

**RANCANG BANGUN APLIKASI PENGELOLAAN AKTIVITAS  
PERKULIAHAN TATAP MUKA BERBASIS *MOBILE* MENGGUNAKAN  
*FRAMEWORK FLUTTER***

(Skripsi)

Oleh

**KHALID SURYA GUSTI**

**NPM 2015061045**



**JURUSAN TEKNIK ELEKTRO**

**FAKULTAS TEKNIK**

**UNIVERSITAS LAMPUNG**

**2023**

**RANCANG BANGUN APLIKASI PENGELOLAAN AKTIVITAS  
PERKULIAHAN TATAP MUKA BERBASIS *MOBILE* MENGGUNAKAN  
*FRAMEWORK* FLUTTER**

**Oleh**

**KHALID SURYA GUSTI**

**NPM 2015061045**

**Skripsi**

**Sebagai Salah Satu Syarat untuk Mencapai Gelar**

**SARJANA TEKNIK**

**Pada**

**Jurusan Teknik Elektro**

**Fakultas Teknik Universitas Lampung**



**JURUSAN TEKNIK ELEKTRO**

**FAKULTAS TEKNIK**

**UNIVERSITAS LAMPUNG**

**2023**

## ABSTRAK

### RANCANG BANGUN APLIKASI PENGELOLAAN AKTIVITAS PERKULIAHAN TATAP MUKA BERBASIS *MOBILE* MENGGUNAKAN *FRAMEWORK FLUTTER*

Oleh

**Khalid Surya Gusti**

Kompleksitas fitur dan fungsionalitas pada sistem akademik unila (SIKADU) menyebabkan proses penggunaannya menjadi kompleks dan mengalihkan fokus dosen dari proses perkuliahan. Dosen sering kali menghadapi kesulitan dalam mengelola hasil dan laporan penilaian mahasiswa pada perkuliahan tatap muka. Oleh sebab itu, ditemukan kebutuhan untuk mengembangkan sistem informasi terpisah yang berfokus pada proses perkuliahan tatap muka agar aktivitas perkuliahan dapat tercatat dengan baik. Aplikasi ini dikembangkan menggunakan *framework flutter* dengan menggunakan metode RAD, yang memungkinkan keterlibatan langsung pengguna dalam proses pengembangan aplikasi. Sehingga, pengguna dapat memberikan saran secara langsung selama tahap pengembangan dan aplikasi yang dibuat dapat sesuai dengan kebutuhan dan harapan pengguna. Hasil penelitian ini adalah berhasil dikembangkannya sebuah aplikasi pengelolaan aktivitas perkuliahan tatap muka berbasis *mobile* yang memiliki fitur *login*, modul, absensi, penilaian keaktifan, kuis, tugas, UAS, UTS dan laporan penilaian. Penelitian ini menggunakan *blackbox testing* sebanyak dua kali serta mendapatkan hasil bahwa aplikasi telah sesuai dengan yang diharapkan. Selain itu, dilakukan pengujian SUS sebanyak dua kali dan didapatkan hasil bahwa pada pengujian pertama, perhitungan nilai mencapai skor 70. Skor ini masuk dalam kategori *Acceptable* pada penilaian *Acceptability Range*, kategori OK pada penilaian *Adjective Range*, dan memperoleh predikat C pada penilaian *Grade Scale*. Selanjutnya pada pengujian kedua, nilai SUS meningkat menjadi 81.87, yang masuk dalam kategori *Acceptable* pada penilaian *Acceptability Range*, naik ke kategori *good* pada penilaian *Adjective Range*, dan memperoleh predikat B pada penilaian *Grade Scale*.

