ke bagian "*to do*" (untuk dilakukan).

3. *In progress*

Jika ada tugas yang harus dilakukan, maka tugas tersebut dipindahkan dari tahap "*to do*" ke tahap "*in progress*" (sedang berjalan). Pada tahap ini, pekerjaan pada tugas tersebut sedang dilakukan sesuai dengan prioritasnya. Untuk meningkatkan efisiensi dan mengoptimalkan alur kerja karena banyaknya tugas yang belum selesai, penting untuk menetapkan batasan untuk pekerjaan yang sedang berjalan, sehingga tim fokus pada penyelesaian tugas secara sistematis.

4. *Feedback*

Pada tahap umpan balik ini, dosen memberikan masukan tentang kekurangan atau masalah yang ditemui pada tugas-tugas yang telah diuji oleh dosen selama tahap pengujian.

5. *Done*

Sebuah tugas dianggap selesai dan akan masuk ke tahap "Selesai" jika hasil dari tugas tersebut memenuhi persyaratan dan standar pengembangan, pengujian fungsional telah berhasil, dan tidak ada kesalahan pada tugas tersebut.

### 2.13 *User Acceptance Testing (Blackbox)*

*User Acceptance Testing* (UAT) adalah tes perangkat lunak oleh dosen akhir yang dilakukan sebelum sistem informasi baru diperkenalkan ke sebuah

organisasi. Tujuan utama UAT adalah memastikan sistem baru tersebut berfungsi sesuai dengan yang diharapkan dan memenuhi kebutuhan bisnis yang telah ditetapkan. Berikut adalah definisi UAT yang diambil dari ISTQB *Glossary of Testing Terms*[18]:

Tiga aspek penting dari definisi ini yang memandu persiapan dan pelaksanaan UAT adalah:

1. UAT memerlukan pengujian formal yang dirancang dan dieksekusi secara terstruktur untuk memberikan bukti objektif tentang dapat diterimanya sistem.
2. Pengujian mencakup kebutuhan dosen, persyaratan, dan proses bisnis, serta menyoroti kebutuhan dosen tanpa menyebutkan dokumen spesifikasi tertentu.
3. UAT harus memenuhi kriteria penerimaan yang mendefinisikan apa yang dapat diterima oleh dosen.

Dari semua jenis pengujian yang mungkin didefinisikan, dua jenis pengujian yang sangat berguna untuk UAT adalah pengujian fungsional dan pengujian struktural.

Pengujian struktural (*white-box*) didasarkan pada analisis struktur internal dari komponen atau sistem. *White-box* testing sering dikaitkan dengan pengujian kode, tetapi juga bermanfaat untuk menguji segala hal yang memiliki struktur internal.

Pengujian fungsional (*black-box*) adalah proses pengujian yang didasarkan pada analisis spesifikasi fungsi dari suatu komponen atau sistem, yang sering juga disebut sebagai pengujian kotak hitam. Teknik desain pengujian fungsional merupakan suatu prosedur untuk menghasilkan atau memilih kasus uji berdasarkan analisis spesifikasi fungsi dari suatu komponen atau sistem, tanpa perlu merujuk pada struktur internalnya[19].

Teknik pada pengujian *Blackbox* yaitu:

1. *Equivalence partitioning* adalah teknik merancang kasus uji dengan membagi nilai masukan menjadi partisi kelas ekuivalen valid dan tidak valid. Setiap partisi diuji untuk menemukan kesalahan, mengurangi jumlah kasus uji yang diperlukan sambil tetap efektif dalam menutupi skenario yang beragam.

2. *Boundary value analysis* adalah teknik pengujian yang memfokuskan pada nilai-nilai batas atau nilai terdekat dari domain masukan untuk merancang kasus uji. Tujuannya adalah untuk menemukan kesalahan dengan mengujikan kedua nilai batas yang valid dan tidak valid. Dalam teknik ini, satu kasus uji dipilih dari setiap nilai batas.
3. *Cause effect graphing* adalah teknik pengujian perangkat lunak yang fokus pada perilaku eksternal sistem. Teknik ini menghubungkan kondisi masukan dan keluaran secara logis dengan operator Boolean, di mana nilai masukan mewakili 'Sebab' dan nilai keluaran mewakili 'Akibat'. Hubungan ini digunakan untuk membuat kasus uji.
4. *Decision table based testing* adalah teknik yang efektif untuk menangani banyak masukan dan keluaran. Tabel Keputusan mencakup semua kemungkinan nilai dari kondisi tertentu, membuatnya sangat berguna untuk mengonversi alur bisnis yang rumit menjadi kasus uji.
5. *Error guessing* adalah sebuah teknik di mana para pengujian berpengalaman menggunakan asumsi dan intuisi untuk menemukan kecacatan. Keberhasilan teknik ini bergantung pada keterampilan para pengujian dalam mengidentifikasi di mana dan jenis kecacatan yang mungkin ditemukan.

#### 2.14 System Usability Scale

Berdasarkan ISO (9241-11) *Usability* adalah, " Sejah mana sebuah produk dapat digunakan oleh dosen yang ditentukan untuk mencapai tujuan yang ditentukan dengan efektivitas, efisiensi, dan kepuasan dalam konteks penggunaan tertentu." [20]

Metode evaluasi *usability* dapat dibagi menjadi dua kategori: analitik dan empiris.[21]

- Analitik: Metode ini melibatkan para pakar untuk mengevaluasi *usability*. Contohnya adalah *Heuristic Evaluation*, *Cognitive Walkthrough*, dan penggunaan *Guidelines*.
- Empiris: Evaluasi *usability* dilakukan berdasarkan pengalaman langsung dosen. Contohnya adalah *Usability Testing*, *Thinking Aloud*, *User*

*Performance Test, Remote Usability Testing, Beta Test, Forum Test, Cooperative Evaluation, dan Coaching Method.*

Selain itu, ada juga metode evaluasi *usability* yang melibatkan pendapat dosen seperti *User Satisfaction Questionnaire, Field Observation, Focus Group, dan Interviews.*

Salah satu metode evaluasi *usability* yang populer adalah mengumpulkan data dari responden melalui kuesioner. Metode ini mengandung penilaian subjektif para dosen, sehingga perlu dikombinasikan dengan metode lain, seperti *usability testing.* Hasil umpan balik dari pengguna melalui kuesioner, menjadi dasar untuk menyeimbangkan metode lain.

Kuesioner *System Usability Scale (SUS)* banyak digunakan dalam pengukuran kepuasan dosen karena keunggulannya dalam pengelolaan yang cepat dan murah. Metode ini dianggap efisien untuk mengumpulkan data yang valid secara statistik dan memberikan skor yang jelas dan cukup tepat. Pengukuran SUS sering disebut sebagai "*Quick and Dirty test*" karena kemudahan dan kecepatannya. Kuesioner SUS telah digunakan selama lebih dari 25 tahun sebagai bagian dari teknik pengukuran *usability* yang andal. Metode ini tersedia secara bebas sejak 1986 dan telah digunakan oleh para peneliti dan insinyur *usability.* Meskipun belum melalui proses standardisasi formal, SUS telah menjadi bagian dari perangkat evaluasi *usability* komersial seperti Morae dan dianggap sebagai "standar industri".[22]

Terjemahan kuesioner *System Usability Scale (SUS)* ke dalam Bahasa Indonesia telah dilakukan oleh Sharfina & Santoso (2017) dan telah diukur reliabilitasnya. Hasil pengukurannya menunjukkan bahwa kuesioner SUS dalam terjemahan Bahasa Indonesia cukup andal.[23]

Tabel 1 Pertanyaan SUS

No.	Pernyataan
1.	Saya berpikir akan menggunakan sistem ini lagi.
2.	Saya merasa sistem ini rumit untuk digunakan.
3.	Saya merasa sistem ini mudah untuk digunakan.
4.	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini.
5.	Saya merasa fitur-fitur sistem ini berjalan dengan semestinya
6.	Saya merasa ada banyak hal yang tidak konsisten (tidak serasi) pada sistem ini.

7.	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat.
8.	Saya merasa sistem ini membingungkan.
9.	Saya merasa tidak ada hambatan dalam menggunakan sistem ini.
10.	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.

Berikut rumus perhitungan skor SUS.

$$SUS = 2,5 \times \left[ \sum_{n=1}^5 (U_{2n-1} - 1) + (5 - U_{2n}) \right]$$

Kuesioner *System Usability Scale* (SUS) umumnya menggunakan skala *Likert* 1-5, di mana nilai 1 menunjukkan "sangat tidak setuju" dan nilai 5 menunjukkan "sangat setuju", dengan nilai-nilai antara menunjukkan tingkat persetujuan yang berbeda. Skor kuesioner SUS berkisar dari 0 hingga 100, dengan skor rata-rata pada situs web adalah 68. Skor di atas 68 menandakan kepuasan dosen. Berdasarkan studi Lewis (2018), Gambar 2 adalah representasi dari *Sauro-Lewis curved grading scale* (CGS) yang membantu menentukan apakah situs web, aplikasi, atau sistem informasi memiliki tingkat *usability* yang rendah, sedang, atau tinggi.[24]

SUS Score Range	Grade	Percentile Range
84.1 - 100	A +	96 - 100
80.8 - 84.0	A	90 - 95
78.9 - 80.7	A -	85 - 89
77.2 - 78.8	B +	80 - 84
74.1 - 77.1	B	70 - 79
72.6 - 74.0	B -	65 - 69
71.1 - 72.5	C +	60 - 64
65.0 - 71.0	C	41 - 59
62.7 - 64.9	C -	35 - 40
51.7 - 62.6	D	15 - 34
0.0 - 51.6	F	0 - 14

Gambar 2 Sauro Lewis CGS

## 2.15 Penelitian Terkait

Tabel 2 Penelitian Terkait

Judul Penelitian	Isi
<p>“Pengembangan Aplikasi Sistem Pelatihan PEKERTI-AA bagi Dosen berbasis Website di Universitas Brawijaya Malang” oleh Shafa Al Afghany Al Rahman, Denny Sagita Rusdianto, dan Mahardeka Tri Ananta [25].</p>	<ul style="list-style-type: none"> <li>• Pengembangan sistem informasi pelatihan dosen berbasis Laravel.</li> <li>• Fitur utama: autentikasi, profil pengguna, presensi, manajemen tugas dan pelatihan, penilaian, serta integrasi pembayaran.</li> <li>• Kekurangan: belum tersedia fitur lupa password dan ubah pelatihan, yang dapat meningkatkan fungsionalitas sistem.</li> <li>• Kekurangan dan fitur utama, kecuali integrasi pembayaran, akan menjadi referensi dalam pengembangan sistem informasi pelatihan PEKERTI-AA UNILA.</li> </ul>
<p>“Implementasi Sistem Informasi Akademik Berbasis Web Menggunakan Framework Laravel” yang dilakukan oleh Firma Sahrul B, S.Kom., M.Eng., Muhammad Asri Safi’ie, S.Si., dan Ovide Decroly W A., S.T., M.Eng. [26].</p>	<ul style="list-style-type: none"> <li>• Tujuan: Mengimplementasikan Sistem Informasi Akademik berbasis web menggunakan Laravel untuk meningkatkan dokumentasi dan mengurangi entri data manual di institusi pendidikan.</li> <li>• Framework: Laravel dianggap efektif untuk pengembangan sistem informasi web karena fitur modular, interaksi database yang mudah, dan pemeliharaan yang sederhana.</li> <li>• Kesimpulan: Laravel memenuhi semua kebutuhan pengembangan, menunjukkan potensinya sebagai framework yang luas digunakan.</li> <li>• Kontribusi: Penelitian ini menjadi referensi dalam pemilihan Laravel sebagai framework untuk sistem informasi pelatihan PEKERTI-AA di UNILA.</li> </ul>
<p>“Implementasi PostgreSQL Sebagai Sistem Manajemen Basis Data Pada</p>	<ul style="list-style-type: none"> <li>• Pendaftaran Berbasis Web: Membahas implementasi sistem</li> </ul>

<p>Pendaftaran Mahasiswa Baru Berbasis Web” yang dilakukan oleh Tomoyud S. Waruwu [27].</p>	<p>pendaftaran mahasiswa baru secara online yang lebih efisien dibandingkan metode tradisional.</p> <ul style="list-style-type: none"> <li>• PostgreSQL: Digunakan sebagai sistem manajemen basis data karena efisiensi, efektivitas, dan fitur keamanannya.</li> <li>• Desain Database: Proses desain database meliputi dari ERD hingga implementasi fisik menggunakan PostgreSQL.</li> <li>• Fitur Sistem: Mahasiswa baru dapat mendaftar online, mengisi data, dan mengunggah foto, sedangkan admin memiliki fitur untuk mengelola dan memverifikasi pendaftaran.</li> <li>• Kontribusi: Penelitian ini menjadi referensi dalam penggunaan PostgreSQL sebagai basis data yang aman dan efisien serta desain database berbasis ERD untuk pengembangan sistem pendaftaran pada sistem informasi PEKERTI-AA UNILA.</li> </ul>
<p>“Implementasi Sistem Usability Scale Dalam Evaluasi Perspektif Dosen Terhadap Sistem Informasi Akademik Berbasis Mobile” yang dilakukan oleh Baibul Tujni dan Firamon Syakti [28].</p>	<ul style="list-style-type: none"> <li>• Tujuan: Mengevaluasi usability sistem informasi akademik mobile "UBD on your hand" di Universitas Bina Darma menggunakan System Usability Scale (SUS).</li> <li>• Hasil: Sistem mendapat skor rata-rata SUS 87,2, masuk kategori "sangat baik" dan diterima pengguna tanpa perlu perbaikan.</li> <li>• Metodologi: Evaluasi dilakukan dengan 25 mahasiswa dari berbagai fakultas menggunakan metode deskriptif dengan 10 pertanyaan untuk analisis umpan balik.</li> <li>• Kesimpulan: Sistem dianggap berguna, efisien, dan tidak memerlukan modifikasi lebih lanjut.</li> </ul>

	<ul style="list-style-type: none"> <li>• Kontribusi: Penelitian ini menjadi referensi untuk metode evaluasi System Usability Scale (SUS).</li> </ul>
<p>“Penggunaan Framework Laravel Dalam Rancang Bangun Modul Back-End Artikel Website Bisnisbisnis.Id” yang dilakukan oleh Farizan Luthfi [29].</p>	<ul style="list-style-type: none"> <li>• Website Direktori Bisnis: Pembuatan website direktori bisnis Bisnisbisnis.ID menggunakan Laravel dan MySQL.</li> <li>• Modul Back-End: Pengembangan modul Artikel Back-End dengan pengujian Black-Box untuk memenuhi kebutuhan sistem.</li> <li>• Framework Laravel: Dipilih karena sintaksis bersih, perpustakaan fungsional, serta alat migration dan artisan yang memudahkan pengembangan.</li> <li>• Metode Prototyping: Menggunakan metode Prototyping dengan umpan balik iteratif untuk memastikan produk akhir sesuai kebutuhan pengguna.</li> <li>• Kontribusi: Penelitian ini menjadi referensi untuk pemilihan framework serta metode pengujian fungsional blackbox.</li> </ul>
<p>Rancang Bangun Sistem Informasi Media Pembelajaran Interaktif Bahasa Inggris Berbasis Web” yang dilakukan oleh Topan Setiawan, Anggit Suryoprato, M. Fandra Yudha, dan M. Imam Maulana [30].</p>	<ul style="list-style-type: none"> <li>• Tujuan: Desain dan pengembangan sistem informasi media pembelajaran interaktif berbasis web untuk meningkatkan keterampilan bahasa Inggris menggunakan PHP, Laravel, dan MySQL.</li> <li>• Metodologi: Menggunakan metode desain Kanban dengan alat pemodelan UML.</li> <li>• Fitur: Menyediakan gambar, audio, video, animasi, dan latihan untuk memperkaya pengalaman belajar.</li> <li>• Implementasi: Membantu pelajar, terutama pemula, meningkatkan keterampilan</li> </ul>

	<p>bahasa Inggris melalui modul seperti gramatika, tenses, dan percakapan.</p> <ul style="list-style-type: none"> <li>• Kontribusi: Penelitian ini menjadi referensi untuk penggunaan UML dan metode Agile Kanban dalam pengembangan sistem informasi PEKERTI-AA UNILA.</li> </ul>
<p>“Design of the Information System for Kindergarten Learning Evaluation using Kanban Methodology” yang dilakukan oleh M. Fitriawati dan R.H. Lestari [31].</p>	<ul style="list-style-type: none"> <li>• Tujuan: Merancang sistem informasi untuk evaluasi pembelajaran taman kanak-kanak, mempermudah penilaian perkembangan siswa.</li> <li>• Metodologi: Menggunakan metodologi Kanban untuk pengembangan sistem yang efektif.</li> <li>• Tantangan: Guru kesulitan memproses evaluasi harian sesuai Kurikulum 2013, yang memakan waktu.</li> <li>• Manfaat: Sistem ini menyederhanakan proses evaluasi, mengurangi beban kerja, dan meningkatkan efisiensi dalam melacak kemajuan siswa.</li> <li>• Kontribusi: Penelitian ini menjadi referensi untuk penggunaan metode Agile Kanban dalam pengembangan sistem informasi PEKERTI-AA UNILA.</li> </ul>

### III. METODOLOGI PENELITIAN

#### 3.1 Waktu dan Tempat

Penelitian dan pembuatan Tugas Akhir ini adalah dilakukan pada:

Waktu : Februari 2024 – Juli 2024

Tempat : Lab. Komputer Unila

Jadwal kegiatan penelitian yang dilakukan adalah sebagai berikut:

Tabel 3 Jadwal kegiatan penelitian

No.	Aktivitas	Februari	Maret	April	Mei	Juni	Juli	Agustus
1	Studi Literatur	■						
2	Analisis Kebutuhan		■					
3	Membuat Diagram ( <i>Usecase, Activity, ER</i> )		■					
4	Membuat <i>Wireframe</i>		■					
5	Merancang Papan Kanban			■				
6	Proses Kanban ( <i>Coding</i> )			■	■	■	■	
7	Pengujian <i>Blackbox</i>						■	
8	Pengujian System <i>Usability Scale</i>						■	
9	Penulisan Laporan	■	■	■	■	■	■	■

### 3.2 Alat

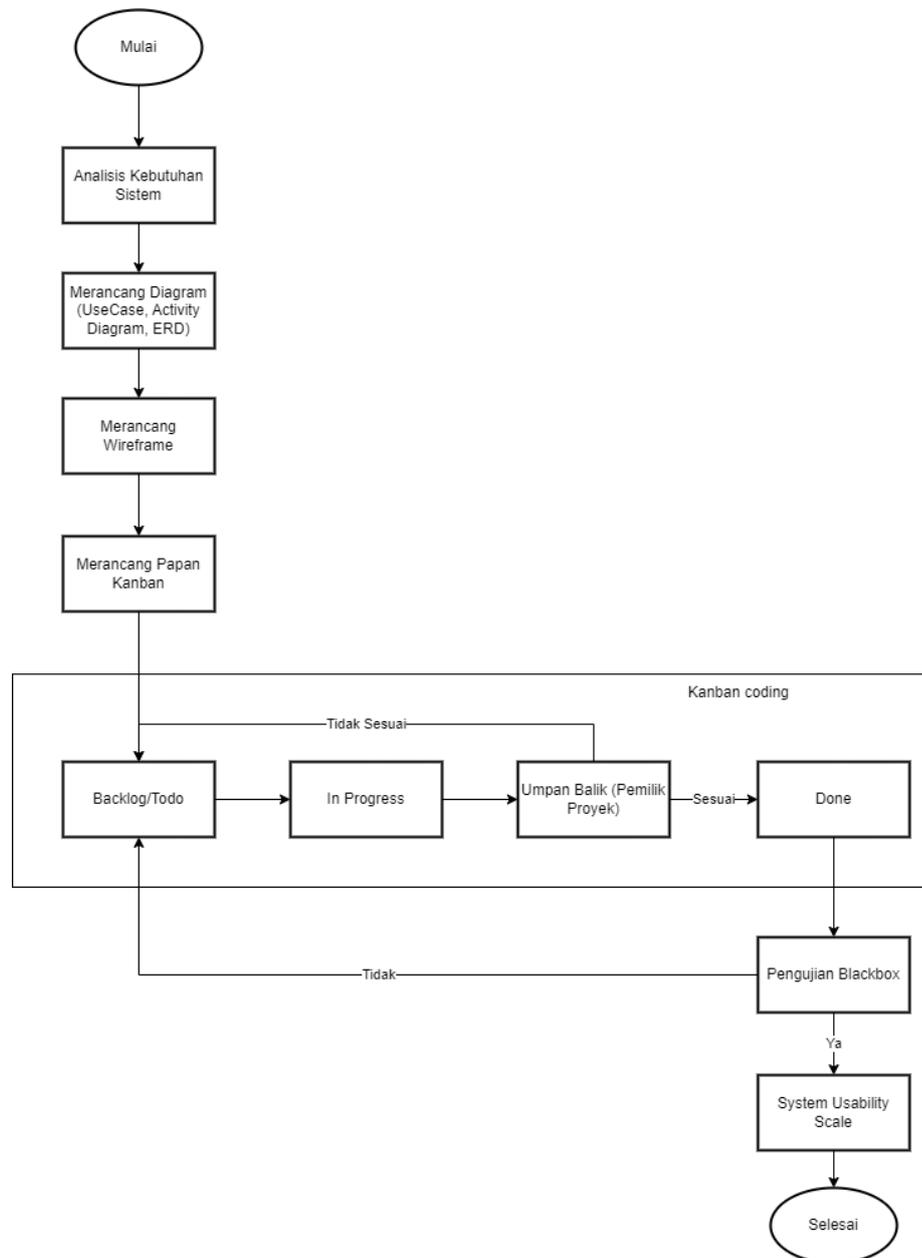
Adapun alat yang dibutuhkan dalam penelitian ini adalah sebagai berikut:

Tabel 4 Alat

No	Perangkat	Spesifikasi	Kegunaan
1	Laptop	Intel I7-1255U, RAM 12GB, VGA Geforce MX550, SSD 512GB	Perangkat untuk merancang sistem
2	<i>Laravel</i>	<i>Laravel version 10</i>	Kerangka kerja program PHP untuk membangun <i>website</i>
3	<i>PostgreSql</i>	<i>PostgreSql version 16</i>	Perangkat lunak sebagai <i>database server</i>
4	<i>PHPStorm</i>	<i>Version 2023.2.3</i>	Perangkat lunak sebagai text editor php
5	<i>Mockup</i>	<i>Balsamiq</i>	<i>Software</i> untuk desain wireframe
6	Desain <i>requirements</i>	chatuml	Website untuk membuat uml berbasis kode
7	PHP	<i>Php 8.1</i>	Bahasa pemrograman yang digunakan laravel
8	<i>Jira</i>		<i>Website</i> untuk manajemen kanban

### 3.3 Tahapan Penelitian

Adapun tahap penelitian yang dilakukan adalah sebagai berikut.



Gambar 3 Alur Penelitian

Berdasarkan gambar 3 alur penelitian adalah sebagai berikut.