Kata kunci : Siakadu, *Mobile*, Flutter, RAD

## **ABSTRACT**

### **DESIGN AND DEVELOPMENT OF APPLICATION FOR ACTIVITY MANAGEMENT IN FACE-TO-FACE LECTURE USING FLUTTER FRAMEWORK**

**By**

**Khalid Surya Gusti**

*The complexity of the features and functionality of the Unila academic system (SIKADU) causes the process to become complicated and diverts focus from the lecture process. Lecturers often face difficulties in managing student results and assessment reports during face-to-face lectures. Therefore, a need was found to develop a separate information system that focuses on the face-to-face lecture process so that lecture activities can be recorded properly. This application was developed using the Flutter framework, using the RAD method which allows direct user involvement in the application development process. So, users can provide suggestions directly during the development stage, so that the application created can meet the user's needs and expectations. The result of this research is the successful development of a mobile-based face-to-face lecture activity management application which has login features, modules, attendance, activity assessment, quizzes, assignments, UAS, UTS and assessment reports. This research used black box testing twice and obtained results that the application was as expected. Apart from that, the SUS test was carried out twice and the result was that in the first test, the calculated score reached a score of 70. This score was included in the Acceptable category in the Acceptability Range assessment, OK category in the Adjective Range assessment, and received a C predicate in the Grade Scale assessment. Furthermore, in the second test, the SUS score increased to 81.87, which was included in the Acceptable category in the Acceptability Range assessment, rose to the good category in the Adjective Range assessment, and obtained a B predicate in the Grade Scale assessment.*

**Keyword** : Siakadu, Mobile, Flutter, RAD

Judul Skripsi : **RANCANG BANGUN APLIKASI  
PENGELOLAAN AKTIVITAS  
PERKULIAHAN TATAP MUKA BERBASIS  
MOBILE MENGGUNAKAN FRAMEWORK  
FLUTTER**

Nama Mahasiswa : **Khalid Surya Gusti**

Nomor Pokok Mahasiswa : 2015061045

Program Studi : Teknik Informatika

Fakultas : Teknik



**MENYETUJUI**  
1. Komisi Pembimbing

Pembimbing Utama

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

Pembimbing Pendamping

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

2. Mengetahui

Ketua Jurusan Teknik Elektro

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

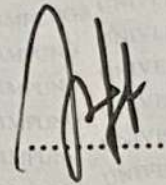
Ketua Program Studi Teknik  
Informatika

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

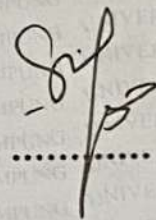
**MENGESAHKAN**

1. Tim Penguji

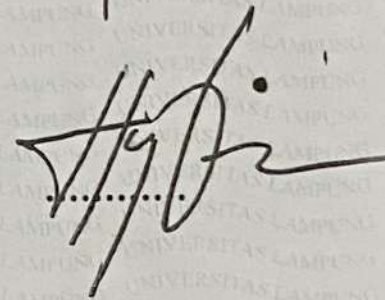
Ketua : **Yessi Mulyani, S.T., M.T.**



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



Penguji : **Ir. Ing. Hery Dian Septama, S.T.**



2. Dekan Fakultas Teknik



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

NIP. 19750928 200112 1 002



Tanggal Lulus Ujian Skripsi : **12 Desember 2023**

## SURAT PERNYATAAN

Saya yang bertandatangan di bawah ini , menyatakan bahwa skripsi saya dengan judul “Rancang Bangun Aplikasi Pengelolaan Aktivitas Perkuliahan Tatap Muka Berbasis Mobile Menggunakan *Framework* Flutter” dibuat oleh saya sendiri. Semua hasil yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti bahwa skripsi ini merupakan salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan hukum atau akademik yang berlaku.

Bandar Lampung, 19 Desember 2023  
Pembuat pernyataan,



Khalid Surya Gusti  
NPM 2015061043

## RIWAYAT HIDUP



Penulis dilahirkan di Jambi, pada tanggal 9 Juni 2001. Penulis merupakan anak ketiga dari tiga bersaudara dari pasangan Bapak Gusparman dan Ibu Kartini

Penulis menyelesaikan pendidikannya di SDN 76 Muaro Jambi pada tahun 2013, SMP Islam Al-Falah Jambi pada tahun 2016, dan SMA Negeri 4 Kota Jambi pada tahun 2019. Pada tahun 2020 penulis terdaftar sebagai mahasiswa Program Studi Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik Universitas Lampung melalui jalur SBMPTN. Selama menjadi mahasiswa, penulis melakukan beberapa kegiatan, antara lain:

1. Menjadi anggota biasa Himpunan Mahasiswa Teknik Elektro Universitas Lampung, Departemen Sosial dan Wirausaha Divisi Kewirausahaan pada tahun 2021/2022.
2. Mengikuti program Studi Independen Kampus Merdeka dari Kementerian Pendidikan dan Budaya dengan mengambil kelas *UI/UX Researcher & Designer* di Binar Academy pada tahun 2022.
3. Mengikuti program Studi Independen Kampus Merdeka dari Kementerian Pendidikan dan Budaya dengan mengambil kelas Pengembang *Front-End Web* dan *Back-End Developer* di Dicoding Indonesia pada tahun 2023.
4. Melaksanakan Kuliah Kerja Nyata di Desa Kesumadadi, Kecamatan Bekri, Kabupaten Lampung Tengah, Provinsi Lampung pada bulan Januari 2023.
5. Melakukan Kerja Praktik di PT Ardilia Nugraha Safetya pada bulan Januari sampai Februari tahun 2023 dengan membuat Website Company Profile PT. Ardilia Nugraha Safetya.
6. Menjadi Asisten Laboratorium Teknik Komputer Jurusan Teknik Elektro Universitas Lampung pada tahun 2023.



## **MOTTO**

“Apabila semangatmu goyah, lihatlah kembali ke belakang dan ingatlah tentang alasanmu memulai”

**(Penulis)**

“Sesungguhnya bersama kesulitan ada kemudahan. Maka apabila kamu telah selesai (dari sesuatu urusan), tetaplah bekerja keras (untuk urusan yang lain). Dan hanya kepada Tuhanmulah kamu berharap”

**(Q.S Al-Insyirah : 6-8)**

“bila kau tak tahan lelahnya belajar maka kamu harus tahan lelahnya kebodohan”

**(Imam Syafi'i)**

*“if you dont believe you are the best, then you will never achieve all that you are capable of”*

**(Cristiano Ronaldo)**

“Keberhasilan adalah kemampuan untuk melewati dan mengatasi dari satu kegagalan ke kegagalan berikutnya tanpa kehilangan semangat”**(Anonymous)**

## SANWACANA

Puji syukur kehadirat Allah SWT yang telah memberikan rahmat dan hidayat-Nya sehingga penulis dapat menyelesaikan penelitian yang berjudul “Rancang Bangun Aplikasi Pengelolaan Aktivitas Perkuliahan Tatap Muka Berbasis Mobile Menggunakan *Framework* Flutter”. Selama pelaksanaan penelitian ini penulis menerima banyak dukungan, bimbingan serta bantuan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Kedua orang tua tercinta dan seluruh keluarga yang selalu memberikan doa, motivasi dan kasih sayang tiada terkira yang selalu mengingatkan penulis untuk menyelesaikan penelitian ini;
2. Bapak Dr. Eng. Helmy Fitriawan, S.T., M.Sc., selaku Dekan Fakultas Teknik Universitas Lampung;
3. Ibu Herlinawati, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Universitas Lampung;
4. Ibu Yessi Mulyani, S.T.,M.T. selaku Ketua Program Studi Teknik Informatika Universitas Lampung dan Pembimbing Utama yang telah meluangkan waktunya untuk memberikan bimbingan dan membantu penulis dalam menyelesaikan penelitian ini;
5. Ibu Ir. Trisya Septiana, S.T., M.T., IPM., selaku Pembimbing Pendamping yang telah membimbing dan memberikan dukungan dalam menyelesaikan penelitian ini;
6. Bapak Ing, Hery Dian Septama, S.T. selaku Penguji yang telah memberikan banyak saran dan masukan terhadap penelitian ini;
7. Bapak Ir. Gigih Forda Nama, S.T., M.T.I. selaku Pembimbing Akademik yang selalu memberikan dukungan serta bimbingan agar menjadi lebih baik;