1. Analisis Kebutuhan adalah tahap pengumpulan informasi mengenai sistem yang akan dibuat seperti kebutuhan fungsional dan kebutuhan non fungsional yang dilakukan dengan melakukan wawancara terhadap klien yaitu LP3M UNILA.

2. Membangun Diagram yaitu proses membuat diagram *usecase*, *activity*, dan *entity relationships* diagram yang akan memudahkan pengembangan web.
3. Merancang desain halaman menggunakan wireframe.
4. Merancang papan kanban adalah proses mempersiapkan alat berupa melakukan pembuatan akun pada platform Jira serta menginisiasi proyek.
5. *Backlog/ToDo* yaitu mengumpulkan tugas-tugas yang akan dikerjakan berdasarkan *usecase* dan *activity* diagram yang sudah dibuat.
6. *In progress* adalah daftar fitur yang sedang dalam proses pengerjaan. Pada tahap ini, fitur yang ada dalam daftar ini akan dikerjakan, kemudian dipindahkan ke tahap umpan balik selanjutnya.
7. Umpan Balik adalah tahap di mana fitur yang telah selesai dikerjakan akan ditampilkan kepada klien untuk mendapatkan umpan balik.
8. *Done* adalah daftar fitur yang sudah dikerjakan dan diterima oleh klien.
9. Pengujian blackbox adalah tahap di mana setelah semua fitur dinyatakan "Done", website akan diuji untuk memastikan fungsionalitas fitur-fitur tersebut berjalan dengan baik dan sesuai dengan kebutuhan. Jika semua fitur telah diterima, proses akan berlanjut ke pengujian SUS. Jika terdapat revisi atau perbaikan yang diperlukan, task debugging akan dilakukan.
10. *System Usability Scale* adalah tahap di mana setelah semua fitur dalam keadaan "Done", website akan didistribusikan kepada calon dosen yaitu dosen untuk dilakukan kuesioner sesuai kaidah *system usability scale*.

### **3.4 Analisis Kebutuhan Sistem**

Dalam tahap ini, tujuan dari pengembangan sistem ditentukan, kebutuhan dosen dianalisis, permasalahan diidentifikasi, dan data atau informasi yang relevan dengan penelitian dikumpulkan dengan cara melakukan wawancara terhadap klien ini yaitu Kepala Pusat PAI & IP pada LP3M UNILA.

### 3.4.1 Kebutuhan Fungsional

Adapun setelah dilakukannya identifikasi terhadap tujuan pengembangan aplikasi didapatkan kebutuhan fungsional dosen dan admin yaitu:

1. Dosen dapat mendaftarkan diri melalui *email*, *login*, dan reset *password*.
2. Dosen dibagi menjadi dosen internal (UNILA) atau dosen *external*.
3. Dosen dapat mendaftar pelatihan serta mengunggah persyaratan pelatihan.
4. Dosen dapat mengakses *dasbor* pelatihan dan melihat pengumuman.
5. Dosen dapat mengakses modul serta materi, tugas, dan ujian.
6. Dosen dapat mengerjakan ujian dan tugas.
7. Dosen dapat memperbarui data diri
8. Dosen dapat mengunduh sertifikat
9. Admin dapat mengelola pelatihan dan periode.
10. Admin dapat menilai dan mengoreksi ujian atau tugas.
11. Admin dapat meluluskan dan mencetak sertifikat peserta.
12. Admin dapat melakukan validasi data pendaftar.
13. Admin dapat mengelola modul beserta materi, tugas, dan kuis.
14. Admin dapat mengelola pengumuman.

### 3.4.2 Kebutuhan Non Fungsional

Adapun setelah dilakukannya identifikasi terhadap tujuan pengembangan aplikasi didapatkan kebutuhan non fungsional sistem yaitu

1. *Usability*: memiliki navigasi yang jelas dan mudah
2. *Performance*: halaman situs web dapat dimuat kurang dari 5 detik

### 3.5 Membangun UML

Dalam tahap ini terdapat beberapa diagram yang akan menjadi acuan selama pengembangan yaitu:

- ***Use Case***

Tujuan dari Use case dalam penelitian ini adalah untuk menggambarkan bagaimana user berinteraksi dengan sistem. Serta untuk menguraikan langkah-langkah yang terjadi dalam sebuah sistem.

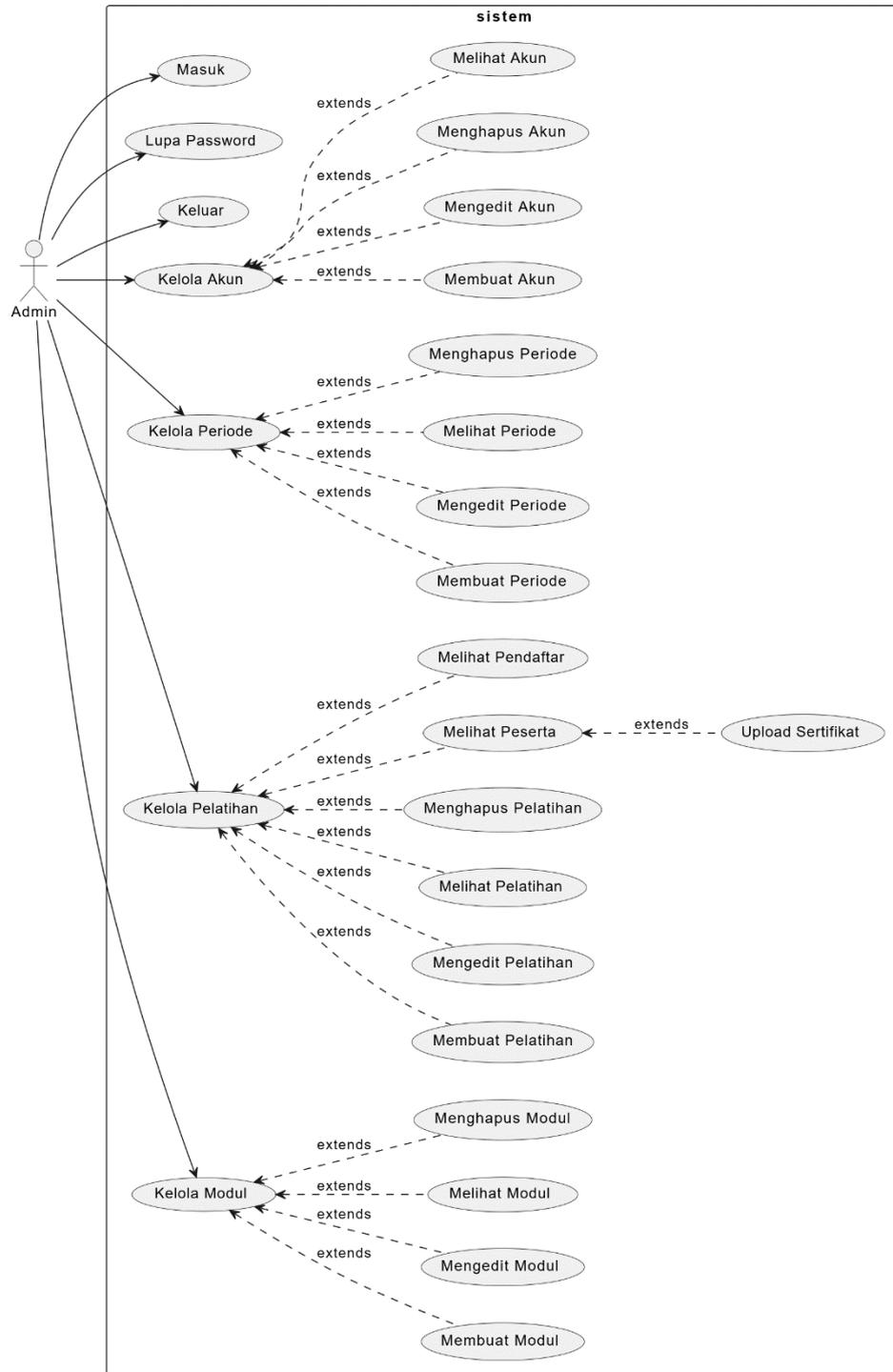
- ***Activity Diagram***

Tujuan dari *activity* diagram dalam penelitian ini adalah untuk menggambarkan bagaimana alur dosen dalam berinteraksi dengan sistem.

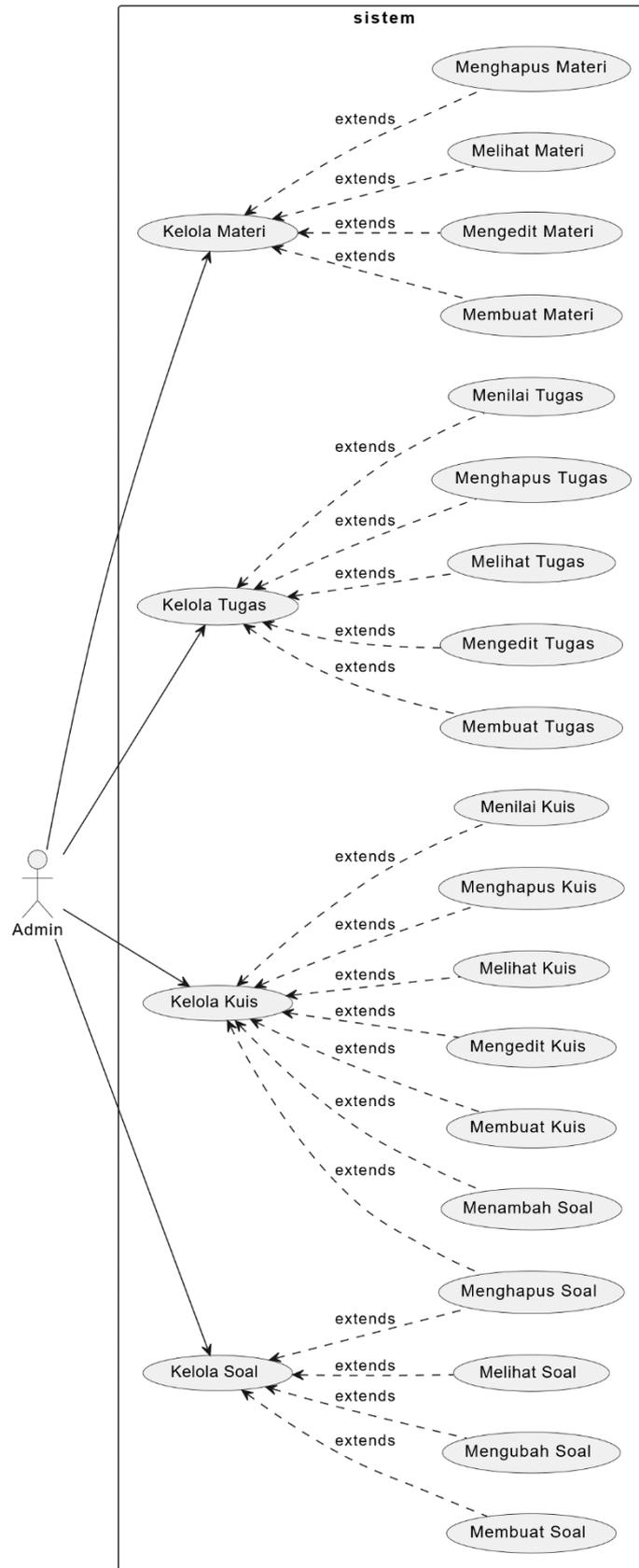
- ***Class Diagram***

Tujuan dari *class* diagram dalam penelitian ini adalah untuk menggambarkan struktur kode dalam sistem.

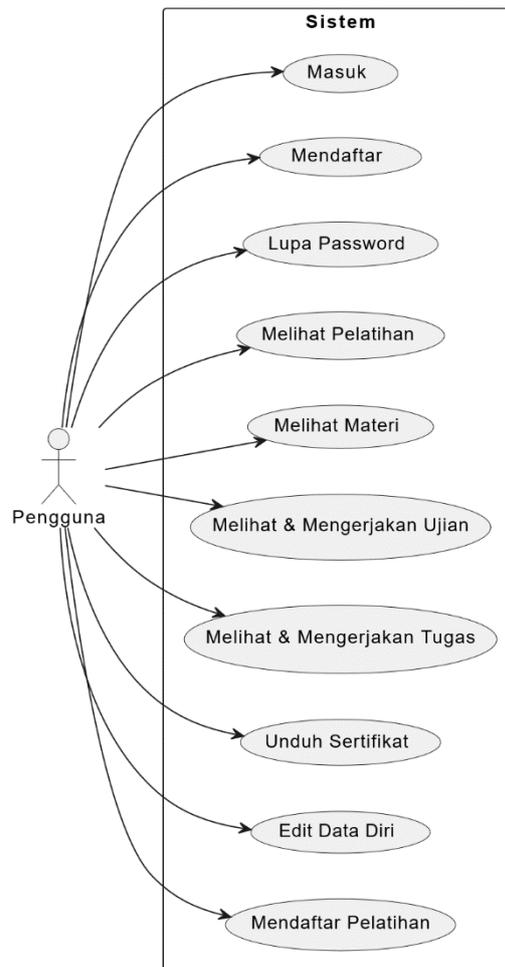
### 3.5.1 Use case diagram



Gambar 4 Usecase admin



Gambar 5 Usecase Admin



Gambar 6 Use Case Diagram Dosen

Berdasarkan gambar 4 hingga 6 sistem pembelajaran *online* ini memiliki dua aktor utama, yaitu dosen dan admin. Dosen adalah dosen yang akan belajar di sistem ini, sedangkan admin adalah pengelola sistem.

Dosen dapat melakukan berbagai hal di sistem ini, mulai dari mendaftar, mengakses materi, mengerjakan ujian/tugas, hingga mencetak sertifikat. Dosen harus mendaftar terlebih dahulu sebelum dapat mengakses sistem ini. Setelah mendaftar, dosen dapat mengisi data verifikasi dan pembayaran untuk mengaktifkan dapat mengakses pelatihan.

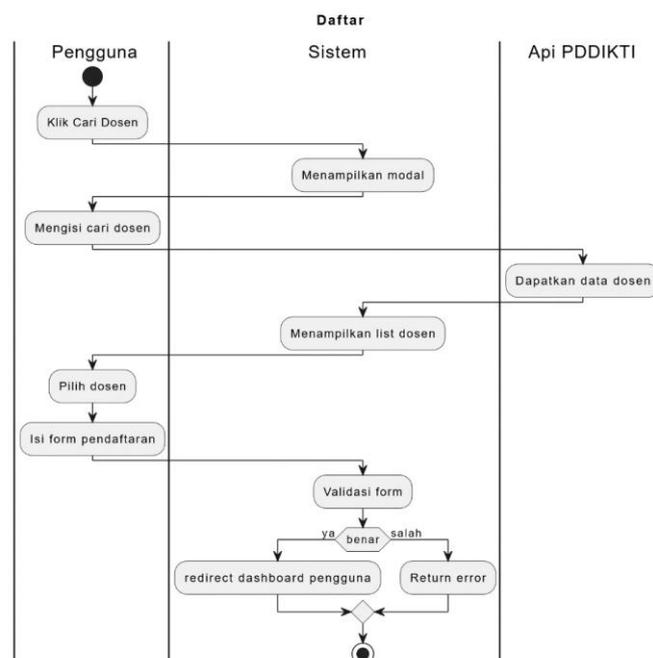
Dosen dapat mengakses *dashbor* untuk melihat informasi pribadi, materi, dan ujian/tugas yang dapat diaksesnya. Dosen dapat mengakses materi untuk belajar,

dan dapat mengerjakan ujian/tugas yang tersedia. *User* juga dapat *login* dan reset *password* jika lupa.

Admin dapat melakukan berbagai hal di sistem ini, mulai dari memvalidasi data *user*, membuat ujian/tugas, mengoreksi ujian/tugas, hingga membuat sertifikat. Admin juga dapat mencetak sertifikat dan mengunduh sertifikat untuk *user* yang telah menyelesaikan ujian/tugas. Selain itu, admin juga dapat membuat pemberitahuan untuk *user*, membuat materi, dan melihat kemajuan belajar *user*.

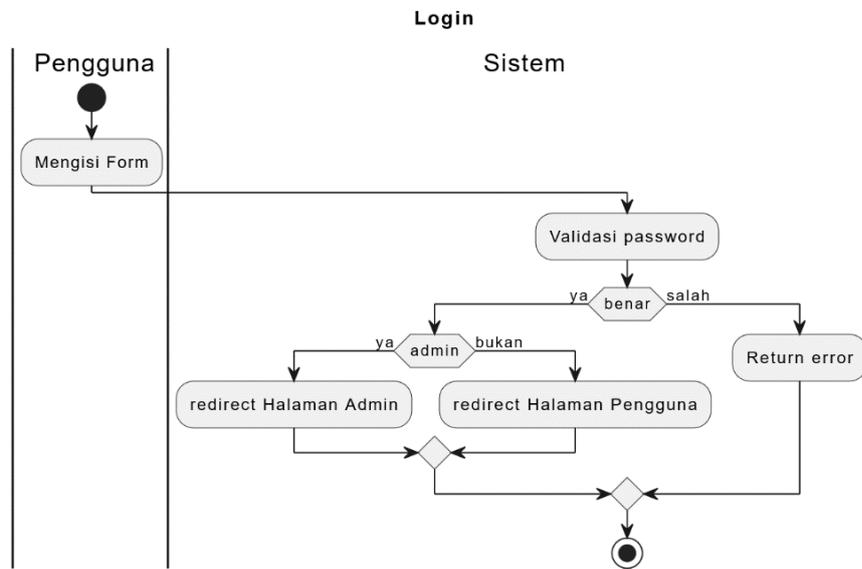
### 3.5.2 Activity Diagram

Pada gambar di bawah ini adalah gambaran tentang rancangan aplikasi yang akan dibuat:



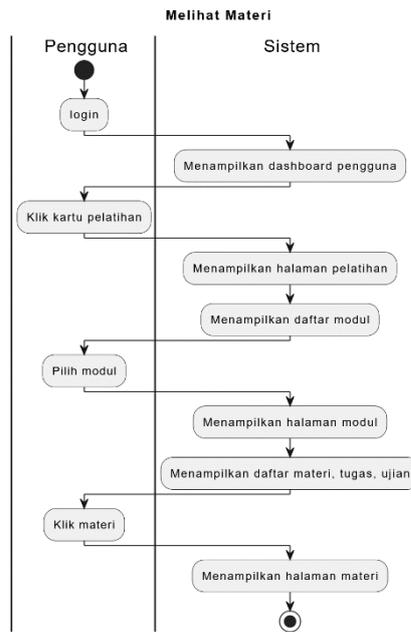
Gambar 7 Mendaftar

Gambar 7 menggambarkan proses pendaftaran dosen dalam sebuah sistem. Dosen memulai dengan mengklik tombol "Cari Dosen", yang memunculkan sebuah modal. Mereka kemudian mencari dan memilih dosen, mengisi formulir pendaftaran, dan sistem melakukan validasi data. Jika data benar, dosen diarahkan ke *dasbor*; jika tidak, pesan *error* ditampilkan. Ini adalah representasi visual yang membantu memahami langkah-langkah dalam proses pendaftaran dengan jelas.



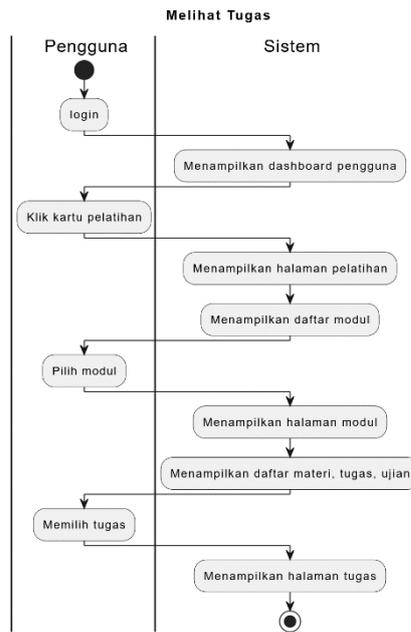
Gambar 8 *Login*

Gambar 8 menggambarkan alur proses *login* dalam sistem. Dosen memulai dengan mengisi formulir *login* dengan detail akun mereka. Sistem kemudian melakukan validasi terhadap *password* yang dimasukkan oleh dosen. Jika *password* yang dimasukkan benar, sistem memeriksa apakah dosen adalah admin atau bukan. Jika dosen adalah admin, mereka diarahkan ke halaman admin; jika tidak, mereka diarahkan ke halaman dosen. Namun, jika *password* yang dimasukkan salah, sistem akan mengembalikan pesan *error* kepada dosen. Ini adalah langkah-langkah yang digambarkan secara visual untuk memahami alur proses *login* dalam sistem dengan jelas.



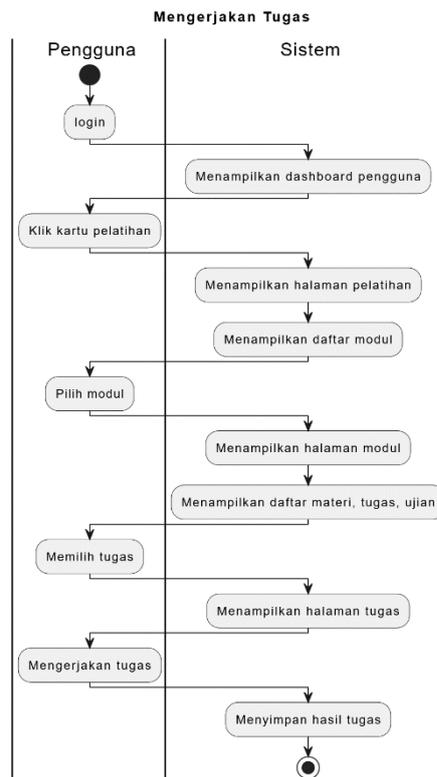
Gambar 9 Melihat Materi

Gambar 9 menjelaskan proses ketika seorang dosen melihat materi dalam sistem. Proses dimulai dengan dosen melakukan *login*, dan sistem menampilkan dasbor dosen. Dosen kemudian memilih kartu pelatihan dari dasbor, dan sistem menampilkan halaman pelatihan dengan daftar modul yang tersedia. Setelah itu, dosen memilih modul yang ingin dilihat, dan sistem menampilkan halaman modul dengan daftar materi, tugas, dan ujian yang terkait. Terakhir, dosen memilih materi yang ingin dilihat, dan sistem menampilkan halaman materi dengan informasi yang relevan.



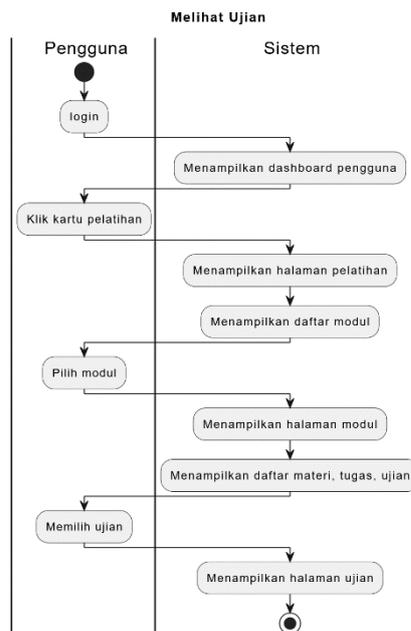
Gambar 10 Melihat Tugas

Gambar 10 menjelaskan proses ketika seorang dosen melihat tugas dalam sistem. Proses dimulai dengan dosen melakukan login, kemudian sistem menampilkan dasbor dosen. Selanjutnya, dosen memilih kartu pelatihan dari dasbor, dan sistem menampilkan halaman pelatihan dengan daftar modul yang tersedia. Dosen memilih modul yang diinginkan, dan sistem menampilkan halaman modul dengan daftar materi, tugas, dan ujian. Terakhir, dosen memilih tugas yang ingin dilihat, dan sistem menampilkan halaman tugas dengan informasi terkait tugas tersebut.



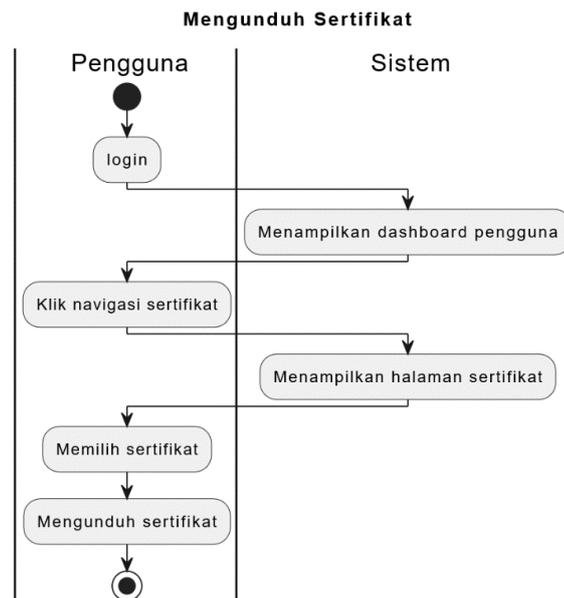
Gambar 11 Mengerjakan tugas

Gambar 11 menggambarkan proses ketika seorang dosen mengerjakan tugas dalam sistem. Proses dimulai dengan dosen melakukan login, kemudian sistem menampilkan dasbor dosen. Selanjutnya, dosen memilih kartu pelatihan dari dasbor, dan sistem menampilkan halaman pelatihan dengan daftar modul yang tersedia. Dosen memilih modul yang diinginkan, dan sistem menampilkan halaman modul dengan daftar materi, tugas, dan ujian. Setelah itu, dosen memilih tugas yang ingin dikerjakan, dan sistem menampilkan halaman tugas. Dosen mengerjakan tugas dan kemudian menyimpan hasilnya.



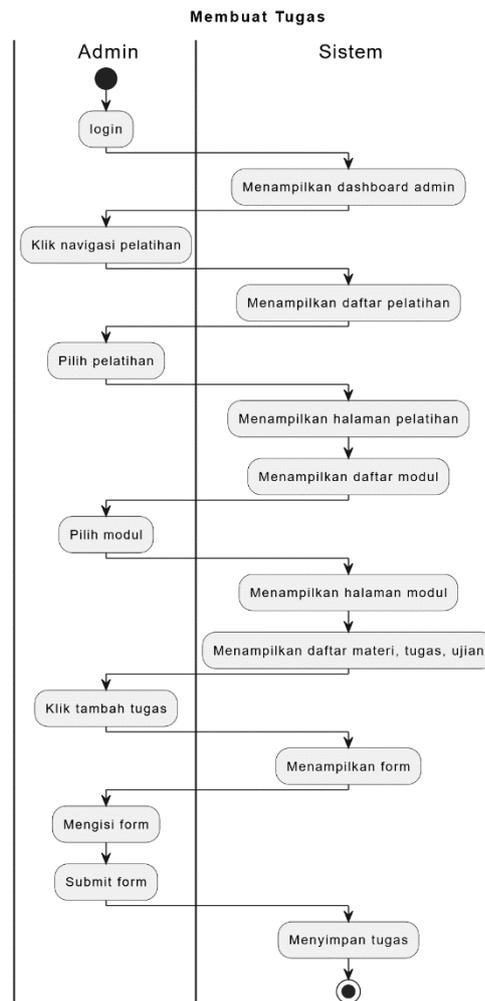
Gambar 12 Melihat Ujian

Gambar 12 menggambarkan proses ketika seorang dosen melihat informasi ujian dalam sistem. Pertama, dosen melakukan login ke sistem. Setelah berhasil login, sistem menampilkan dasbor dosen yang berisi berbagai informasi terkait dengan pelatihan. Kemudian, dosen memilih kartu pelatihan dari dasbor. Setelah memilih kartu pelatihan, sistem menampilkan halaman pelatihan yang berisi daftar modul. Dosen memilih modul yang ingin dilihat, dan sistem menampilkan halaman modul dengan daftar materi, tugas, dan ujian yang tersedia. Terakhir, dosen memilih ujian yang ingin dilihat, dan sistem menampilkan halaman ujian dengan detail mengenai ujian tersebut. Proses ini menunjukkan langkah-langkah yang diambil oleh dosen dan sistem untuk melihat informasi ujian dalam sistem pelatihan.



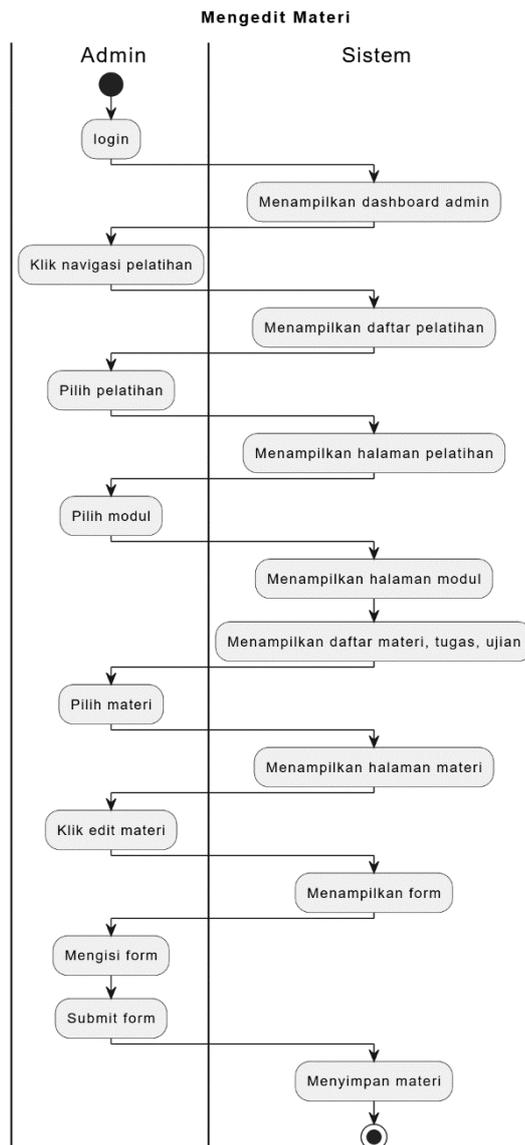
Gambar 13 Mengunduh sertifikat

Gambar 13 mengilustrasikan proses ketika seorang dosen mengunduh sertifikat dalam sistem. Proses dimulai dengan dosen melakukan login, kemudian sistem menampilkan dasbor dosen. Selanjutnya, dosen mengklik navigasi menu untuk sertifikat, dan sistem menampilkan halaman yang berisi daftar sertifikat yang tersedia. Dosen kemudian memilih sertifikat yang ingin diunduh, dan sistem memberikan opsi untuk mengunduh sertifikat tersebut. Setelah dosen memilih untuk mengunduh, sistem memberikan sertifikat dalam bentuk file yang dapat diunduh oleh dosen.



Gambar 14 Membuat Tugas

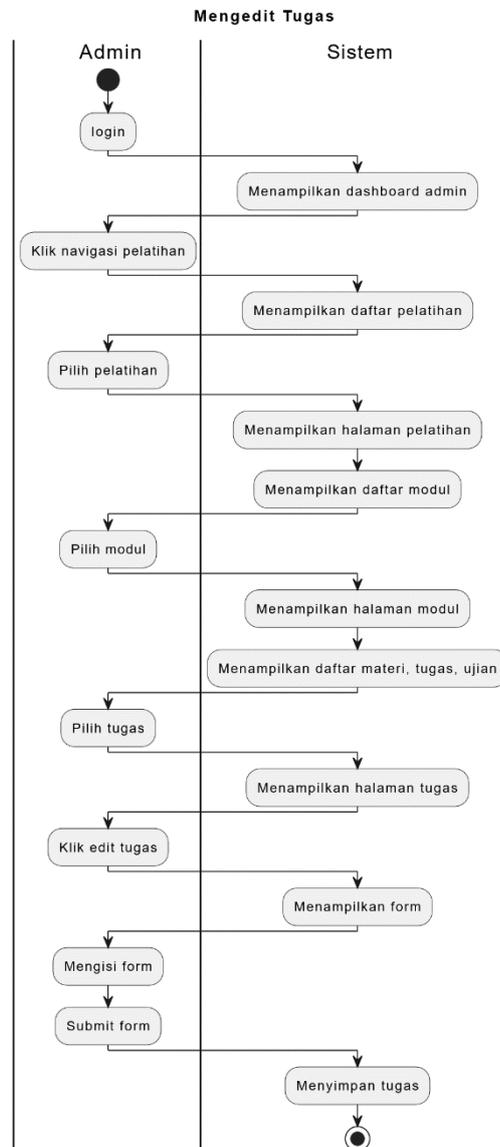
Gambar 14 mengilustrasikan proses pembuatan tugas oleh seorang admin dalam sistem. Proses dimulai dengan admin melakukan login, dan sistem menampilkan dasbor admin. Admin kemudian memilih navigasi pelatihan, dan sistem menampilkan daftar pelatihan. Setelah memilih pelatihan yang sesuai, sistem menampilkan halaman pelatihan beserta daftar modul yang terkait. Admin memilih modul yang ingin ditambahkan tugasnya, dan sistem menampilkan halaman modul bersama dengan daftar materi, tugas, dan ujian yang ada. Selanjutnya, admin mengklik opsi tambah tugas, dan sistem menampilkan formulir pembuatan tugas. Admin mengisi formulir tersebut dengan informasi yang relevan, kemudian mensubmit formulir. Akhirnya, sistem menyimpan tugas yang baru dibuat oleh admin.



Gambar 15 Mengedit Materi

Gambar 15 menjelaskan proses pengeditan materi oleh seorang admin dalam sistem. Proses dimulai dengan admin melakukan login, dan sistem menampilkan dasbor admin. Admin kemudian memilih navigasi pelatihan, dan sistem menampilkan daftar pelatihan. Setelah memilih pelatihan yang sesuai, sistem menampilkan halaman pelatihan bersama dengan daftar modul yang terkait. Admin memilih modul yang berisi materi yang ingin diedit, dan sistem menampilkan halaman modul dengan daftar materi, tugas, dan ujian yang ada. Selanjutnya, admin memilih materi yang ingin diedit, dan sistem menyediakan opsi untuk mengedit materi tersebut. Admin mengklik opsi edit materi, dan sistem menampilkan

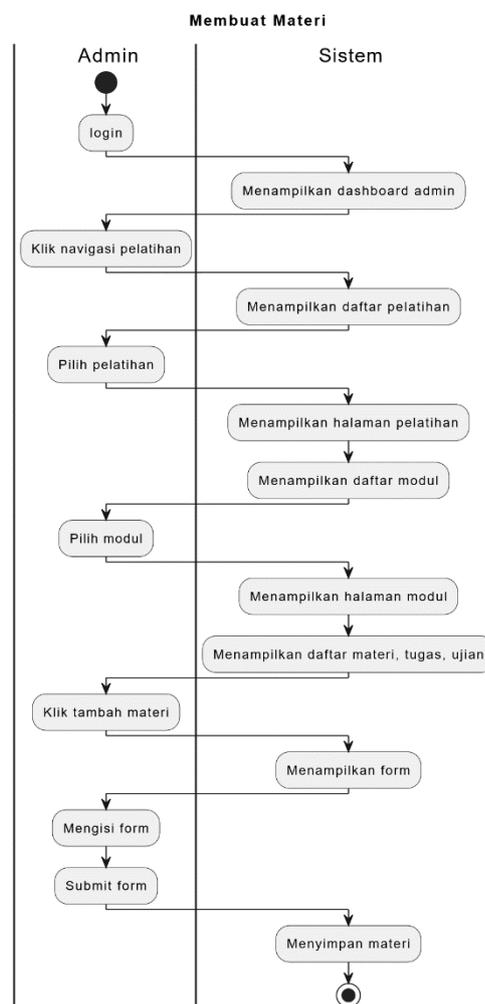
formulir pengeditan materi. Setelah mengisi formulir dengan perubahan yang diinginkan, admin menyubmit formulir tersebut. Akhirnya, sistem menyimpan materi yang telah diedit oleh admin.



Gambar 16 Mengedit Tugas

Gambar 16 menjelaskan proses pengeditan tugas oleh seorang admin dalam sistem. Proses dimulai dengan admin melakukan *login*, dan sistem menampilkan dasbor admin. Admin kemudian memilih navigasi pelatihan, dan sistem menampilkan daftar pelatihan. Setelah memilih pelatihan yang sesuai, sistem menampilkan halaman pelatihan bersama dengan daftar modul yang terkait. Admin

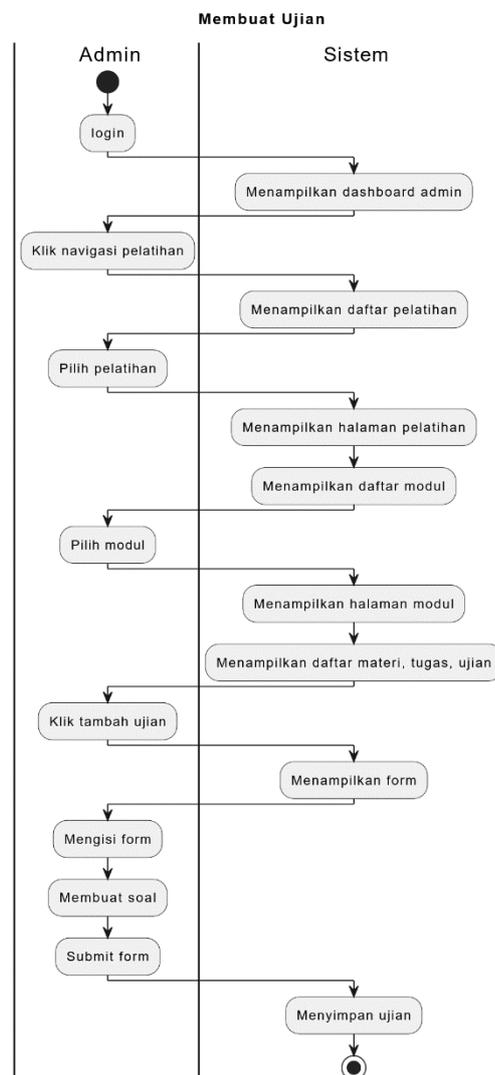
memilih modul yang berisi tugas yang ingin diedit, dan sistem menampilkan halaman modul dengan daftar materi, tugas, dan ujian. Selanjutnya, admin memilih tugas yang ingin diedit, dan sistem menampilkan halaman tugas beserta opsi untuk mengedit tugas. Admin mengklik opsi edit tugas, dan sistem menampilkan formulir pengeditan tugas. Setelah mengisi formulir dengan perubahan yang diinginkan, admin submit formulir tersebut. Akhirnya, sistem menyimpan tugas yang telah diedit oleh admin.



Gambar 17 Membuat Materi

Gambar 17 menggambarkan proses pembuatan materi oleh seorang admin dalam sistem. Proses dimulai dengan admin melakukan *login*, dan sistem menampilkan dasbor admin. Admin kemudian memilih navigasi pelatihan, dan sistem menampilkan daftar pelatihan. Setelah memilih pelatihan yang sesuai, sistem

menampilkan halaman pelatihan bersama dengan daftar modul yang terkait. Admin memilih modul yang ingin ditambahkan materinya, dan sistem menampilkan halaman modul dengan daftar materi, tugas, dan ujian. Selanjutnya, admin mengeklik opsi tambah materi, dan sistem menampilkan formulir pembuatan materi. Admin mengisi formulir tersebut dengan informasi yang relevan, kemudian mensubmit formulir. Akhirnya, sistem menyimpan materi yang baru dibuat oleh admin.

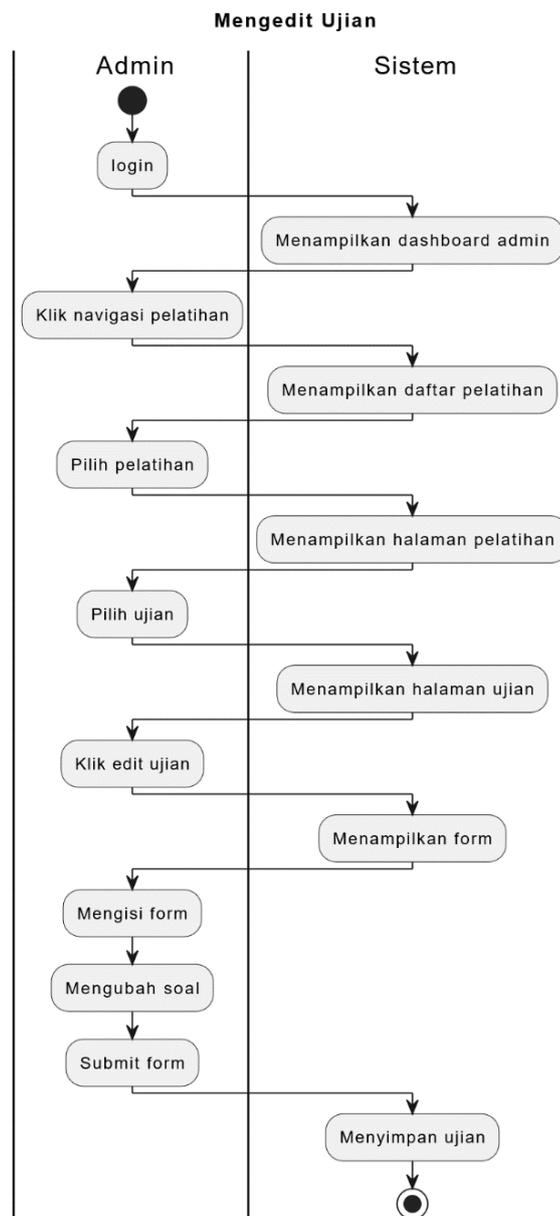


Gambar 18 Membuat Ujian

Gambar 18 menggambarkan proses pembuatan ujian oleh seorang admin dalam sistem. Pertama, admin melakukan *login* ke sistem dan kemudian sistem menampilkan dasbor admin. Selanjutnya, admin mengeklik navigasi pelatihan

untuk melihat daftar pelatihan yang tersedia. Setelah itu, admin memilih pelatihan yang ingin ditambahkan ujian olehnya. Sistem menampilkan halaman pelatihan dengan daftar modul yang tersedia. Admin memilih modul yang relevan untuk ditambahkan ujian. Kemudian, sistem menampilkan halaman modul dengan daftar materi, tugas, dan ujian yang sudah ada.

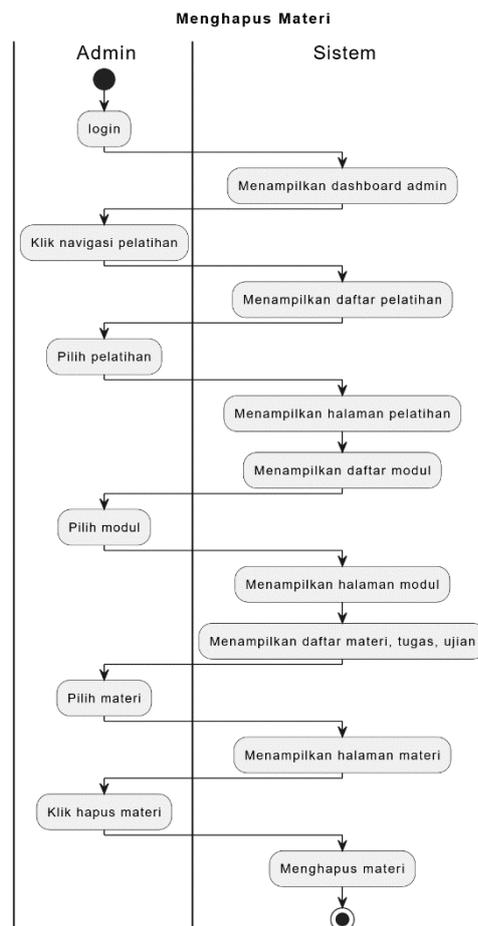
Selanjutnya, admin memilih opsi untuk menambahkan ujian, dan sistem menampilkan formulir untuk membuat ujian. Admin mengisi formulir dengan informasi yang diperlukan dan membuat soal-soal untuk ujian tersebut. Setelah selesai, admin mengirimkan formulir dengan mengklik tombol submit. Sistem kemudian menyimpan ujian yang baru dibuat oleh admin. Proses ini menggambarkan langkah-langkah yang diambil oleh admin dalam membuat ujian di dalam sistem pelatihan.



Gambar 19 Mengedit Ujian

Gambar 19 menggambarkan proses ketika seorang admin mengedit ujian dalam sistem. Proses dimulai dengan admin melakukan *login*, dan sistem menampilkan dasbor admin setelah *login* berhasil. Selanjutnya, admin memilih navigasi untuk melihat pelatihan, dan sistem menampilkan daftar pelatihan yang tersedia. Admin kemudian memilih pelatihan tertentu, dan sistem menampilkan halaman pelatihan terkait. Setelah itu, admin memilih ujian yang ingin diedit dari halaman tersebut, dan sistem menampilkan halaman ujian yang berisi detail mengenai ujian tersebut. Admin mengklik opsi untuk mengedit ujian, dan sistem

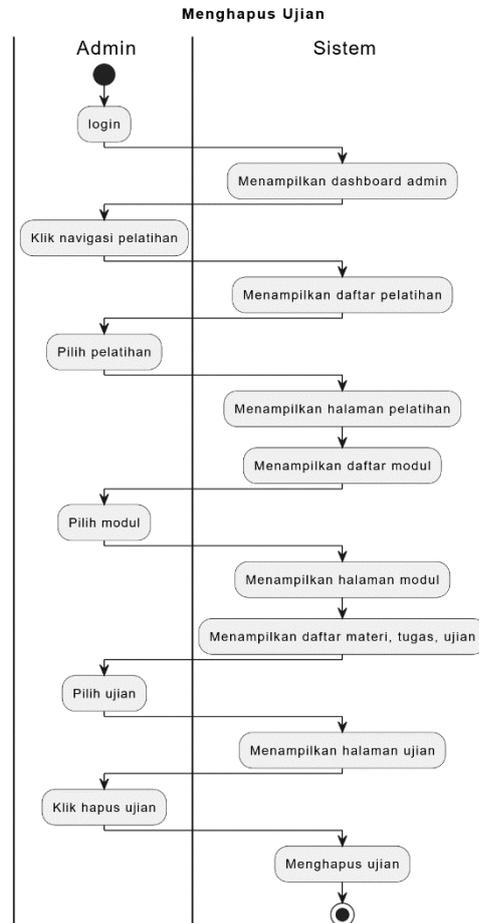
menampilkan formulir yang memungkinkan admin untuk mengubah detail dan soal ujian. Admin mengisi formulir tersebut dengan informasi yang diperlukan, lalu mengirimkan formulir tersebut. Setelah itu, sistem menyimpan perubahan yang telah dilakukan pada ujian tersebut.