8. Mbak Rika selaku *Admin* Program Studi Teknik Informatika yang telah banyak membantu penulis dalam segala urusan *administrasi* selama perkuliahan dan penelitian;
9. Seluruh dosen dan staf Jurusan Teknik Informatika Unila yang memberi masukan dan mempermudah proses penelitian ini.
10. Levian Dandra, Nourma Layyina Putri, Ni Putu Tiara, dan segenap teman-teman Tim Riset Aplikasi Perkuliahan yang telah membantu penulis selama pembuatan aplikasi.
11. Addam RQH, Fathan Nur Sidiq, Nyayu Maharani, Andronikus Aditya, Muhammad Rizky Rifaldi dan seluruh teman-teman PSTI Angkatan 2020 telah mewarnai masa perkuliahan penulis dan menulis banyak cerita bersama.
12. Bagus Widiatmoko, Fajar Laksana, Nessa Aqila Azra Yasmin, Kwinny Intan Filya, Septario Ferintarozzi dan segenap teman-teman PSTI B 2020 yang telah menjadi rumah bagi penulis selama masa perkuliahan. Menemani dan membantu penulis dikalah susah dan sedih;
13. Keluarga besar Teknik Elektro Angkatan 2020 yang telah menjadi teman seperjuangan sejak mahasiswa baru. Terima kasih telah mewarnai masa perkuliahan penulis dan menulis banyak cerita bersama;

Penulis berharap agar laporan ini dapat menjadi referensi bagi pengembangan keilmuan di bidang teknik informatika. Oleh karena itu, semoga penelitian ini bermanfaat bagi yang membacanya.

Bandar Lampung, 19 Desember 2023  
Penulis,

Khalid Surya Gusti

## DAFTAR ISI

	Halaman
<b>DAFTAR ISI</b> .....	<b>ix</b>
<b>DAFTAR TABEL</b> .....	<b>xiii</b>
<b>DAFTAR GAMBAR</b> .....	<b>xiv</b>
<b>I. PENDAHULUAN</b> .....	<b>17</b>
1.1 Latar Belakang.....	17
1.2 Perumusan Masalah.....	19
1.3 Tujuan Penelitian.....	19
1.4 Manfaat Penelitian.....	19
1.5 Batasan Masalah.....	20
1.6 Sistematika Penulisan Tugas Akhir.....	20
<b>II. TINJAUAN PUSTAKA</b> .....	<b>22</b>
2.1 SIAKADU .....	22
2.2 Android.....	22
2.3 <i>Unified Modelling Language (UML)</i> .....	23
2.4 <i>User Flow</i> .....	25
2.5 Dart.....	26
2.6 Flutter .....	26
2.7 <i>Rapid Application Development (RAD)</i> .....	27
2.8 <i>Blackbox Testing</i> .....	30
2.9 <i>System Usability Scale (SUS)</i> .....	30
2.10 Penelitian Terkait.....	32
<b>III. METODOLOGI PENELITIAN</b> .....	<b>35</b>
3.1 Waktu dan Tempat Penelitian.....	35
3.2 Alat dan Bahan Penelitian.....	35
3.3 Struktur Tim <i>Capstone</i> .....	36
3.4 Tahapan Penelitian .....	37
3.4.1 <i>Requirements Planning</i> .....	38
3.4.1.1 Studi Literatur .....	38
3.4.1.2 Wawancara .....	38
3.4.1.3 Kebutuhan Fungsional dan Non Fungsional .....	40
3.4.2 <i>User Design</i> .....	42
3.4.2.1 <i>Use Case Diagram</i> .....	42
3.4.2.2 <i>Activity Diagram</i> .....	45
3.4.2.3 <i>User flow</i> .....	59
3.4.2.4 <i>Mockup Aplikasi</i> .....	61
3.4.3 <i>Construction</i> .....	72

3.4.4 Cutover.....	74
<b>IV. HASIL DAN PEMBAHASAN .....</b>	<b>75</b>
4.1 Iterasi 1 .....	75
4.2 Iterasi 2 .....	112
<b>V. KESIMPULAN DAN SARAN .....</b>	<b>130</b>
5.1 Kesimpulan.....	130
5.2 Saran .....	130
<b>DAFTAR PUSTAKA .....</b>	<b>132</b>

## DAFTAR TABEL

	Hal.
Tabel 2.1 Kelebihan dan kekurangan RAD .....	29
Tabel 3.1 Jadwal Penelitian.....	35
Tabel 3.2 Alat ( <i>Hardware</i> dan <i>Software</i> ) pengembangan sistem .....	35
Tabel 3.3 Jumlah Informan .....	35
Tabel 3.4 Kebutuhan Fungsional .....	40
Tabel 3.5 Kebutuhan Non Fungsional .....	42
Tabel 3.6 <i>Use case diagram</i> .....	44
Tabel 4.1 Tahapan Pengujian Halaman <i>Login</i> .....	105
Tabel 4.2 Tahapan Pengujian Halaman Utama.....	105
Tabel 4.3 Tahapan Pengujian Halaman Pertemuan .....	106
Tabel 4.4 Tahapan Pengujian Modul Pembelajaran .....	106
Tabel 4.5 Tahapan Pengujian Halaman Absensi.....	102
Tabel 4.6 Tahapan Pengujian Halaman Keaktifan.....	107
Tabel 4.7 Tahapan Pengujian Halaman Kuis.....	108
Tabel 4.8 Tahapan Pengujian Halaman Tugas.....	108
Tabel 4.9 Tahapan Pengujian Halaman Laporan .....	108
Tabel 4.10 Tahapan Pengujian <i>Sign Out</i> .....	108
Tabel 4.11 Hasil Pengujian SUS 1 .....	109
Tabel 4.12 Hasil Perhitungan SUS 1 .....	110
Tabel 4.13 Saran dan Masukan .....	111
Tabel 4.14 Tahapan Pengujian Halaman UTS.....	126
Tabel 4.15 Tahapan Pengujian Halaman UAS .....	126
Tabel 4.16 Tahapan Pengujian Halaman Keaktifan.....	127
Tabel 4.17 Tahapan Pengujian SUS 2.....	128
Tabel 4.18 Tahapan Perhitungan SUS 2 .....	128
Tabel 4.19 Tahapan Rincian Pengujian 1 & 2 .....	129

## DAFTAR GAMBAR

	Hal.
Gambar 2.1 Halaman SIAKADU .....	24
Gambar 2.2 Logo Android .....	24
Gambar 2.3 Komponen <i>Use Case Diagram</i> .....	24
Gambar 2.4 Komponen <i>Activity Diagram</i> .....	25
Gambar 2.5 Komponen <i>User Flow</i> .....	25
Gambar 2.6 Tahapan Rapid Application Development .....	28
Gambar 2.7 Item Pertanyaan SUS .....	19
Gambar 2.8 Contoh Jawaban SUS .....	20
Gambar 3.1 Struktur Tim <i>Capstone</i> .....	36
Gambar 3.2 Tahapan Penelitian .....	37
Gambar 3.3 <i>Use case Diagram</i> .....	43
Gambar 3.4 <i>Activity Diagram</i> Masuk Aplikasi.....	45
Gambar 3.5 <i>Activity Diagram</i> Memilih Daftar Pertemuan .....	45
Gambar 3.4 <i>Activity Diagram</i> Memilih Penilaian Mahasiswa .....	45
Gambar 3.5 <i>Activity Diagram</i> Mengakses Modul .....	48
Gambar 3.6 <i>Activity Diagram</i> Penilaian Keaktifan .....	49
Gambar 3.7 <i>Activity Diagram</i> Melakukan Absensi.....	50
Gambar 3.8 <i>Activity Diagram</i> Penilaian Kuis .....	52
Gambar 3.9 <i>Activity Diagram</i> Penilaian Tugas .....	53
Gambar 3.10 <i>Activity Diagram</i> Penilaian UTS.....	55
Gambar 3.11 <i>Activity Diagram</i> Penilaian UAS .....	56
Gambar 3.12 <i>Activity Diagram</i> Mengunduh Laporan .....	58
Gambar 3.13 <i>Activity Diagram</i> Keluar Aplikasi.....	59
Gambar 3.14 <i>User Flow</i> .....	60
Gambar 3.15 <i>Mockup</i> Halaman <i>Login</i> .....	61
Gambar 3.16 <i>Mockup</i> Halaman Utama .....	62
Gambar 3.17 <i>Mockup</i> Halaman Pertemuan .....	63
Gambar 3.18 <i>Mockup</i> Halaman Detail Pertemuan.....	64
Gambar 3.19 <i>Mockup</i> Halaman Modul .....	65
Gambar 3.20 <i>Mockup</i> Halaman Penilaian Mahasiswa.....	65
Gambar 3.21 <i>Mockup</i> Halaman Absensi.....	66
Gambar 3.22 <i>Mockup</i> Halaman Keaktifan.....	67
Gambar 3.23 <i>Mockup</i> Halaman Kuis .....	68
Gambar 3.24 <i>Mockup</i> Halaman Tugas.....	69
Gambar 3.25 <i>Mockup</i> Halaman UTS .....	70