Gambar 20 Menghapus Materi

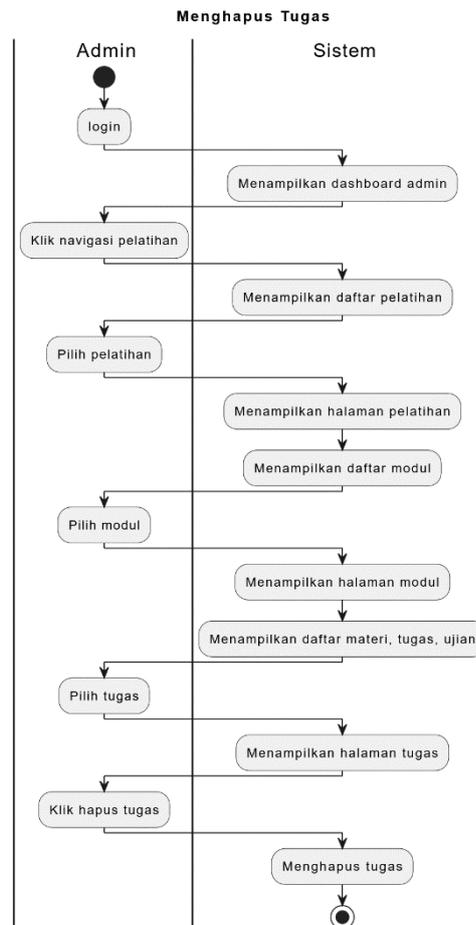
Gambar 20 menggambarkan proses penghapusan materi oleh seorang admin dalam sistem. Proses dimulai dengan admin melakukan *login*, dan sistem menampilkan dasbor admin. Admin kemudian memilih navigasi pelatihan, dan sistem menampilkan daftar pelatihan. Setelah memilih pelatihan yang sesuai, sistem menampilkan halaman pelatihan bersama dengan daftar modul yang terkait. Admin memilih modul yang berisi materi yang ingin dihapus, dan sistem menampilkan halaman modul dengan daftar materi, tugas, dan ujian yang ada. Selanjutnya, admin memilih materi yang ingin dihapus, dan sistem menyediakan opsi untuk menghapus

materi tersebut. Admin mengklik opsi hapus materi, dan sistem akan menghapus materi tersebut dari sistem.



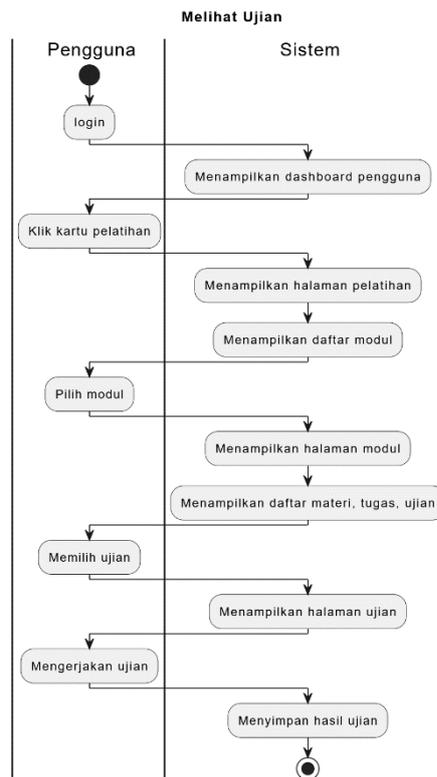
Gambar 21 Menghapus Ujian

Gambar 21 menggambarkan proses penghapusan ujian dalam sistem. Proses dimulai saat admin melakukan *login* ke sistem, yang kemudian menampilkan dasbor admin. Setelah itu, admin memilih navigasi pelatihan, dan sistem menampilkan daftar pelatihan. Admin kemudian memilih pelatihan yang diinginkan, dan sistem menampilkan halaman pelatihan beserta daftar modul. Selanjutnya, admin memilih modul yang berisi ujian yang ingin dihapus. Sistem akan menampilkan halaman modul dengan daftar materi, tugas, dan ujian. Admin memilih ujian yang ingin dihapus, dan sistem akan menghapus ujian tersebut dari sistem.



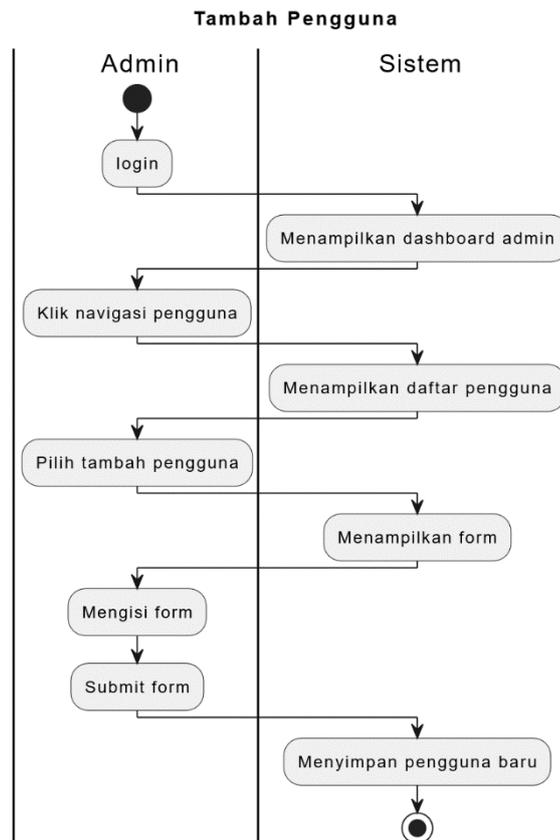
Gambar 22 Menghapus Tugas

Gambar 22 menggambarkan proses penghapusan tugas oleh seorang admin dalam sistem. Proses dimulai dengan admin melakukan *login*, dan sistem menampilkan dasbor admin. Admin kemudian memilih navigasi pelatihan, dan sistem menampilkan daftar pelatihan. Setelah memilih pelatihan yang sesuai, sistem menampilkan halaman pelatihan beserta daftar modul yang terkait. Admin memilih modul yang berisi tugas yang ingin dihapus, dan sistem menampilkan halaman modul dengan daftar materi, tugas, dan ujian. Selanjutnya, admin memilih tugas yang ingin dihapus, dan sistem menyediakan opsi untuk menghapus tugas tersebut. Admin mengklik opsi hapus tugas, dan sistem akan menghapus tugas tersebut dari sistem.



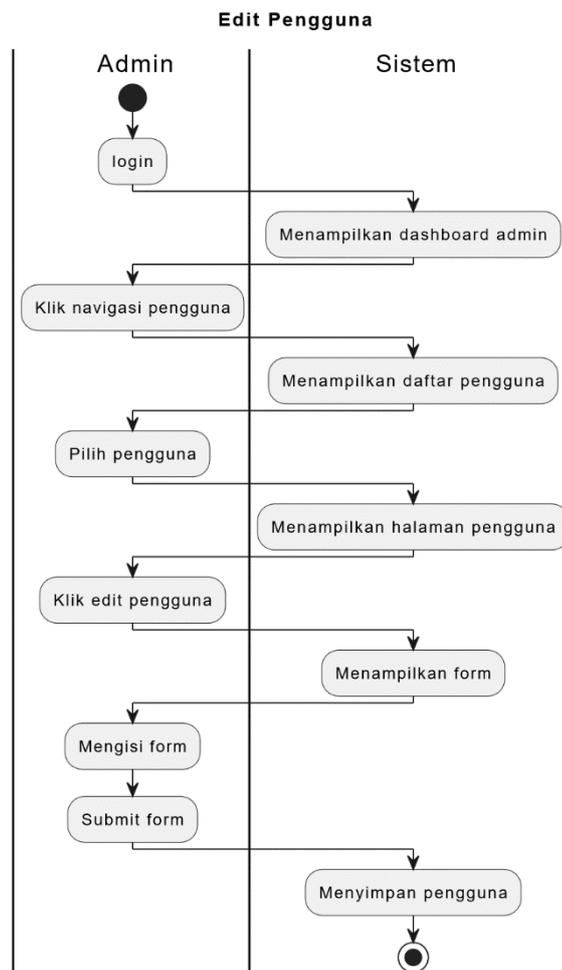
Gambar 23 Melihat Ujian

Gambar 23 menjelaskan proses ketika seorang dosen melihat informasi ujian dalam sistem. Proses dimulai dengan dosen melakukan *login*, kemudian sistem menampilkan dasbor dosen. Selanjutnya, dosen memilih kartu pelatihan dari dasbor, dan sistem menampilkan halaman pelatihan dengan daftar modul yang tersedia. Dosen memilih modul yang diinginkan, dan sistem menampilkan halaman modul dengan daftar materi, tugas, dan ujian. Terakhir, dosen memilih ujian yang ingin dilihat, dan sistem menampilkan halaman ujian dengan informasi terkait ujian tersebut.



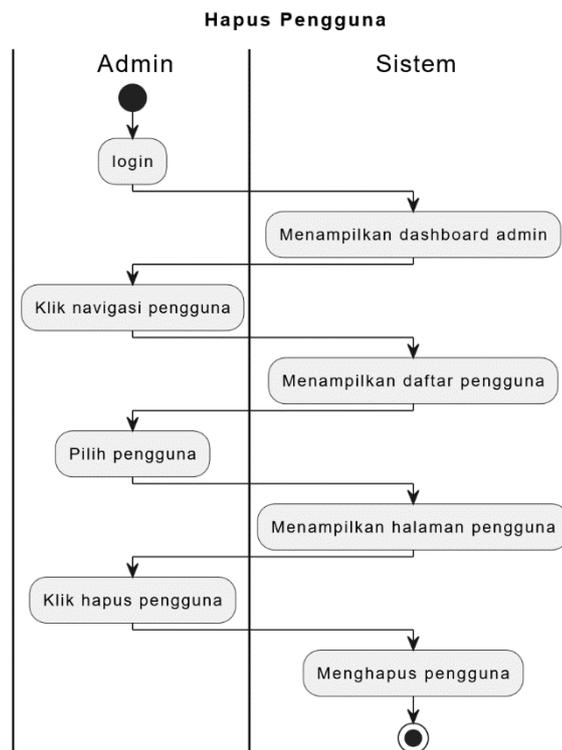
Gambar 24 Membuat Akun

Gambar 24 menggambarkan proses ketika seorang admin menambahkan dosen baru dalam sistem. Proses dimulai dengan admin melakukan *login*, dan sistem menampilkan dasbor admin setelah *login* berhasil. Selanjutnya, admin memilih navigasi untuk mengelola dosen, dan sistem menampilkan daftar dosen yang sudah ada. Admin kemudian memilih opsi untuk menambah dosen baru, dan sistem menampilkan formulir yang memungkinkan admin untuk mengisi detail dosen baru. Admin mengisi formulir tersebut dengan informasi yang diperlukan, lalu mengirimkan formulir tersebut. Setelah itu, sistem menyimpan dosen baru yang telah ditambahkan oleh admin.



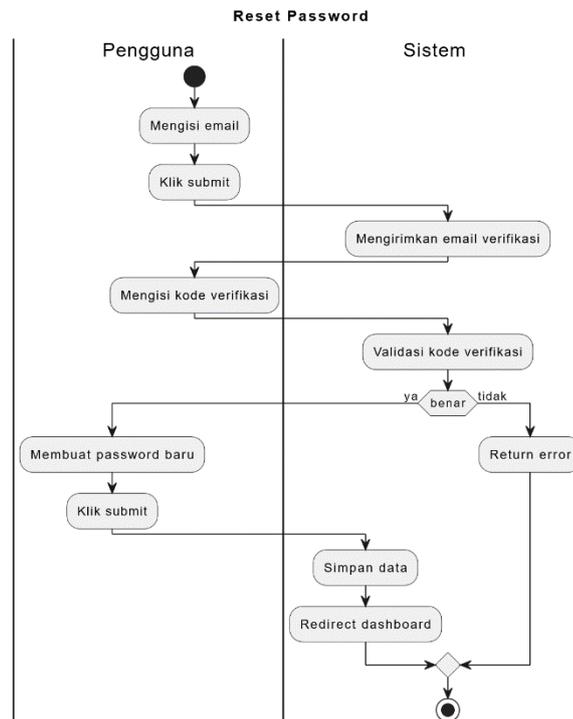
Gambar 25 Mengubah Akun

Gambar 25 menggambarkan proses ketika seorang admin mengedit dosen dalam sistem. Proses dimulai dengan admin melakukan *login*, dan sistem menampilkan dasbor admin setelah *login* berhasil. Selanjutnya, admin memilih navigasi untuk mengelola dosen, dan sistem menampilkan daftar dosen yang sudah ada. Admin kemudian memilih dosen yang ingin diedit, dan sistem menampilkan halaman detail dosen terkait. Admin mengklik opsi untuk mengedit dosen, dan sistem menampilkan formulir yang memungkinkan admin untuk mengubah detail dosen tersebut. Admin mengisi formulir tersebut dengan informasi yang diperlukan, lalu mengirimkan formulir tersebut. Setelah itu, sistem menyimpan perubahan yang telah dilakukan pada dosen tersebut.



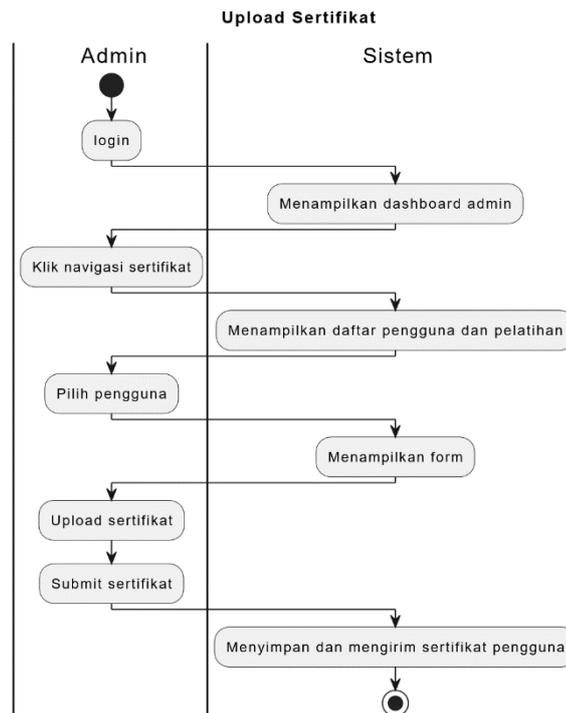
Gambar 26 Menghapus Akun

Gambar 26 menggambarkan proses ketika seorang admin menghapus dosen dalam sistem. Proses dimulai dengan admin melakukan *login*, dan sistem menampilkan dasbor admin setelah *login* berhasil. Selanjutnya, admin memilih navigasi untuk mengelola dosen, dan sistem menampilkan daftar dosen yang sudah ada. Admin kemudian memilih dosen yang ingin dihapus, dan sistem menampilkan halaman detail dosen terkait. Admin mengklik opsi untuk menghapus dosen, dan sistem menghapus dosen tersebut dari sistem.



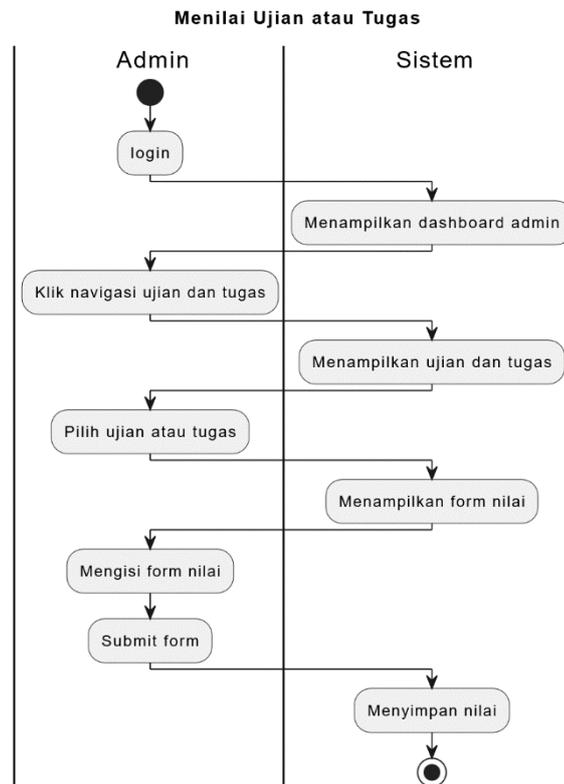
Gambar 27 Reset *Password*

Gambar 27 menggambarkan proses reset *password* oleh seorang dosen dalam sistem. Proses dimulai saat dosen mengisi email mereka dan mengklik tombol submit. Sistem kemudian mengirimkan email verifikasi ke alamat email yang diberikan. Setelah menerima email verifikasi, dosen mengisi kode verifikasi yang diterima dan memasukkan *password* baru. Setelah itu, sistem melakukan validasi terhadap kode verifikasi yang dimasukkan oleh dosen. Jika kode verifikasi benar, dosen membuat *password* baru dan mengklik tombol submit. Sistem kemudian menyimpan data dan mengarahkan dosen ke dasbor. Namun, jika kode verifikasi tidak benar, sistem akan mengembalikan pesan *error* kepada dosen.



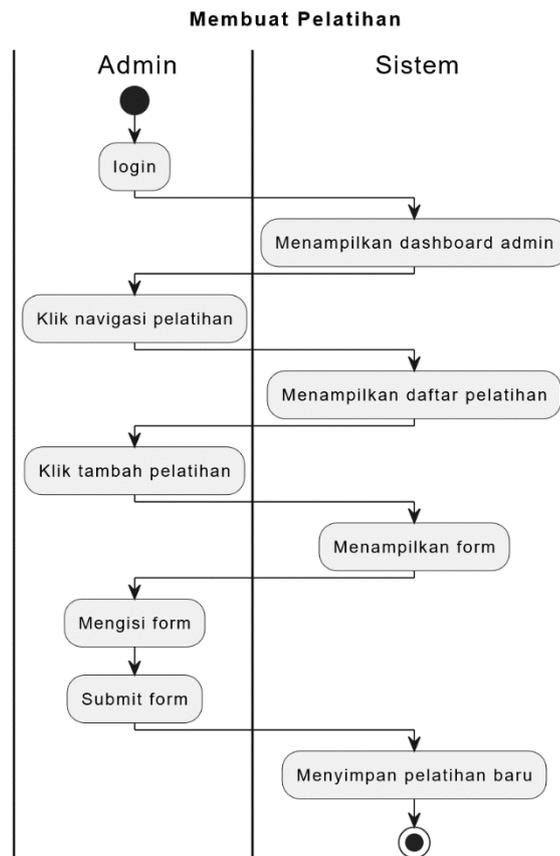
Gambar 28 Upload Sertifikat

Gambar 28 menggambarkan proses saat seorang admin mengunggah sertifikat untuk dosen dalam sistem. Proses dimulai dengan admin melakukan *login*, dan sistem menampilkan dasbor admin setelah *login* berhasil. Selanjutnya, admin memilih navigasi untuk mengelola sertifikat, dan sistem menampilkan daftar dosen dan pelatihan yang terkait. Admin memilih dosen yang akan menerima sertifikat, dan sistem menampilkan formulir unggah yang memungkinkan admin untuk mengunggah sertifikat. Admin mengunggah sertifikat tersebut dan menekan tombol submit. Setelah itu, sistem menyimpan dan mengirim sertifikat kepada dosen terkait.



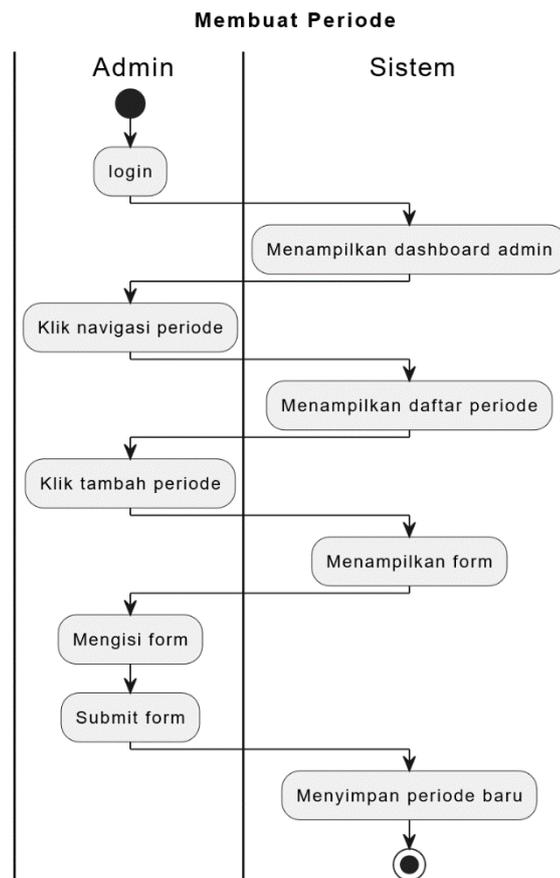
Gambar 29 Menilai Ujian atau Tugas

Gambar 29 menggambarkan proses saat seorang admin menilai ujian atau tugas dalam sistem. Proses dimulai dengan admin melakukan *login*, dan sistem menampilkan dasbor admin setelah *login* berhasil. Selanjutnya, admin memilih navigasi untuk menilai ujian dan tugas, dan sistem menampilkan daftar ujian dan tugas yang tersedia. Admin memilih ujian atau tugas yang akan dinilai, dan sistem menampilkan formulir nilai yang memungkinkan admin untuk memberikan penilaian. Admin mengisi formulir tersebut dengan nilai yang sesuai, lalu mengirimkan formulir tersebut. Setelah itu, sistem menyimpan nilai yang telah diberikan oleh admin.