Gambar 3.26	<i>Mockup</i> Halaman UAS.....	71
Gambar 3.27	<i>Mockup</i> Halaman Laporan .....	72
Gambar 4.1	<i>Source Code</i> Halaman <i>Login</i> .....	75
Gambar 4.2	Halaman <i>Login</i> .....	76
Gambar 4.3	<i>Source Code</i> Halaman Utama .....	77
Gambar 4.4	<i>Source Code Controller</i> Halaman Utama .....	78
Gambar 4.5	Halaman Utama.....	79
Gambar 4.6	<i>Source Code</i> Pertemuan .....	80
Gambar 4.7	<i>Source Code Controller</i> Pertemuan .....	81
Gambar 4.8	Halaman Pertemuan .....	82
Gambar 4.9	<i>Source Code</i> Detail Pertemuan .....	83
Gambar 4.10	Halaman Detail Pertemuan.....	84
Gambar 4.11	<i>Source Code</i> Modul.....	85
Gambar 4.12	<i>Source Code Controller</i> Modul.....	86
Gambar 4.13	Halaman Modul.....	87
Gambar 4.14	<i>Source Code</i> Penilaian Mahasiswa .....	88
Gambar 4.15	Halaman Penilaian Mahasiswa.....	89
Gambar 4.16	<i>Source Code</i> Absensi .....	90
Gambar 4.17	<i>Source Code Controller</i> Absensi .....	91
Gambar 4.18	Halaman Absensi.....	92
Gambar 4.19	<i>Source Code</i> Keaktifan .....	93
Gambar 4.20	<i>Source Code Controller</i> Keaktifan.....	94
Gambar 4.21	Halaman Keaktifan.....	95
Gambar 4.22	<i>Source Code</i> Kuis.....	96
Gambar 4.23	<i>Source Code Controller</i> Kuis.....	97
Gambar 4.24	Halaman Kuis .....	98
Gambar 4.25	<i>Source Code</i> Tugas .....	99
Gambar 4.26	<i>Source Code Controller</i> Tugas.....	100
Gambar 4.27	Halaman Tugas.....	101
Gambar 4.28	<i>Source Code</i> Laporan .....	102
Gambar 4.29	<i>Source Code Controller</i> Laporan .....	103
Gambar 4.30	Halaman Laporan .....	104
Gambar 4.31	Ketentuan Skor SUS.....	110
Gambar 4.32	<i>Source Code</i> Perbaikan Batas Nilai .....	112
Gambar 4.33	Tampilan Perbaikan Batas Nilai.....	113
Gambar 4.34	Tampilan Perbaikan Ikon .....	114
Gambar 4.35	<i>Source Code</i> Perbaikan Pengelolaan <i>State</i> .....	115
Gambar 4.36	Tampilan Perbaikan Halaman Modul .....	116
Gambar 4.37	<i>Source Code</i> Perbaikan Notifikasi .....	117



Gambar 4.38	<i>Source Code</i> Perbaikan <i>Pop up</i> .....	118
Gambar 4.39	Halaman Keaktifan.....	119
Gambar 4.40	<i>Source Code Controller UTS</i> .....	120
Gambar 4.41	<i>Source Code</i> UTS.....	121
Gambar 4.42	Halaman UTS.....	122
Gambar 4.43	<i>Source Code Controller UAS</i> .....	123
Gambar 4.44	<i>Source Code</i> UAS .....	124
Gambar 4.45	Halaman UAS .....	125
Gambar 4.46	Grafik Pengujian SUS .....	129

## I. PENDAHULUAN

### 1.1 Latar Belakang

Perkuliahan adalah interaksi belajar-mengajar antara mahasiswa dan dosen menggunakan sumber belajar di lingkungan akademik, yang melibatkan komunikasi langsung maupun tidak langsung [1]. Proses perkuliahan pada lingkungan perguruan tinggi biasanya dilakukan dengan penyampaian materi melalui tatap muka, baik berupa pelajaran ataupun ceramah [2]. Proses tersebut menciptakan hubungan yang erat antara dosen dan mahasiswa, dimana mahasiswa berperan sebagai penerima materi dan dosen bertanggung jawab memberikan materi [3].

Dalam lingkup akademik, dosen memiliki tugas yang kompleks. Selain memiliki tanggung jawab dalam memberi dan mengelola kuliah, dosen juga memiliki tugas dalam melakukan penelitian dan pengabdian kepada perguruan tinggi maupun masyarakat, sehingga dosen memerlukan manajemen yang baik agar dapat menjalankan kegiatan perkuliahan dengan baik [4]. Oleh karena itu, diperlukan suatu solusi yang dapat membantu dosen, berupa sistem informasi yang memberikan kemudahan dan kepraktisan dalam mengelola proses perkuliahan tatap muka secara *real-time* dan otomatis, sehingga di akhir perkuliahan dosen telah mendapatkan dokumentasi perkuliahan secara lengkap dan dapat memberikan penilaian kepada mahasiswa dengan baik dan akurat.

Salah satu pendukung kebutuhan dosen pada proses perkuliahan adalah melalui sistem informasi akademik berbasis internet yang disebut SIAKAD [5]. SIAKAD dibuat untuk mempermudah kegiatan administrasi akademik di perguruan tinggi, seperti penyusunan kurikulum dan jadwal kuliah, pengelolaan kartu rencana studi (KRS), distribusi materi kuliah, pengisian data nilai mahasiswa, serta mengelola data mahasiswa dan dosen [6]. Pemanfaatan SIAKAD pada sebuah perguruan tinggi berperan penting sebagai sarana pendukung dalam keberlangsungan kegiatan perkuliahan [7].

Universitas Lampung telah memiliki Sistem Informasi yang bernama Sistem Informasi Akademik Unila atau SIAKADU. SIAKADU adalah sebuah sistem informasi yang digunakan sebagai media manajemen pengelolaan perkuliahan. SIAKADU digunakan oleh dosen dan mahasiswa untuk mendapatkan informasi akademis dengan cepat dan akurat [8]. Saat ini, Universitas Lampung menggunakan sistem informasi akademik terpadu (SIAKADU) versi 6. Sistem ini mengusung konsep *user friendly* yang memberikan kemudahan bagi pengguna untuk mengakses fitur-fitur tambahan seperti jadwal mata kuliah, pengisian KRS, transkrip nilai, konsultasi perkuliahan, data perkembangan studi mahasiswa, pengumuman, serta pelaporan kepada orang tua [9].