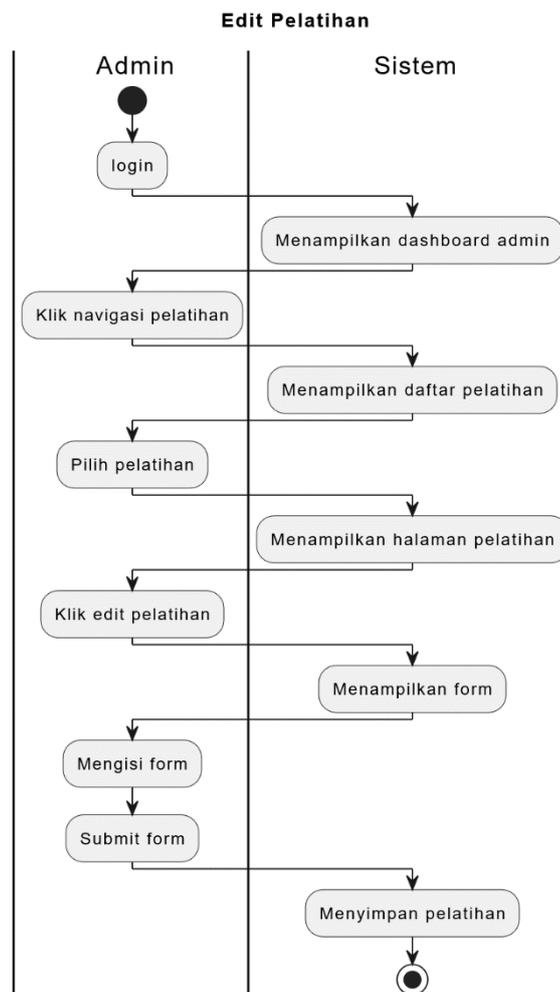
Gambar 30 Membuat Pelatihan

Gambar 30 menggambarkan proses ketika seorang admin membuat pelatihan baru dalam sistem. Proses dimulai dengan admin melakukan *login*, dan sistem menampilkan dasbor admin setelah *login* berhasil. Selanjutnya, admin memilih navigasi untuk membuat pelatihan, dan sistem menampilkan daftar pelatihan yang sudah ada. Admin kemudian memilih opsi untuk menambahkan pelatihan baru, dan sistem menampilkan formulir yang memungkinkan admin untuk mengisi detail pelatihan baru. Admin mengisi formulir tersebut dengan informasi yang diperlukan, lalu mengirimkan formulir tersebut. Setelah itu, sistem menyimpan pelatihan baru yang telah dibuat oleh admin.



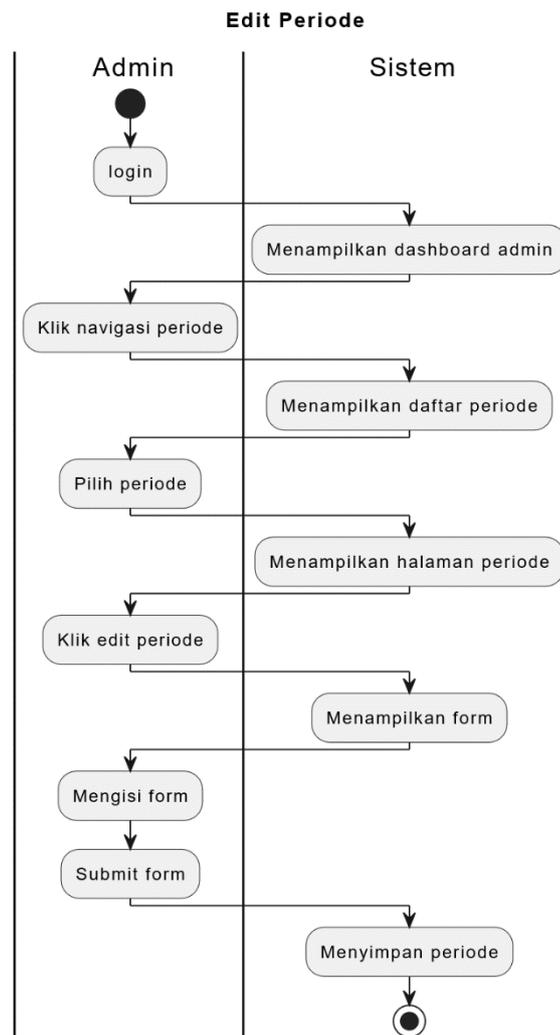
Gambar 31 Membuat Periode

Gambar 31 menggambarkan proses ketika seorang admin membuat periode baru dalam sistem. Proses dimulai dengan admin melakukan *login*, dan sistem menampilkan dasbor admin setelah *login* berhasil. Selanjutnya, admin memilih navigasi untuk membuat periode, dan sistem menampilkan daftar periode yang sudah ada. Admin kemudian memilih opsi untuk menambahkan periode baru, dan sistem menampilkan formulir yang memungkinkan admin untuk mengisi detail periode baru. Admin mengisi formulir tersebut dengan informasi yang diperlukan, lalu mengirimkan formulir tersebut. Setelah itu, sistem menyimpan periode baru yang telah dibuat oleh admin.



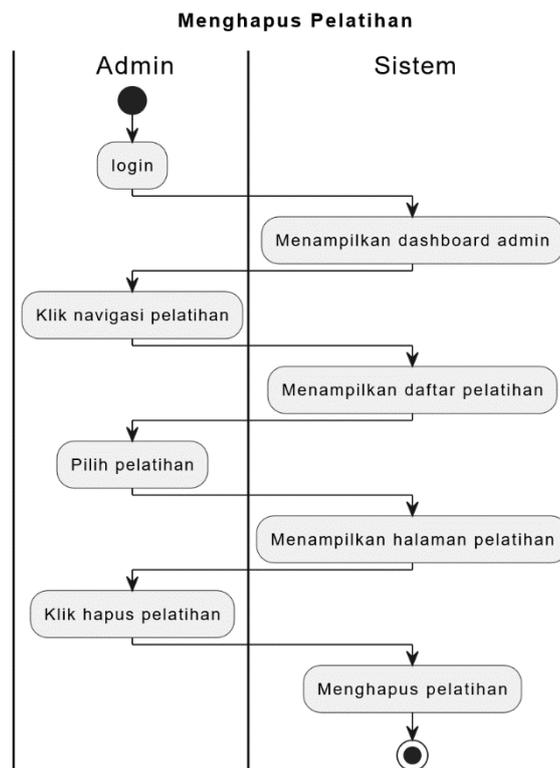
Gambar 32 Edit Pelatihan

Gambar 32 menggambarkan proses ketika seorang admin mengedit pelatihan dalam sistem. Proses dimulai dengan admin melakukan *login*, dan sistem menampilkan dasbor admin setelah *login* berhasil. Selanjutnya, admin memilih navigasi untuk mengelola pelatihan, dan sistem menampilkan daftar pelatihan yang sudah ada. Admin memilih pelatihan yang ingin diedit, dan sistem menampilkan halaman detail pelatihan terkait. Admin mengklik opsi untuk mengedit pelatihan, dan sistem menampilkan formulir yang memungkinkan admin untuk mengubah detail pelatihan tersebut. Admin mengisi formulir tersebut dengan informasi yang diperlukan, lalu mengirimkan formulir tersebut. Setelah itu, sistem menyimpan perubahan yang telah dilakukan pada pelatihan tersebut.



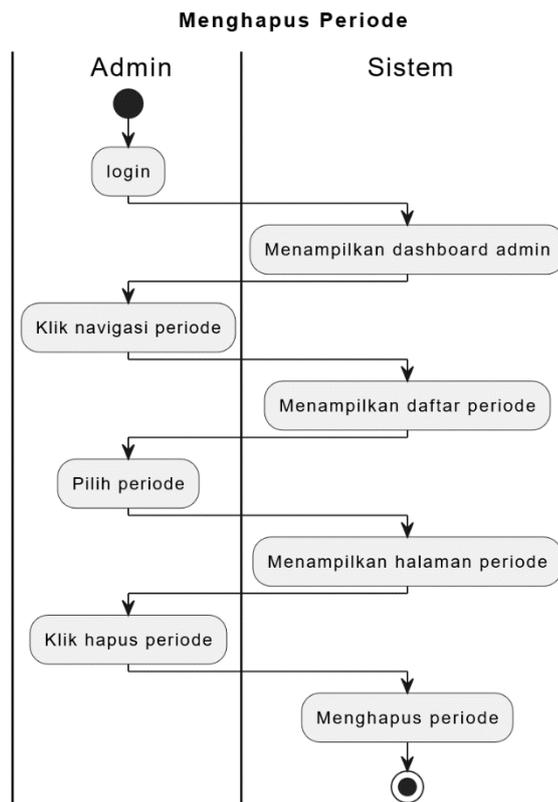
Gambar 33 Edit Periode

Gambar 33 menggambarkan proses ketika seorang admin mengedit periode dalam sistem. Proses dimulai dengan admin melakukan *login*, dan sistem menampilkan dasbor admin setelah *login* berhasil. Selanjutnya, admin memilih navigasi untuk mengelola periode, dan sistem menampilkan daftar periode yang sudah ada. Admin memilih periode yang ingin diedit, dan sistem menampilkan halaman detail periode terkait. Admin mengeklik opsi untuk mengedit periode, dan sistem menampilkan formulir yang memungkinkan admin untuk mengubah detail periode tersebut. Admin mengisi formulir tersebut dengan informasi yang diperlukan, lalu mengirimkan formulir tersebut. Setelah itu, sistem menyimpan perubahan yang telah dilakukan pada periode tersebut.



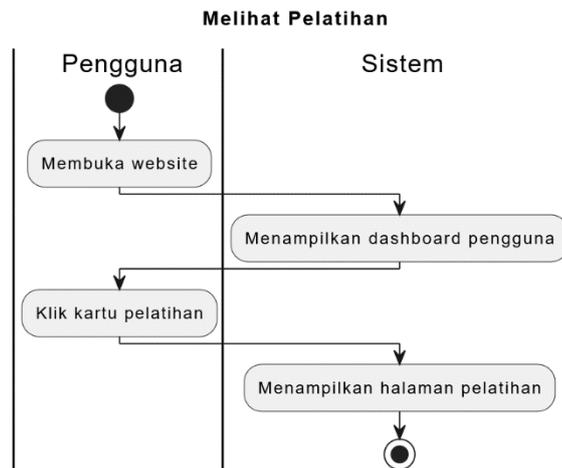
Gambar 34 Menghapus Pelatihan

Gambar 34 menggambarkan proses ketika seorang admin menghapus pelatihan dalam sistem. Proses dimulai dengan admin melakukan *login*, dan sistem menampilkan dasbor admin setelah *login* berhasil. Selanjutnya, admin memilih navigasi untuk mengelola pelatihan, dan sistem menampilkan daftar pelatihan yang sudah ada. Admin memilih pelatihan yang ingin dihapus, dan sistem menampilkan halaman detail pelatihan terkait. Admin mengeklik opsi untuk menghapus pelatihan, dan sistem menghapus pelatihan tersebut dari sistem.



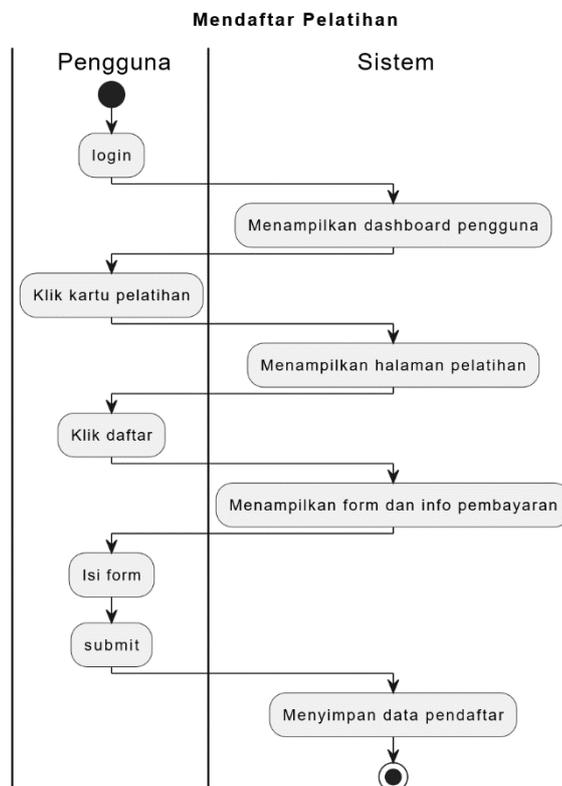
Gambar 35 Menghapus Periode

Gambar 35 menggambarkan proses ketika seorang admin menghapus periode dalam sistem. Proses dimulai dengan admin melakukan *login*, dan sistem menampilkan dasbor admin setelah *login* berhasil. Selanjutnya, admin memilih navigasi untuk mengelola periode, dan sistem menampilkan daftar periode yang sudah ada. Admin memilih periode yang ingin dihapus, dan sistem menampilkan halaman detail periode terkait. Admin mengeklik opsi untuk menghapus periode, dan sistem menghapus periode tersebut dari sistem.



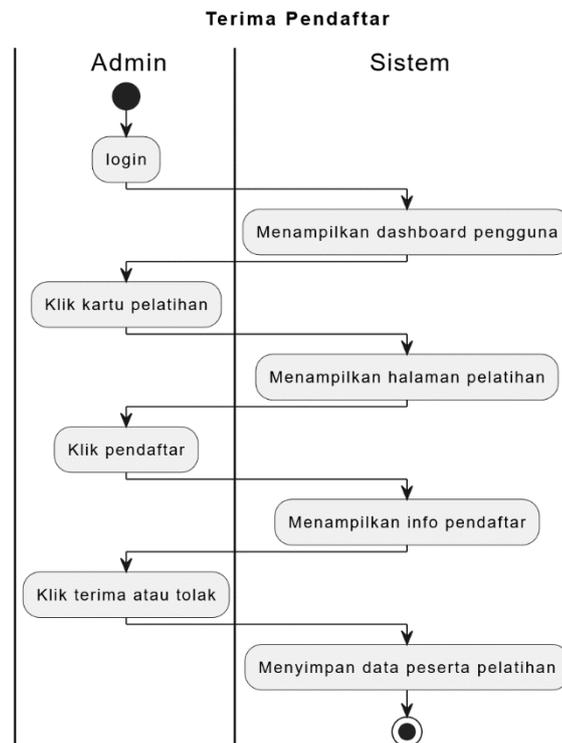
Gambar 36 Melihat Pelatihan

Gambar 36 menggambarkan proses ketika seorang dosen melihat pelatihan dalam sistem. Proses dimulai saat dosen membuka situs web yang memuat sistem, dan sistem menampilkan dasbor dosen. Dosen kemudian memilih kartu pelatihan yang menarik minatnya, dan sistem menampilkan halaman pelatihan yang berisi informasi detail tentang pelatihan tersebut.



Gambar 37 Mendaftar Pelatihan

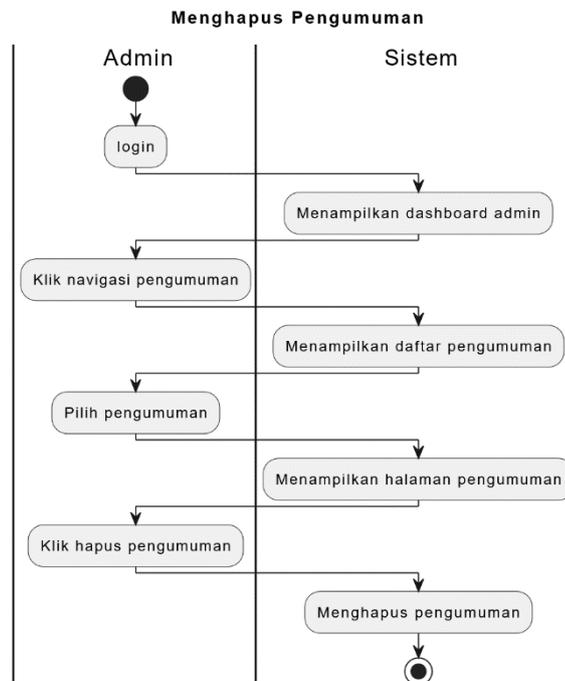
Gambar 37 menggambarkan proses ketika seorang dosen mendaftar untuk sebuah pelatihan dalam sistem. Proses dimulai saat dosen melakukan *login* ke dalam sistem, dan sistem menampilkan dasbor dosen setelah *login* berhasil. Dosen kemudian memilih kartu pelatihan yang diminati dari dasbor, dan sistem menampilkan halaman pelatihan yang berisi informasi detail tentang pelatihan tersebut. Setelah itu, dosen memilih opsi untuk mendaftar, dan sistem menampilkan formulir pendaftaran bersama informasi pembayaran yang diperlukan. Dosen mengisi formulir pendaftaran dengan data yang dibutuhkan, lalu mengirimkan formulir tersebut. Setelah dosen mengirimkan formulir pendaftaran, sistem menyimpan data pendaftar.



Gambar 38 Terima Pendaftar

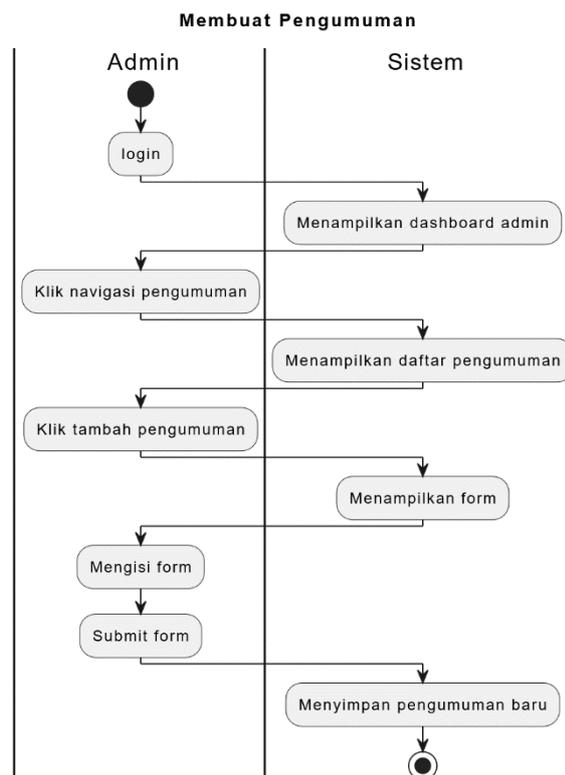
Gambar 38 menggambarkan proses ketika seorang admin menerima atau menolak pendaftar dalam sebuah pelatihan dalam sistem. Proses dimulai saat admin melakukan *login* ke dalam sistem, dan sistem menampilkan dasbor admin setelah *login* berhasil. Admin kemudian memilih kartu pelatihan yang diminati dari dasbor, dan sistem menampilkan halaman pelatihan yang berisi informasi detail tentang pelatihan tersebut. Selanjutnya, admin memilih opsi untuk melihat daftar pendaftar,

dan sistem menampilkan informasi tentang pendaftar yang telah mendaftar untuk pelatihan tersebut. Admin kemudian memilih apakah akan menerima atau menolak setiap pendaftar, dan sistem menyimpan data peserta pelatihan sesuai dengan keputusan admin.



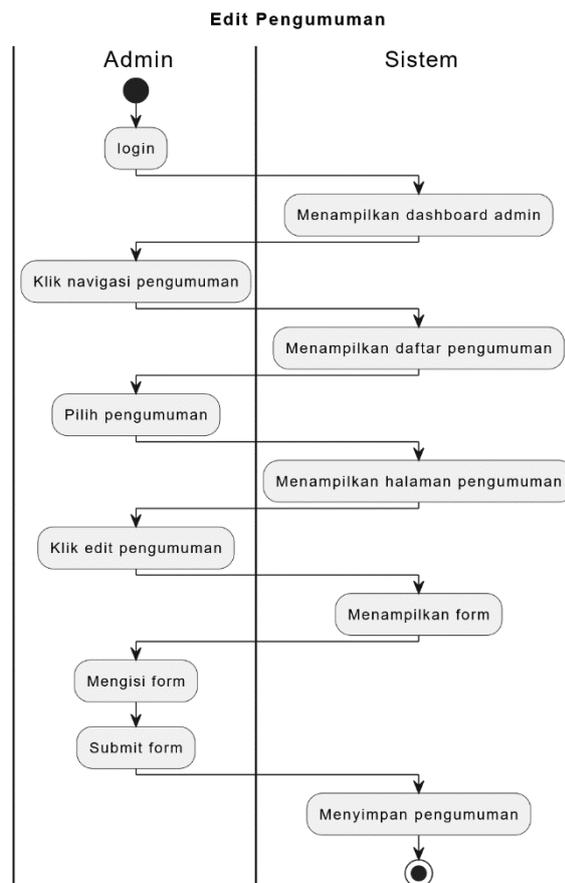
Gambar 39 Menghapus Pengumuman

Gambar 39 menjelaskan proses penghapusan pengumuman oleh seorang admin dalam sistem. Proses dimulai dengan admin melakukan *login*, dan sistem menampilkan dasbor admin. Admin kemudian memilih navigasi pengumuman, dan sistem menampilkan daftar pengumuman yang tersedia. Setelah memilih pengumuman yang ingin dihapus, sistem menampilkan halaman pengumuman tersebut. Admin kemudian mengklik opsi hapus pengumuman, dan sistem akan menghapus pengumuman tersebut dari sistem.



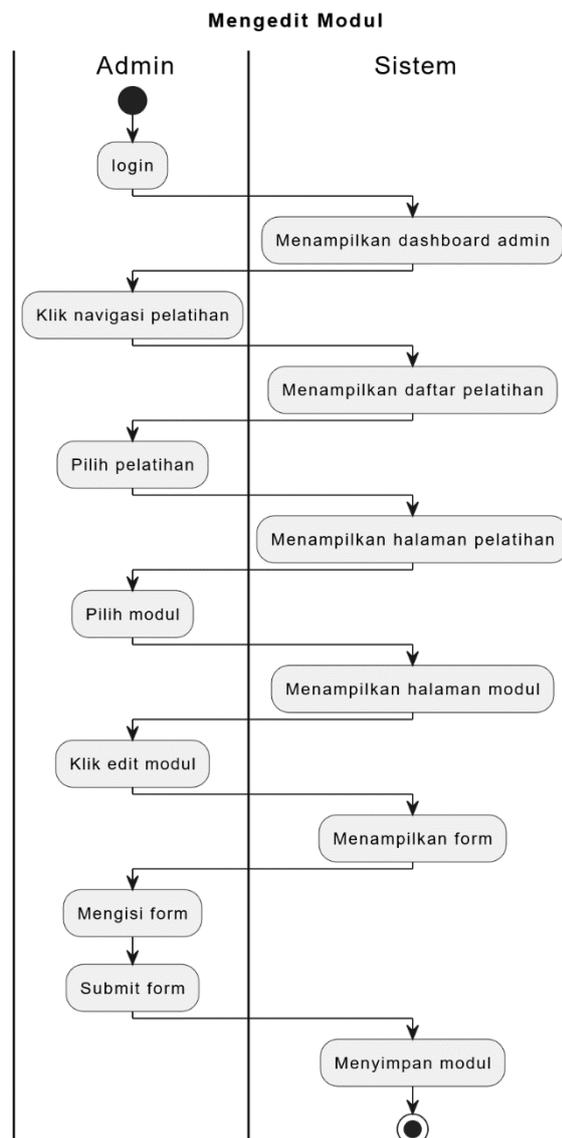
Gambar 40 Membuat Pengumuman

Gambar 40 menggambarkan proses pembuatan pengumuman oleh seorang admin dalam sistem. Proses dimulai dengan admin melakukan *login*, dan sistem menampilkan dasbor admin. Admin kemudian memilih navigasi pengumuman, dan sistem menampilkan daftar pengumuman yang sudah ada. Setelah itu, admin mengeklik opsi tambah pengumuman, dan sistem menampilkan formulir pembuatan pengumuman. Admin mengisi formulir dengan informasi yang relevan, kemudian mensubmit formulir tersebut. Akhirnya, sistem menyimpan pengumuman baru yang telah dibuat oleh admin.



Gambar 41 Edit Pengumuman

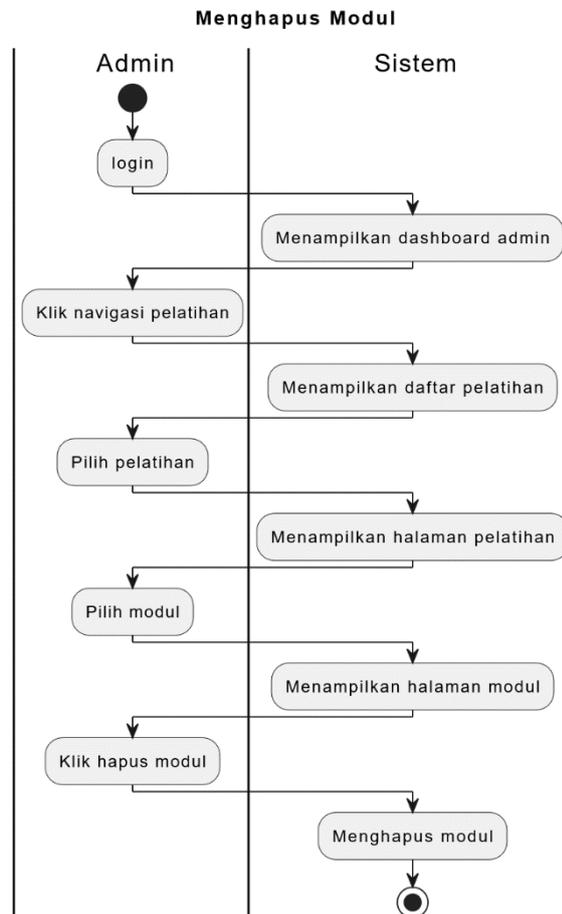
Gambar 41 menjelaskan proses pengeditan pengumuman oleh seorang admin dalam sistem. Proses dimulai dengan admin melakukan *login*, dan sistem menampilkan dasbor admin. Admin kemudian memilih navigasi pengumuman, dan sistem menampilkan daftar pengumuman yang sudah ada. Setelah memilih pengumuman yang ingin diedit, sistem menampilkan halaman pengumuman tersebut. Admin mengeklik opsi edit pengumuman, dan sistem menampilkan formulir pengeditan pengumuman. Admin mengisi formulir dengan perubahan yang diinginkan, kemudian mensubmit formulir tersebut. Akhirnya, sistem menyimpan pengumuman yang telah diedit oleh admin.



Gambar 42 Mengedit Modul

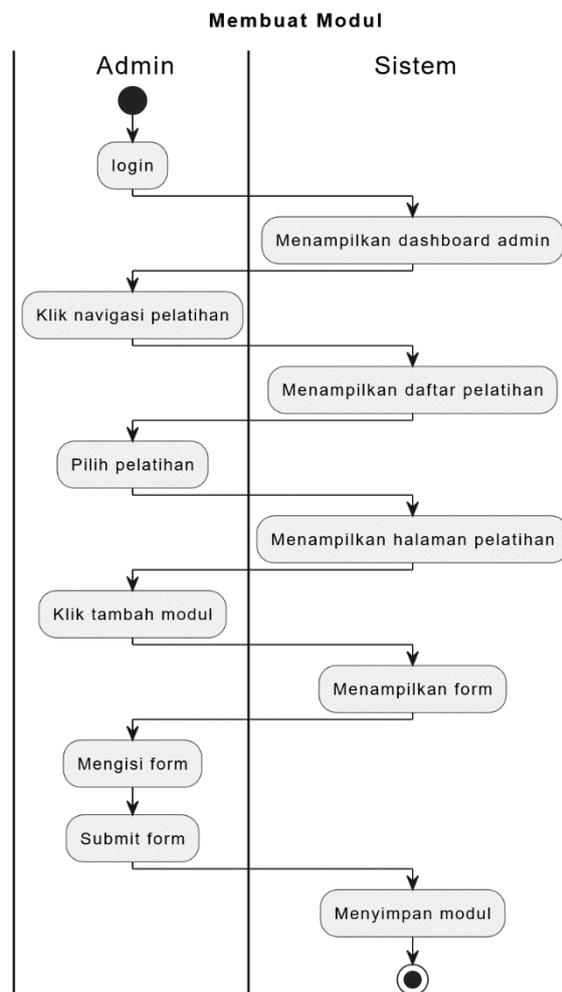
Gambar 42 menjelaskan proses pengeditan modul oleh seorang admin dalam sistem. Proses dimulai dengan admin melakukan *login*, dan sistem menampilkan dasbor admin. Admin kemudian memilih navigasi pelatihan, dan sistem menampilkan daftar pelatihan yang tersedia. Setelah memilih pelatihan yang sesuai, sistem menampilkan halaman pelatihan bersama dengan daftar modul yang terkait. Admin memilih modul yang ingin diedit, dan sistem menampilkan halaman modul tersebut. Selanjutnya, admin mengeklik opsi edit modul, dan sistem menampilkan formulir pengeditan modul. Admin mengisi formulir dengan

perubahan yang diinginkan, kemudian mensubmit formulir tersebut. Akhirnya, sistem menyimpan modul yang telah diedit oleh admin.



Gambar 43 Menghapus Modul

Gambar 43 menggambarkan proses penghapusan modul oleh seorang admin dalam sistem. Proses dimulai dengan admin melakukan *login*, dan sistem menampilkan dasbor admin. Admin kemudian memilih navigasi pelatihan, dan sistem menampilkan daftar pelatihan yang tersedia. Setelah memilih pelatihan yang sesuai, sistem menampilkan halaman pelatihan bersama dengan daftar modul yang terkait. Admin memilih modul yang ingin dihapus, dan sistem menampilkan halaman modul tersebut. Selanjutnya, admin mengeklik opsi hapus modul, dan sistem akan menghapus modul tersebut dari sistem.



Gambar 44 Membuat Modul

Gambar 44 menjelaskan proses pembuatan modul oleh seorang admin dalam sistem. Proses dimulai dengan admin melakukan *login*, dan sistem menampilkan dasbor admin. Admin kemudian memilih navigasi pelatihan, dan sistem menampilkan daftar pelatihan yang tersedia. Setelah memilih pelatihan yang sesuai, sistem menampilkan halaman pelatihan bersama dengan opsi untuk menambah modul. Admin mengeklik opsi tambah modul, dan sistem menampilkan formulir pembuatan modul. Admin mengisi formulir dengan informasi yang relevan, kemudian mensubmit formulir tersebut. Akhirnya, sistem menyimpan modul baru yang telah dibuat oleh admin.





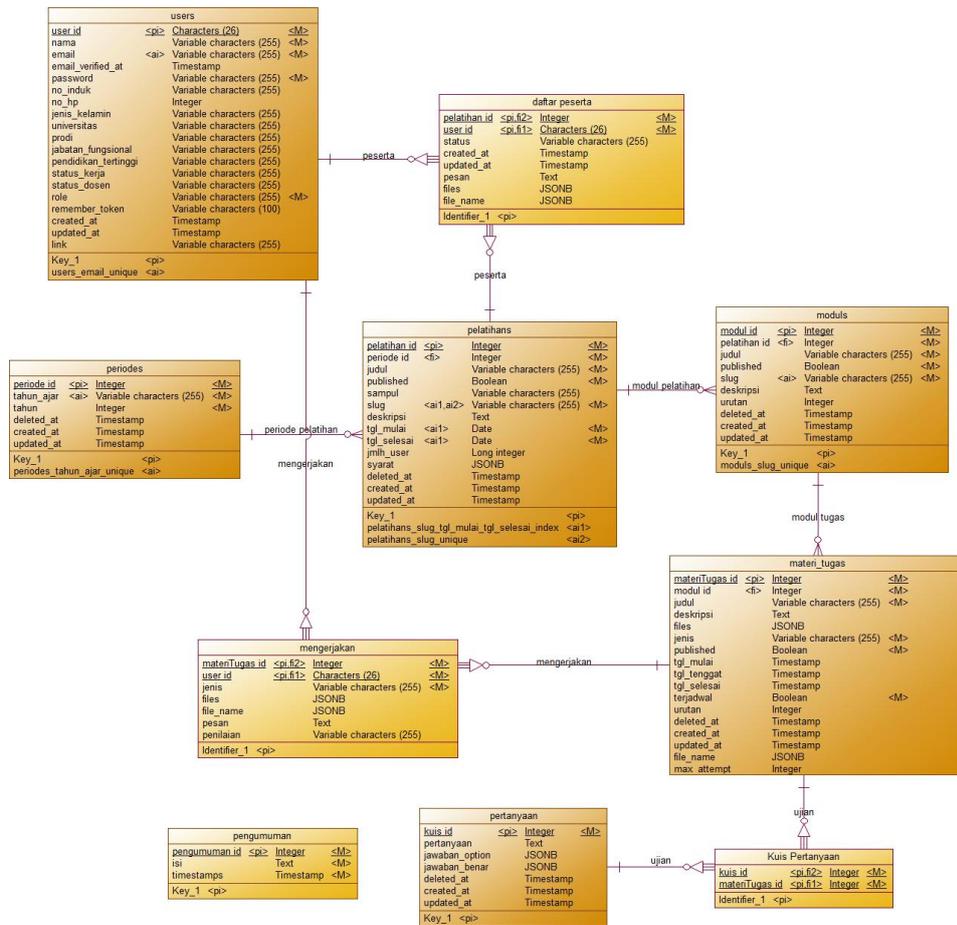
Gambar 46 Class Diagram Panel Dosen

Class diagram pada gambar 45 dan 46 merepresentasikan struktur inheritance dan pengelolaan resource dalam sistem manajemen berbasis resource. Diagram ini terdiri dari komponen-komponen utama berikut:

1. **Macroable:** Kelas yang memungkinkan penambahan metode dinamis melalui macro. Metode seperti `macro()`, `__call()`, dan `__callStatic()` digunakan untuk menambah fleksibilitas eksekusi metode, baik secara instans maupun statis.
2. **Resource:** Kelas dasar bagi resource lain dalam sistem. Resource-resource seperti `BankSoalResource`, `ChatbotDatasResource`, dan `SertifikatResource` mewarisi kelas ini untuk mengimplementasikan fitur seperti tabel, form, navigasi, otorisasi, dan pencarian global.
3. **NestedResource:** Menyediakan dukungan navigasi bertingkat dengan fitur breadcrumbs dan relasi ancestor. Resource seperti `MateriResource`, `ModulResource`, dan `PelatihanResource` mewarisi kelas ini untuk mengelola resource hierarkis yang kompleks.

4. Resource Individu:
  - a. BankSoalResource, ChatbotDatasResource: Mengelola soal dan data chatbot, mendukung operasi seperti tabel, form, navigasi, dan pencarian global.
  - b. KuisResource, DiskusiResource, MateriResource, ModulResource, TugasResource: Menambahkan fitur infolist dan relasi, serta beberapa resource ini memiliki struktur navigasi bertingkat.
  - c. PelatihanResource, PendaftaranResource: Mengelola pelatihan dan pendaftaran, dengan fitur tambahan seperti otorisasi dan pencarian global.
  - d. SertifikatResource, PeriodeResource: Mengelola sertifikat dan periode pelatihan atau kegiatan, dilengkapi dengan form, navigasi, dan pencarian global.
  - e. UserResource, StatUserResource: Berhubungan dengan data pengguna dan statistik, serta kontrol akses yang lebih ketat.

### 3.6 Entity Relationship Diagram



Gambar 47 ERD

Pada gambar 47 menunjukkan diagram aliran konten *database*. Diagram ini menunjukkan hubungan antara 5 tabel, yaitu:

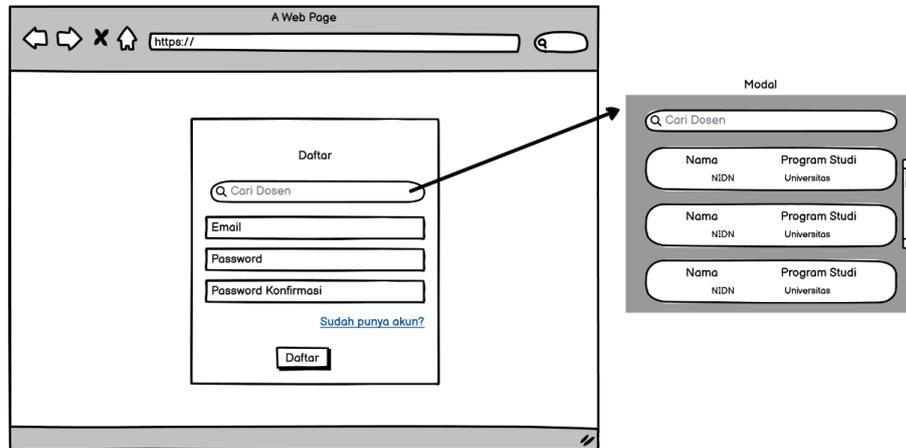
- *User*, menyimpan informasi mengenai user atau dosen
- Materi Tugas, menyimpan informasi mengenai tugas atau materi.
- Pelatihan, menyimpan informasi mengenai pelatihan.
- Pertanyaan, menyimpan informasi berupa pertanyaan yang dapat digunakan kembali.
- Pengumuman, menyimpan informasi mengenai pengumuman.
- Periode, menyimpan informasi mengenai periode.
- Modul, menyimpan informasi berupa judul dan deskripsi modul.

Hubungan antara tabel-tabel tersebut adalah sebagai berikut:

- Tabel *User* memiliki hubungan M:M dengan Tabel Materi Tugas. Hal ini berarti bahwa satu dosen dapat melihat dan mengerjakan banyak materi atau tugas, dan satu materi atau tugas dapat dimiliki oleh banyak dosen. Hubungan ini diwakili oleh tabel *mengerjakan* yang berfungsi untuk mengetahui materi atau tugas apa saja yang sudah dilihat dan dikerjakan oleh dosen.
- Tabel *User* memiliki hubungan M:M dengan Tabel Pelatihan. Hal ini berarti bahwa satu dosen dapat mengikuti banyak pelatihan, dan satu kelas dapat diikuti oleh banyak dosen. Hubungan ini diwakili oleh tabel *daftar peserta* pelatihan.
- Tabel materi Tugas memiliki hubungan M:M dengan Tabel Pertanyaan. Hal ini berarti bahwa satu ujian dapat memiliki banyak pertanyaan. Hubungan ini diwakili oleh tabel *Kuis Pertanyaan* yang berfungsi untuk menghubungkan pertanyaan dengan tugas ujian.
- Tabel Periode memiliki hubungan 1:M dengan tabel Pelatihan. Hal ini berarti bahwa dalam satu periode boleh memiliki banyak pelatihan.
- Tabel Pelatihan memiliki hubungan 1:M dengan tabel Modul. Hal ini berarti bahwa dalam satu pelatihan boleh memiliki banyak modul.
- Tabel Modul memiliki hubungan 1:M dengan tabel Materi Tugas. Hal ini berarti bahwa dalam satu modul boleh memiliki banyak materi tugas.
- Tabel materi tugas memiliki hubungan 1:M dengan tabel kuis. Hal ini berarti bahwa dalam satu materi tugas boleh memiliki banyak pertanyaan.

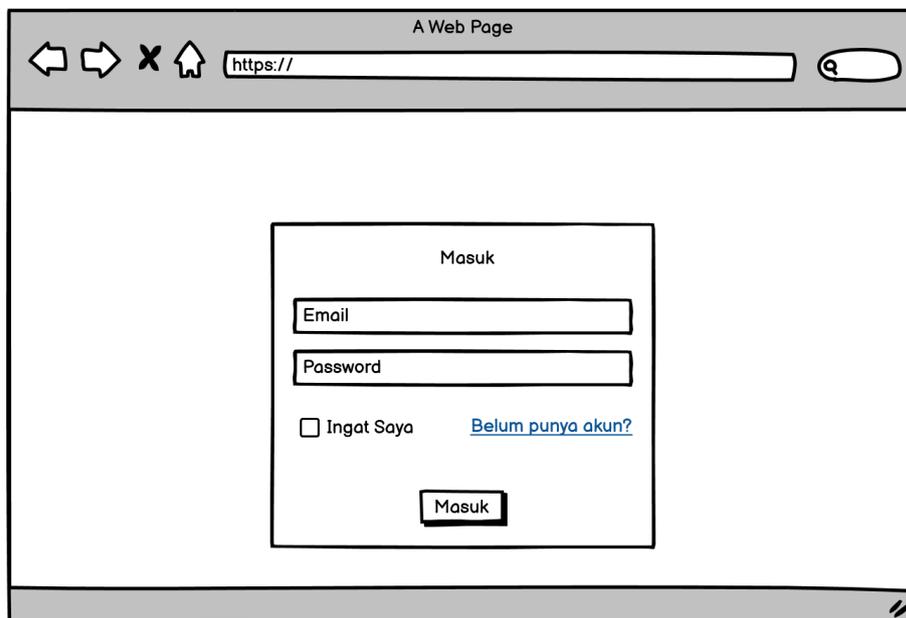
### 3.7 Wireframe

Pada gambar di bawah ini adalah gambaran tentang rancangan aplikasi yang akan dibuat:



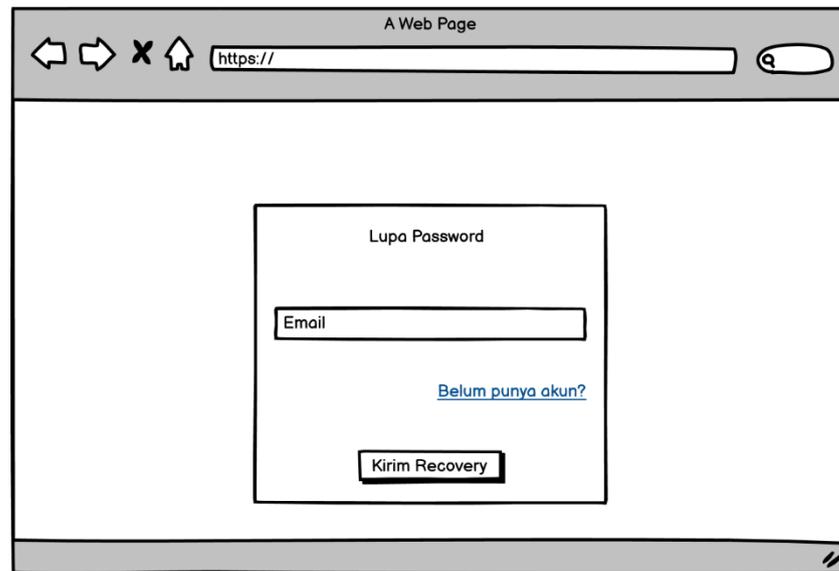
Gambar 48 Halaman Register

Pada gambar 48, dosen dapat melakukan registrasi akun dengan mencari menggunakan nomor induk atau nama pada kolom pencarian, lalu mengisi formulir email dan *password*.



Gambar 49 Halaman *Login*

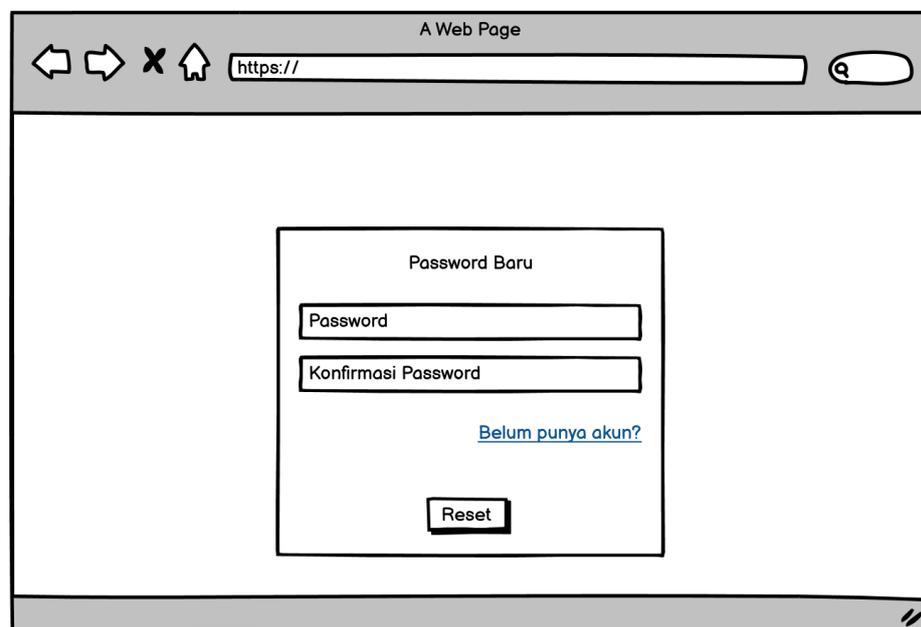
Pada gambar 49 dosen dapat melakukan *login* dengan memasukkan email dan password untuk masuk ke *dasbor* dosen. Apabila email yang terdaftar yaitu email admin maka akan masuk ke dasbor admin.



A screenshot of a web browser window titled "A Web Page". The address bar shows "https://". The main content area displays a form titled "Lupa Password". The form contains an "Email" input field, a blue link labeled "Belum punya akun?", and a "Kirim Recovery" button.

Gambar 50 Halaman lupa *password*

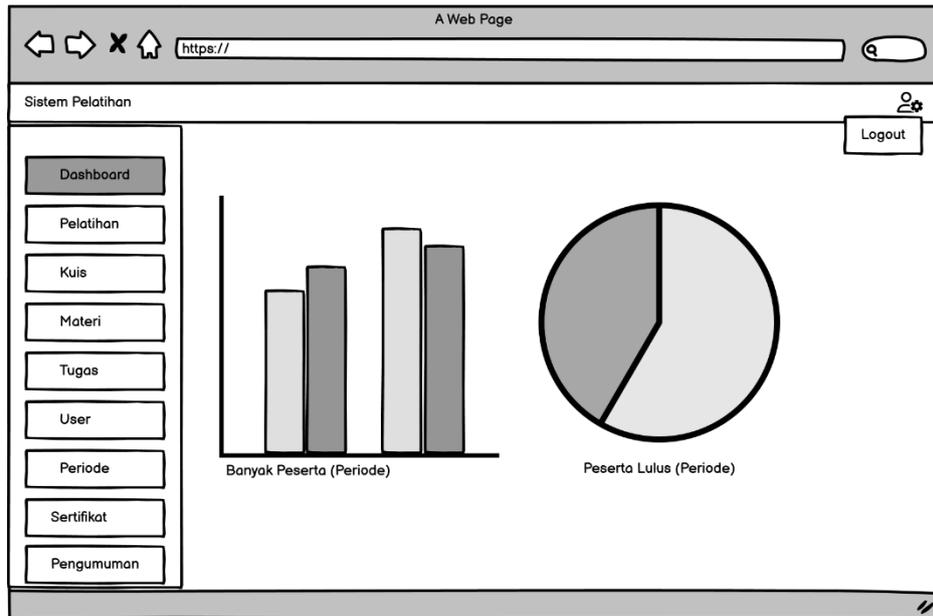
Pada gambar 50, dosen dapat melakukan reset *password* menggunakan email yang kemudian akan mendapatkan tautan yang dikirimkan melalui email.



A screenshot of a web browser window titled "A Web Page". The address bar shows "https://". The main content area displays a form titled "Password Baru". The form contains two input fields: "Password" and "Konfirmasi Password", a blue link labeled "Belum punya akun?", and a "Reset" button.

Gambar 51 Halaman *Password Baru*

Pada gambar 51, setelah dosen mereset *password* menggunakan email, dosen mendapatkan tautan dalam email yang akan mengarahkan ke halaman ini, kemudian dosen dapat membuat *password* baru.



Gambar 52 Halaman Dasbor Admin

Pada gambar 52, adalah gambaran mengenai dasbor untuk admin yang berisikan grafik-grafik informasi seperti grafik banyak peserta atau banyak peserta lulus.

Pelatihan	Tgl Selesai	Tgl Mulai
Pelatihan 1	20 Mar 2021	21 Jan 2021
Pelatihan 1	20 Mar 2021	21 Jan 2021
Pelatihan 1	20 Mar 2021	21 Jan 2021
Pelatihan 1	20 Mar 2021	21 Jan 2021
Pelatihan 1	20 Mar 2021	21 Jan 2021

Create << < 1 2 3 4 5 6 7 8 \* table dump

Gambar 53 Halaman Admin Pelatihan

Pada gambar 53 merupakan daftar pelatihan yang sudah dibuat, admin dapat menambah pelatihan baru dengan tombol 'create'.

A Web Page

https://

Sistem Pelatihan

Logout

Home > Pelatihan > Form Pelatihan

text input label

text input label

date chooser label

date chooser label

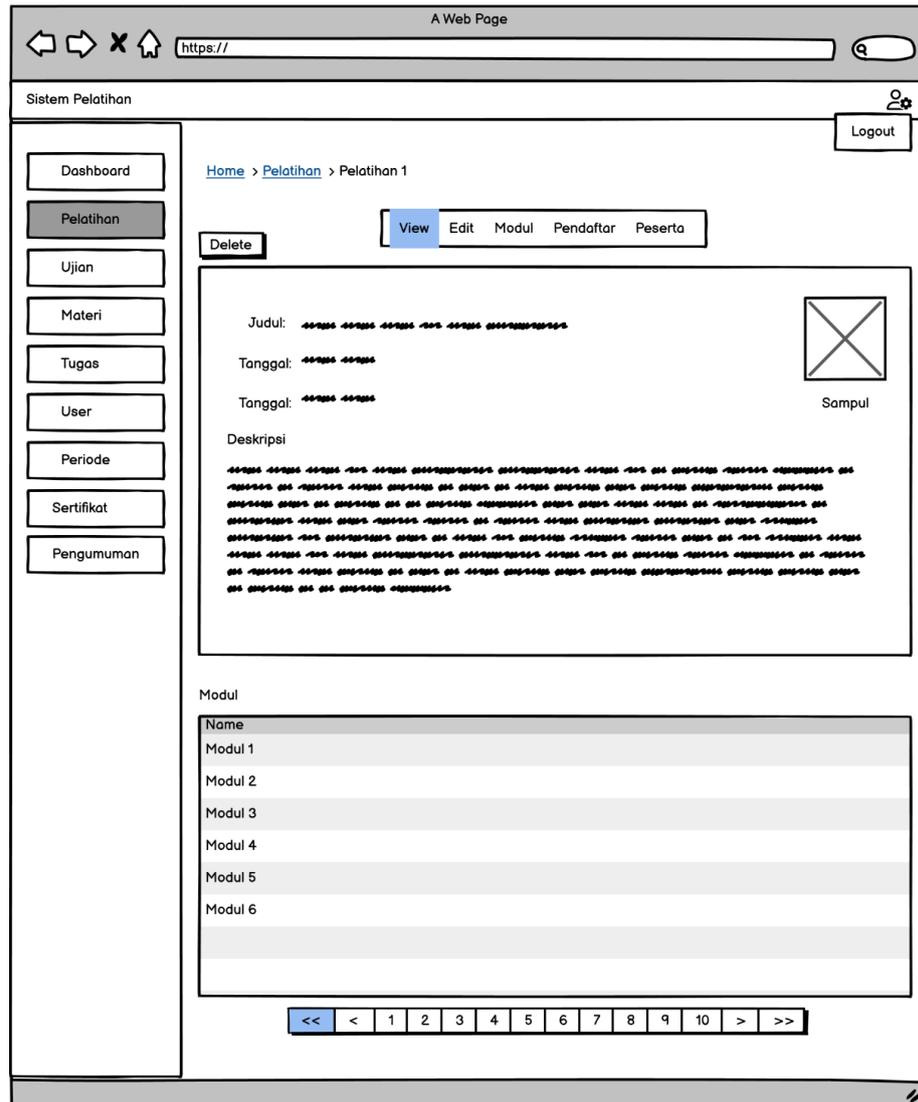
ComboBox

text field label

Submit

Gambar 54 Halaman Admin Form Pelatihan

Pada gambar 54 merupakan halaman formulir pelatihan jika admin ingin menambah pelatihan baru mengisi *input* seperti tanggal mulai, tanggal selesai, judul, dan deskripsi.



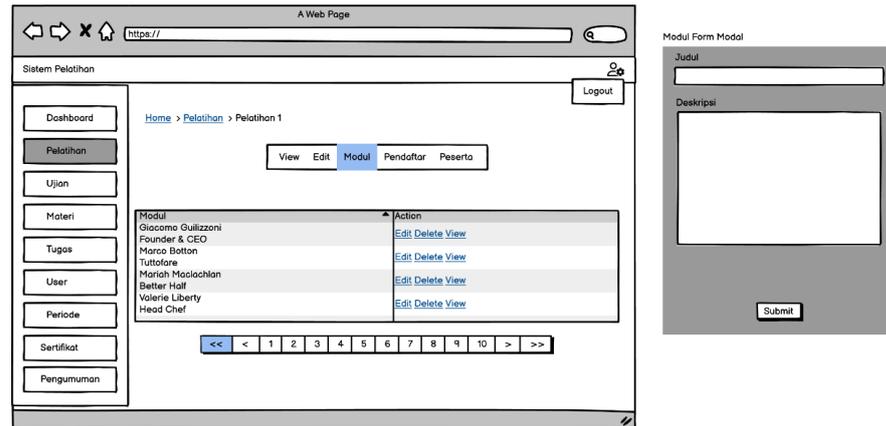
Gambar 55 Halaman Admin Detail Pelatihan

Pada gambar 55 merupakan halaman detail pelatihan yang muncul setelah admin memilih salah satu pelatihan daftar pelatihan. Halaman ini berisikan deskripsi serta informasi pelatihan dan modul terkait.

The image shows a web browser window titled "A Web Page" with a URL bar containing "https://". The page content is titled "Sistem Pelatihan" and includes a "Logout" button in the top right corner. A breadcrumb trail reads "Home > Pelatihan > Pelatihan 1". Below this is a navigation menu with tabs: "View", "Edit" (which is highlighted in blue), "Modul", "Pendaftar", and "Peserta". The main content area contains several form elements: two "text input label" fields, two "date chooser label" fields (each with a date input and a calendar icon), a "ComboBox" dropdown, a "text field label" (a large text area), and an "Upload a file:" section with a file input field and a "Browse..." button. A "Submit" button is located at the bottom center of the form area.

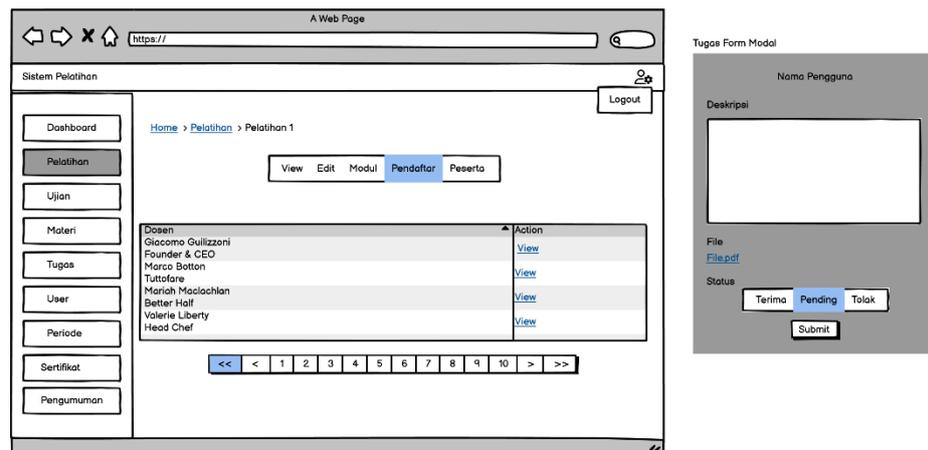
Gambar 56 Halaman Admin Edit Pelatihan

Pada gambar 56 merupakan halaman dimana admin dapat melakukan perubahan informasi mengenai pelatihan dengan memilih tab edit pada halaman detail pelatihan.



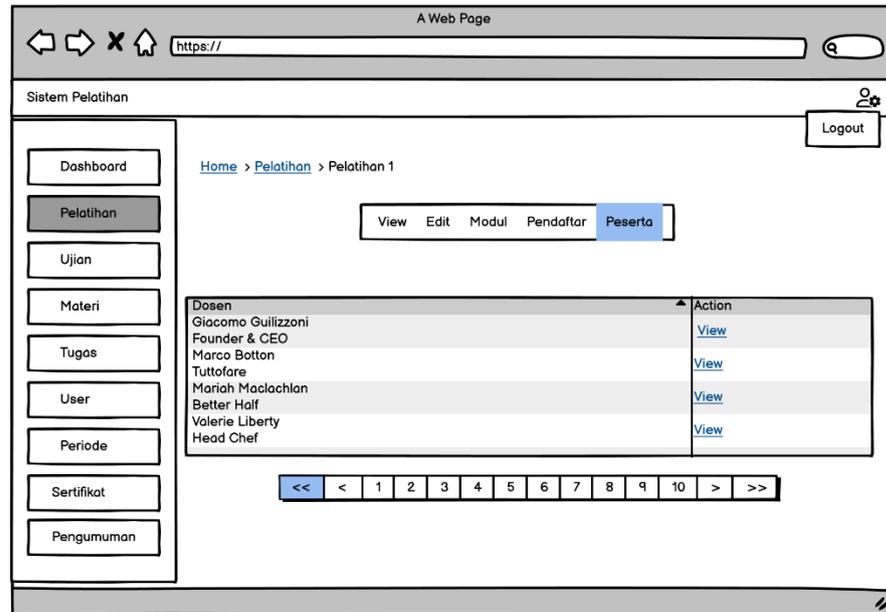
Gambar 57 Halaman Admin Modul

Pada gambar 57 merupakan halaman daftar modul yang muncul setelah admin memilih tab modul pada detail pelatihan. Halaman ini berisikan daftar modul yang terkait dengan pelatihan. Admin juga dapat melihat isi modul ataupun menghapus, mengubah, dan membuat modul.



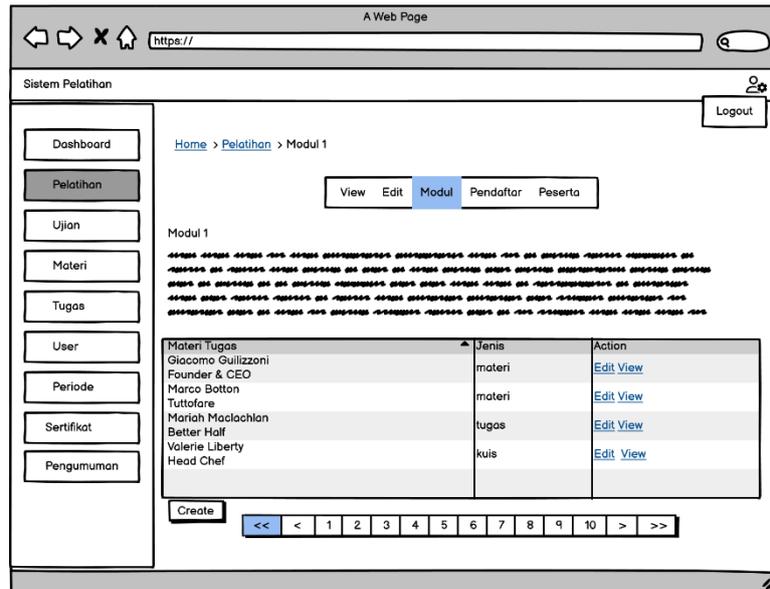
Gambar 58 Halaman Admin Pendaftar

Pada gambar 58 merupakan halaman daftar dari para pendaftar pelatihan yang muncul setelah admin memilih tab pendaftar. Admin dapat melihat detail dari pendaftar dan dapat menentukan apakah pendaftar lulus menjadi peserta atau tidak.



Gambar 59 Halaman Admin Peserta

Pada gambar 59 merupakan halaman daftar dari peserta pada pelatihan terkait. Halaman ini muncul setelah admin memilih tab peserta. Admin juga dapat melihat detail peserta.



**Tugas Form Modal**

Judul:

Tgl Mulai:    Tgl Tenggat:    Tgl Selesai:

Deskripsi:

Submit

**Materi Form Modal**

Judul:

Deskripsi:

File:

Submit

**Kuis Form Modal**

Judul:

Tgl Mulai:    Tgl Selesai:

Deskripsi:

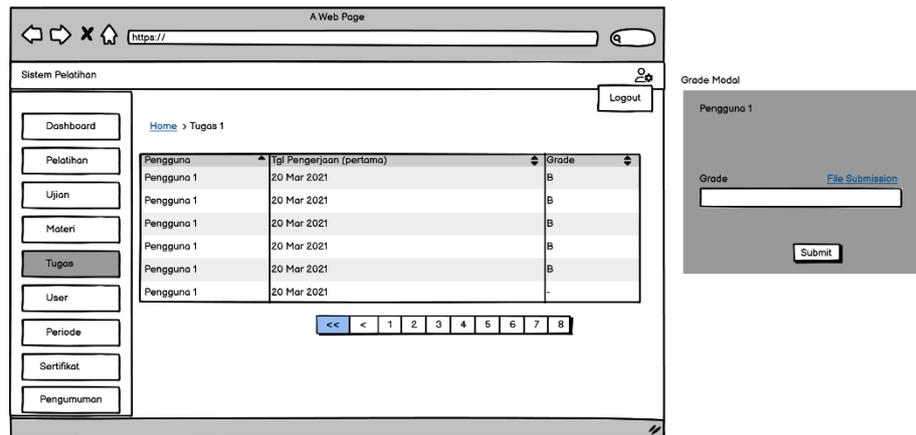
Pertanyaan

\*Pilihan 1  
Pilihan 2  
Pilihan 3

Submit

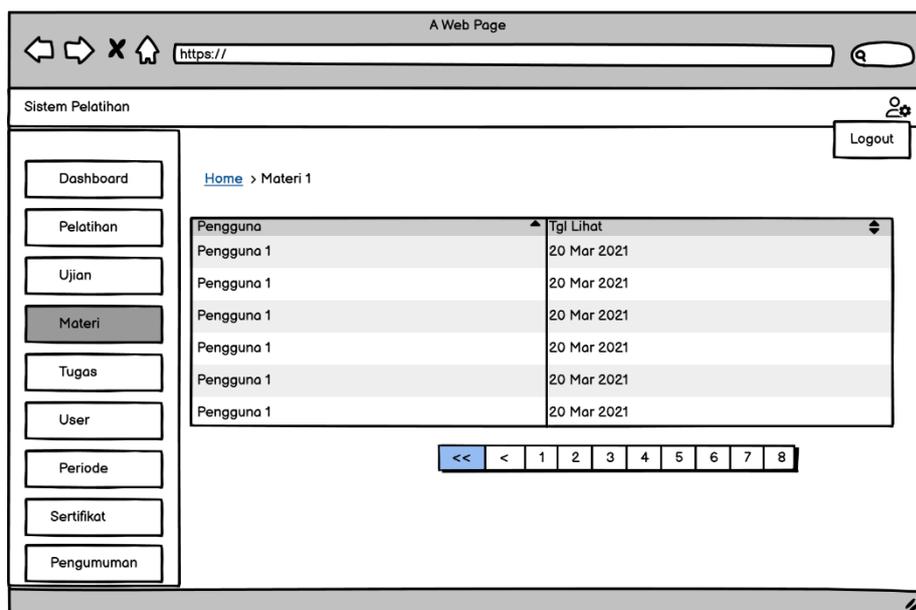
Gambar 60 Halaman Detail Materi tugas

Pada gambar 60 merupakan halaman daftar materi, tugas, dan ujian yang terkait dengan modul. Halaman muncul jika admin memilih salah satu modul dari daftar modul di halaman pelatihan. Di halaman ini admin dapat mengelola materi, tugas, dan ujian.



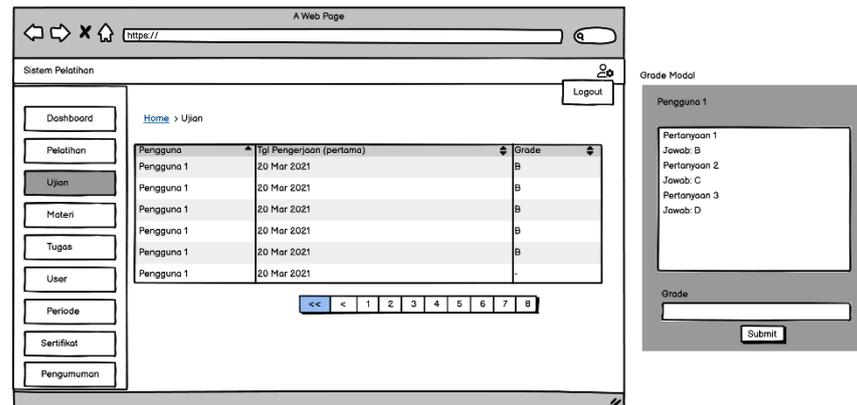
Gambar 61 Halaman Admin Penilaian

Pada gambar 61 merupakan halaman untuk admin melakukan penilaian tugas terhadap peserta pelatihan. Halaman ini berisi daftar dosen yang sudah mengerjakan tugas.



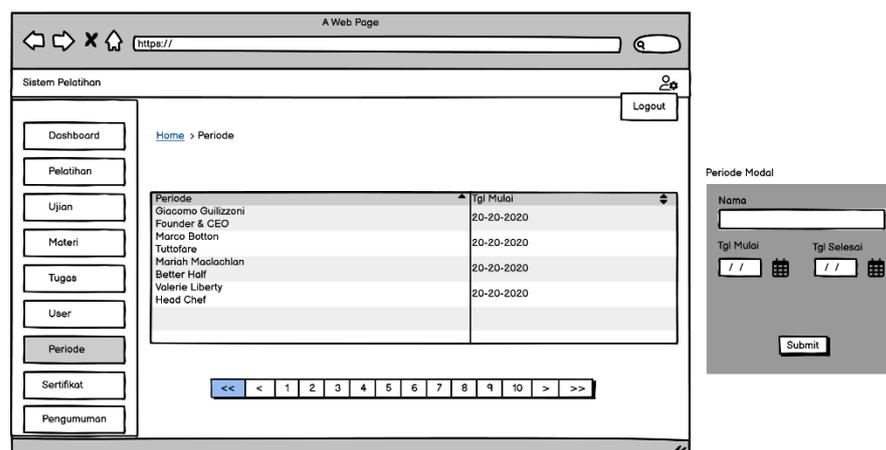
Gambar 62 Halaman Admin Daftar Materi

Pada gambar 62 merupakan halaman daftar dosen yang sudah melihat materi. Halaman ini muncul jika admin memilih navigasi materi.



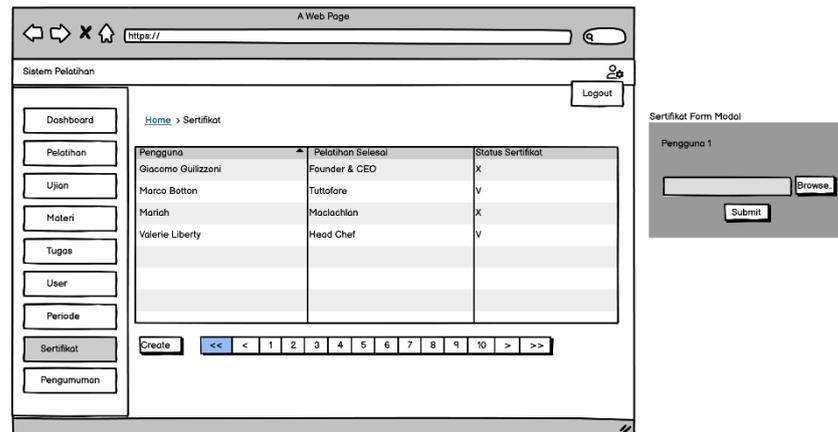
Gambar 63 Halaman Admin Penilaian Ujian

Pada gambar 63 merupakan halaman penilaian ujian, halaman ini berisi daftar peserta yang sudah mengerjakan ujian. Admin dapat melakukan penilaian dengan memilih kolom peserta.



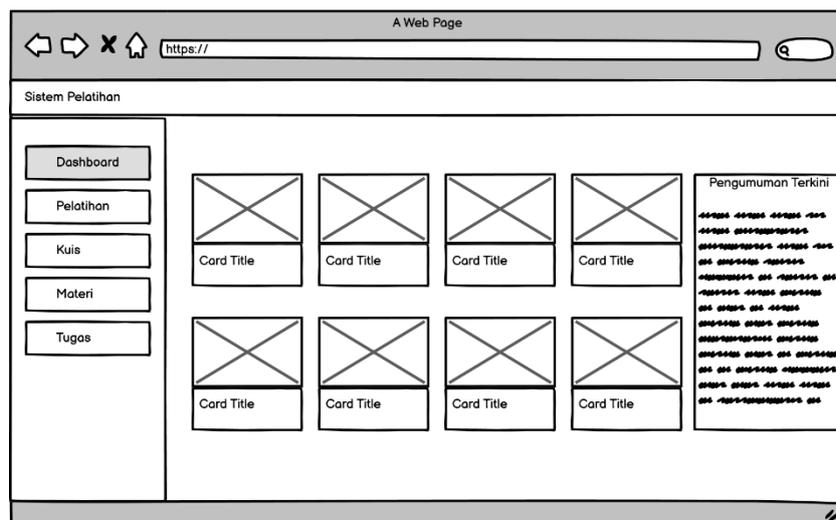
Gambar 64 Halaman Admin Periode

Pada gambar 64 merupakan halaman periode yang muncul setelah admin memilih navigasi periode. Pada halaman ini admin dapat membuat, mengedit, dan menghapus periode.



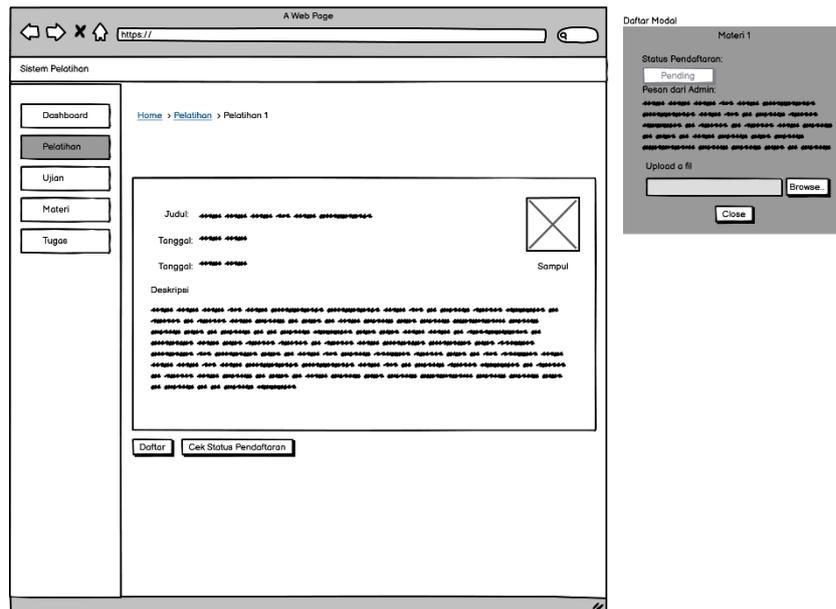
Gambar 65 Halaman Admin Sertifikat

Pada gambar 65 merupakan halaman daftar sertifikat yang muncul jika admin memilih navigasi sertifikat. Pada halaman ini admin dapat mengunggah sertifikat untuk dosen yang sudah menyelesaikan pelatihan.



Gambar 66 Halaman Dosen Dasbor

Pada gambar 66 merupakan halaman dasbor dari dosen yang berisi daftar pelatihan yang tersedia dan pengumuman. Dosen dapat memilih salah satu pelatihan untuk melihat detail pelatihan.



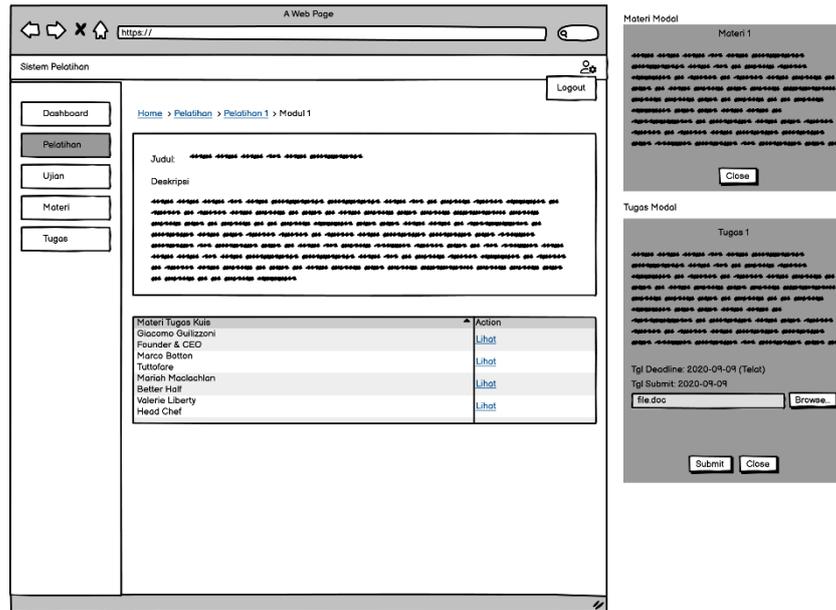
Gambar 67 Halaman Dosen Detail Pelatihan

Pada gambar 67 merupakan halaman detail pelatihan yang muncul setelah dosen memilih salah satu kartu pelatihan pada dasbor. Pada halaman ini berisi informasi mengenai pelatihan dan dosen juga dapat melakukan pendaftaran dengan tombol daftar yang akan menampilkan formulir pendaftaran.



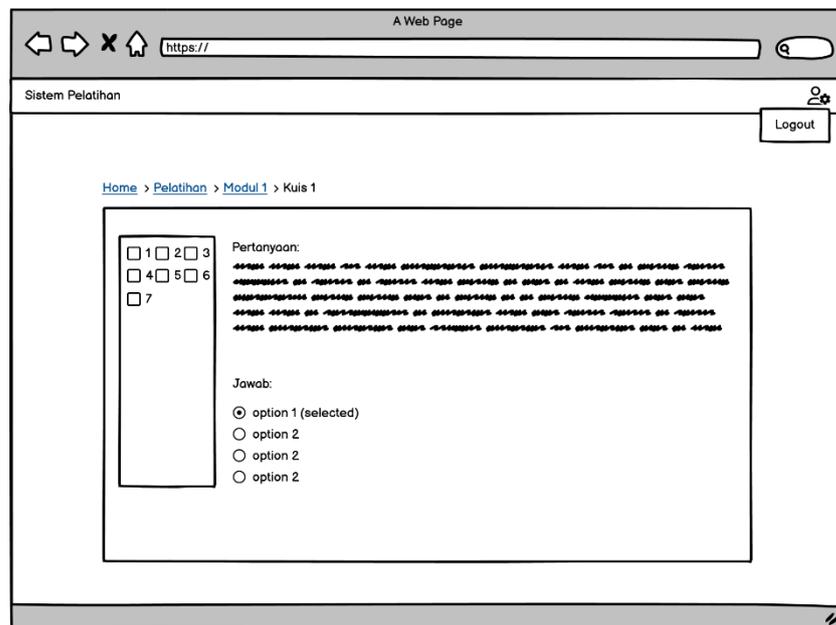
Gambar 68 Halaman Dosen Detail Pelatihan

Pada gambar 68 merupakan halaman detail pelatihan yang berisikan modul dikarenakan dosen sudah menjadi peserta pelatihan. Pada halaman ini dosen dapat memilih modul untuk melihat materi, tugas, atau ujian.



Gambar 69 Halaman Dosen Modul Pelatihan

Pada gambar 69 merupakan halaman detail dari modul yang berisikan materi, tugas, dan ujian. Dosen dapat melihat detail materi, tugas, atau ujian dan mengerjakannya.

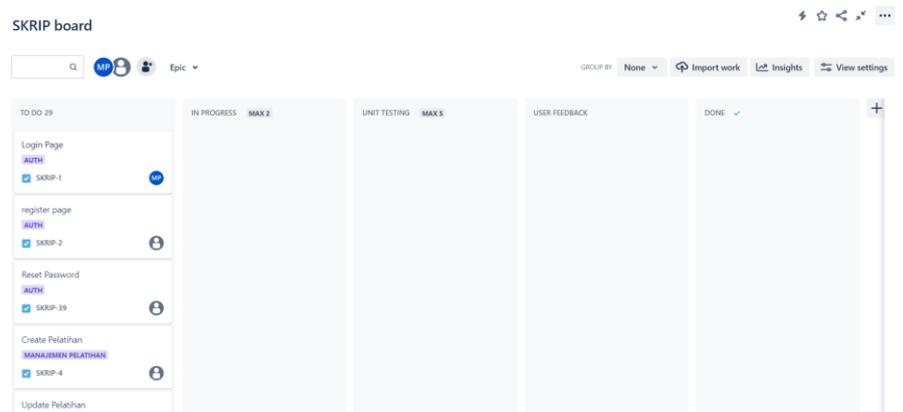


Gambar 70 Halaman Dosen Pengerjaan Ujian

Pada gambar 70 merupakan halaman untuk dosen dapat mengerjakan ujian berupa pilihan ganda, dosen juga dapat berpindah soal menggunakan navigasi nomor yang tersedia.

### 3.8 Kanban

Adapun papan kanban yang digunakan adalah sebagai berikut.



Gambar 71 Papan Kanban

Pada gambar 71 setiap kartu tugas seperti yang sedang dikerjakan harus dipindahkan ke dalam kolom "In Progress". Setelah selesai dikerjakan, kartu tersebut akan dipindahkan ke dalam kolom "Unit Testing". Jika tugas lolos uji unit, akan dipindahkan ke kolom "User Feedback" untuk mendapatkan umpan balik. Namun, jika tugas tidak lolos uji unit dan memerlukan perbaikan, kartu akan dikembalikan ke dalam kolom "Todo".

### 3.9 Pengujian Blackbox

Setelah semua fitur selesai dikerjakan dan sudah dalam status "Done" di kanban, langkah selanjutnya adalah melaksanakan uji terhadap fungsional menggunakan pengujian *blackbox* berbasis *equivalence partitioning* oleh LP3M. Adapun pengujian yang akan dilakukan adalah sebagai berikut.

Tabel 5 Daftar Pengujian

Kode Pengujian	Kasus Pengujian
Q1	Registrasi akun
Q2	Masuk akun
Q3	Lupa kata sandi
Q4	Edit akun
Q5	Logout akun
Q6	Membuat periode
Q7	Mengedit periode
Q8	Membuat pelatihan

Q9	Mengedit pelatihan
Q10	Membuat modul
Q11	Mengedit modul
Q12	Membuat materi
Q13	Mengedit materi
Q14	Menghapus materi
Q15	Membuat tugas
Q16	Mengedit tugas
Q17	Mengerjakan tugas
Q18	Menilai tugas
Q19	Membuat soal
Q20	Mengedit soal
Q21	Menghapus soal
Q22	Membuat kuis
Q23	Menambah pertanyaan
Q24	Menghapus pertanyaan
Q25	Mengerjakan kuis
Q26	Menilai kuis
Q27	Membuat pengumuman
Q28	Mengubah pengumuman
Q29	Menghapus pengumuman
Q30	Menghapus tugas
Q31	Menghapus kuis
Q32	Menghapus modul
Q33	Menghapus pelatihan
Q34	Menghapus periode

### 3.10 System Usability Scale

Setelah semua fitur selesai dikerjakan dan sudah dalam status "Done" di kanban dan menjalankan pengujian *blackbox*, langkah selanjutnya adalah melaksanakan uji terhadap responden menggunakan kuesioner *system usability scale(SUS)*.

Beberapa penelitian menyarankan bahwa sekitar 80% masalah *usability* dapat diidentifikasi dengan hanya lima peserta pertama dalam pengujian *usability*. Oleh karena itu, disarankan untuk melakukan tes dengan jumlah peserta yang sedikit, biasanya 3-5 orang, untuk menghemat waktu dan biaya.[21]

Dalam SUS ini, diperlukan 5 (lima) dosen yang akan menjadi responden yang akan mengisi formulir kepuasan yang kemudian diukur menggunakan skala Likert dan CGS dengan tugas awal adalah sebagai berikut.

1. Melakukan registrasi akun
2. Melakukan *login*/masuk
3. Melihat dan mendaftar pelatihan
4. Melihat modul dan materi
5. Melihat dan mengerjakan tugas & kuis
6. Melihat sertifikat

Tabel 6 Tabel Pernyataan SUS

No.	Pernyataan	Jawaban				
		Sangat Setuju (5)	Setuju (4)	Cukup (3)	Kurang Setuju (2)	Tidak Setuju (1)
1.	Saya berpikir akan menggunakan sistem ini lagi.					
2.	Saya merasa sistem ini rumit untuk digunakan.					
3.	Saya merasa sistem ini mudah untuk digunakan.					
4.	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini.					
5.	Saya merasa fitur-fitur sistem ini berjalan dengan semestinya					
6.	Saya merasa ada banyak hal yang tidak konsisten (tidak serasi) pada sistem ini.					
7.	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat.					
8.	Saya merasa sistem ini membingungkan.					
9.	Saya merasa tidak ada hambatan dalam menggunakan sistem ini.					

10.	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.					
-----	---	--	--	--	--	--

Berikut rumus perhitungan skor SUS dan *Sauro-Lewis curved grading scale (CGS)*:

$$SUS = 2,5 \times \left[ \sum_{n=1}^5 (U_{2n-1} - 1) + (5 - U_{2n}) \right]$$

<i>Grade</i>	<i>SUS</i>	<i>Percentiles range</i>	<i>Adjective</i>	<i>Acceptable</i>	<i>NPS</i>
A+	84.1 - 100	96 - 100	<i>Best Imaginable</i>	<i>Acceptable</i>	<i>Promoter</i>
A	80.8 - 84.0	90 - 95	<i>Excellent</i>	<i>Acceptable</i>	<i>Promoter</i>
A-	78.9 - 80.7	85 - 89	<i>Good</i>	<i>Acceptable</i>	<i>Promoter</i>
B+	77.2 - 77.8	80 - 84		<i>Acceptable</i>	<i>Passive</i>
B	74.1 - 77.1	70 - 79		<i>Acceptable</i>	<i>Passive</i>
B-	72.6 - 74.0	65 - 69		<i>Acceptable</i>	<i>Passive</i>
C+	71.1 - 72.5	60 - 64		<i>Acceptable</i>	<i>Passive</i>
C	65.0 - 71.0	41 - 59	<i>OK</i>	<i>Marginal</i>	<i>Passive</i>
C-	62.7 - 64.9	35 - 40		<i>Marginal</i>	<i>Passive</i>
D	< 51.7	15 - 34		<i>Marginal</i>	<i>Detractor</i>

Gambar 72 *Sauro-Lewis curved grading scale (CGS)*

## V. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Adapun kesimpulan adalah sebagai berikut:

1. Sistem informasi pelatihan dosen ini dibangun dengan menggunakan kerangka kerja Laravel dan melibatkan 60 task yang diselesaikan dalam kurun waktu 5 bulan. Sistem ini memiliki dua peran utama: admin dan dosen. Halaman admin mencakup fitur manajemen pelatihan, modul, materi, tugas, kuis, dan peserta, sementara halaman dosen menyediakan fitur pendaftaran pelatihan, akses modul, materi, tugas, kuis, diskusi, serta pengisian tugas dan kuis. Serta sistem ini terintegrasi dengan API publik PDDIKTI yang memungkinkan sinkronisasi data dosen secara otomatis.
2. Pengujian *blackbox* dilakukan terhadap 103 pengujian fungsional, yang seluruhnya menunjukkan hasil dengan status "valid". Selain itu, sistem ini juga mendapatkan respons positif dari lima responden dalam pengujian SUS, dengan skor rata-rata sebesar 73 dan peringkat B- pada skala Sauro-Lewis.

## 5.2 Saran

Adapun saran adalah sebagai berikut:

1. Melakukan penambahan opsi *login* menggunakan SSO Unila bagi dosen Universitas Lampung dan penambahan fitur pengaturan *enviroment* untuk mempermudah admin mengatur email, nama, logo, dan tema pada web.
2. Untuk meningkatkan pengalaman pengguna, diperlukan penyederhanaan desain dan navigasi antarmuka agar lebih intuitif dan mudah digunakan. Selain itu, perbaiki dokumentasi dan panduan bantuan dengan menyediakan panduan yang jelas dan mudah diakses, sehingga pengguna dapat dengan cepat menemukan informasi yang mereka butuhkan dan memanfaatkan fitur dengan lebih efektif.

## DAFTAR PUSTAKA

- [1] L. Unila, “Pelatihan PEKERTI.” Diakses: 25 Januari 2024. [Daring]. Tersedia pada: <https://lp3m.unila.ac.id/pelatihan-pekerti/>
- [2] LP3M UNILA, “Pelatihan Applied Approach (AA) – LP3M UNILA.” [Daring]. Tersedia pada: <https://lp3m.unila.ac.id/pelatihan-applied-approach-aa/>
- [3] LP3M UNILA, “Pelatihan PEKERTI – LP3M UNILA.” [Daring]. Tersedia pada: <https://lp3m.unila.ac.id/pelatihan-pekerti/>
- [4] E. Y. Anggraeni, E. Risanto, Y. Basuki, D. Nofianto, A. A. C, dan A. Offset, *Pengantar Sistem Informasi*. Penerbit Andi. [Daring]. Tersedia pada: <https://books.google.co.id/books?id=8VNLDwAAQBAJ>
- [5] N. A. Azwani, “Pengertian Dan Penerapan Sistem Informasi, Organisasi Dan Strategi,” Center for Open Science, 2022. doi: DOI: 10.31219/osf.io/a3569.
- [6] R. Destriana dkk., *Diagram UML Dalam Membuat Aplikasi Android Firebase" Studi Kasus Aplikasi Bank Sampah"*. Deepublish, 2021.
- [7] A. Christian, S. Hesinto, dan A. Agustina, “Rancang Bangun Website Sekolah Dengan Menggunakan Framework Bootstrap (Studi Kasus SMP Negeri 6 Prabumulih),” *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 7, no. 1, hlm. 22–27, 2018.
- [8] Z. F. Azzahra dan A. D. Anggoro, “Analisis Teknik Entity-Relationship Diagram dalam Perancangan Database Sebuah Literature Review,” *INTECH (Informatika dan Teknologi)*, vol. 3, no. 1, hlm. 8–11, 2022.

- [9] R. Fahrudin dan R. Ilyasa, "Perancangan aplikasi" nugas" menggunakan metode design thinking dan agile development," *Jurnal Ilmiah Teknologi Infomasi Terapan*, vol. 8, no. 1, hlm. 35–44, 2021.
- [10] M. E. Vermaat, S. L. Sebok, S. M. Freund, J. T. Campbell, dan M. Frydenberg, *Discovering Computers ©2018: Digital Technology, Data, and Devices*. Cengage Learning, 2017. [Daring]. Tersedia pada: <https://books.google.co.id/books?id=0NRBDgAAQBAJ>
- [11] I. Rochmawati, "Analisis user interface situs web iwearup. com," *Com. Visualita*, vol. 7, no. 2, hlm. 31–44, 2019.
- [12] K. Tatroe dan P. MacIntyre, *Programming PHP: Creating Dynamic Web Pages*. O'Reilly Media, 2020. [Daring]. Tersedia pada: <https://books.google.co.id/books?id=kEjWDwAAQBAJ>
- [13] B. Firma Sahrul, M. A. Safi'ie, O. D. Wa, dan others, "Implementasi Sistem Informasi Akademik Berbasis Web Menggunakan Framework Laravel," *TRANSFORMASI*, vol. 12, no. 1, 2016.
- [14] S. Juba, A. Vannahme, dan A. Volkov, *Learning PostgreSQL*. Packt Publishing, 2015.
- [15] A. Stellman dan J. Greene, *Learning Agile: Understanding Scrum, XP, Lean, and Kanban*. O'Reilly, 2014. [Daring]. Tersedia pada: <https://books.google.co.id/books?id=T7BhngEACAAJ>
- [16] H. Alaidaros, M. Omar, dan R. Romli, "The state of the art of agile kanban method: challenges and opportunities," *Independent Journal of Management & Production*, 2021, [Daring]. Tersedia pada: <https://api.semanticscholar.org/CorpusID:244804386>
- [17] A. G. Purwanto, R. Y. Wijaya, T. Timotius, dan I. B. Trisno, "Website System Design Using Agile Kanban Based on QR Code," *Jurnal Informatika dan Sains*, vol. 5, no. 1, hlm. 19–27, 2022, doi: 10.31326/jisa.v5i1.1066.
- [18] B. Hambling, P. Van Goathem, dan P. van Goethem, *User Acceptance Testing: A Step-by-step Guide*. dalam G - Reference, Information and Interdisciplinary Subjects Series. BCS, 2013. [Daring]. Tersedia pada: <https://books.google.co.id/books?id=NzDkkgEACAAJ>

- [19] A. Verma, A. Khatana, dan S. Chaudhary, “A comparative study of black box testing and white box testing,” *International Journal of Computer Sciences and Engineering*, vol. 5, no. 12, hlm. 301–304, 2017.
- [20] C. M. Barnum, *Usability Testing Essentials: Ready, Set ...Test!* Elsevier Science, 2020. [Daring]. Tersedia pada: [https://books.google.co.id/books?id=L6\\_SDwAAQBAJ](https://books.google.co.id/books?id=L6_SDwAAQBAJ)
- [21] S. K. M. T. Dr. Tenia Wahyuningrum, *Buku Referensi Mengukur Usability Perangkat Lunak*. Deepublish, 2021. [Daring]. Tersedia pada: <https://books.google.co.id/books?id=Pzk9EAAAQBAJ>
- [22] J. Brooke, “1,” *J Usability Stud*, vol. 8, hlm. 29–40, Jan 2013.
- [23] Z. Sharfina dan H. B. Santoso, “An Indonesian adaptation of the System Usability Scale (SUS),” dalam *2016 International Conference on Advanced Computer Science and Information Systems (ICACISIS)*, 2016, hlm. 145–148. doi: 10.1109/ICACISIS.2016.7872776.
- [24] J. R. Lewis, “The System Usability Scale: Past, Present, and Future,” *Int J Hum Comput Interact*, vol. 34, no. 7, hlm. 577–590, Jul 2018, doi: 10.1080/10447318.2018.1455307.
- [25] S. A. A. Al Rahman, D. S. Rusdianto, dan M. T. Ananta, “Pengembangan Aplikasi Sistem Pelatihan PEKERTI-AA bagi Dosen berbasis Website di Universitas Brawijaya Malang,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 7, no. 4, hlm. 2020–2028, 2023.
- [26] B. Firma Sahrul, M. A. Safi’ie, dan O. D. Wa, “Implementasi Sistem Informasi Akademik Berbasis Web Menggunakan Framework Laravel,” *TRANSFORMASI*, vol. 12, no. 1, 2016.
- [27] T. S. Waruwu, “Implementasi postgresQL sebagai sistem manajemen basis data pada pendaftaran mahasiswa baru berbasis web,” *Jurnal Mahajana Informasi*, vol. 4, no. 1, hlm. 57–61, 2019.
- [28] B. Tujni dan F. Syakti, “Implementasi Sistem Usability Scale Dalam Evaluasi Perspektif Pengguna Terhadap Sistem Informasi Akademik Berbasis Mobile,” *ILKOM Jurnal Ilmiah*, vol. 11, no. 3, hlm. 241–251, 2019.

- [29] F. Luthfi, “Penggunaan Framework Laravel Dalam Rancang Bangun Modul Back-End Artikel Website Bisnisbisnis. ID,” *JISKA (Jurnal Inform. Sunan Kalijaga)*, vol. 2, no. 1, hlm. 34, 2017.
- [30] T. Setiawan, A. Suryoprato, M. F. Yudha, dan M. I. Maulana, “Rancang Bangun Sistem Informasi Media Pembelajaran Interaktif Bahasa Inggris Berbasis Web,” *Jurnal Accounting Information System (AIMS)*, vol. 5, no. 2, hlm. 186–196, 2022.
- [31] M. Fitriawati dan R. H. Lestari, “Design of the Information System for Kindergarten Learning Evaluation used Kanban Methodology,” *IOP Conf Ser Mater Sci Eng*, vol. 662, no. 2, hlm. 022025, 2019, doi: 10.1088/1757-899X/662/2/022025.