Namun pada sistem SIAKADU, masih terdapat beberapa tantangan yang dihadapi, salah satu permasalahannya adalah kompleksitas fitur dan fungsionalitas sistem. Permasalahan tersebut dapat menyebabkan proses penggunaan menjadi terlalu kompleks dan mengalihkan fokus dari proses perkuliahan, yang seharusnya menjadi perhatian utama dosen. Dosen sering menghadapi kesulitan dalam mengelola hasil absensi dan laporan penilaian mahasiswa pada perkuliahan tatap muka di dalam ruangan kelas. Proses manual untuk mencatat kehadiran mahasiswa, menilai keaktifan dan tugas mahasiswa, serta menghasilkan laporan penilaian memerlukan banyak waktu.

Berdasarkan permasalahan tersebut, ditemukan kebutuhan untuk mengembangkan sistem informasi terpisah yang berfokus pada proses perkuliahan tatap muka agar aktivitas perkuliahan dapat tercatat dengan baik. Untuk memenuhi kebutuhan tersebut, sistem informasi ini dikembangkan menggunakan *framework* flutter. Pemilihan flutter didasarkan pada kemampuan flutter yang fleksibel sehingga memungkinkan pengembang dapat merespon cepat atas perbaikan-perbaikan dan saran dari pengguna [10]. Berdasarkan pemaparan di atas, maka diambil salah satu masalah di atas sebagai tema Skripsi dengan judul "Rancang Bangun Aplikasi Pengelolaan Aktivitas Perkuliahan Tatap Muka Berbasis *Mobile* Menggunakan *Framework* Flutter".

## 1.2 Perumusan Masalah

Berdasarkan latar belakang, kajian masalah yang mendasari penelitian ini adalah:

1. Bagaimana mengembangkan aplikasi menggunakan *framework* flutter yang dapat membantu dosen dalam mengelola proses perkuliahan tatap muka secara *realtime*?
2. Bagaimana mengembangkan aplikasi menggunakan *framework* flutter dengan fitur dan fungsionalitas yang dapat memenuhi kebutuhan dosen?

## 1.3 Tujuan Penelitian

Tujuan penelitian ini adalah:

1. Mengembangkan tahap awal (*front-end*) suatu aplikasi menggunakan *framework* flutter yang dapat membantu dosen dalam aktivitas perkuliahan yang terdigitalisasi.
2. Mengembangkan (*front-end*) aplikasi menggunakan *framework* flutter dengan fitur dan fungsionalitas yang dapat memenuhi kebutuhan dan tantangan yang dihadapi oleh dosen dalam pengelolaan aktivitas perkuliahan.

## 1.4 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan beberapa manfaat sebagai berikut:

### a. Bagi Penulis

Meningkatkan keterampilan dan pemahaman yang lebih mendalam tentang bahasa pemrograman dart dan penggunaan *framework* flutter.

### b. Bagi Perguruan Tinggi

Meningkatkan keakuratan data akademik, mengurangi kesalahan manusia, dan mempercepat respon terhadap aktivitas mengajar.

### c. Bagi Dosen

Memudahkan proses pengelolaan perkuliahan tatap muka, mendapatkan laporan perkuliahan secara *realtime*, serta memudahkan dosen dalam pemantauan perkembangan mahasiswa.

### 1.5 Batasan Masalah

Dalam penelitian ini, pembatasan masalah meliputi hal-hal sebagai berikut:

1. Pembuatan aplikasi difokuskan untuk aktivitas perkuliahan yang digunakan oleh dosen saat mengajar di dalam kelas.
2. Pembuatan *front-end* aplikasi dibuat berbasis *mobile* dengan menggunakan bahasa pemrograman dart dan menggunakan flutter sebagai kerangka kerja (*framework*).
3. Penelitian hanya difokuskan pada aspek pengguna yaitu dosen.
4. Tidak mencakup aspek teknis dan infrastruktur dalam pengembangan atau implementasi aplikasi, seperti keamanan jaringan, skalabilitas, atau arsitektur teknis.

### 1.6 Sistematika Penulisan Tugas Akhir

Sistematika penulisan skripsi akhir ini terdiri dari 5 (lima) bab sebagai berikut:

**BAB I : PENDAHULUAN**

Menjelaskan secara umum tentang latar belakang penelitian, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan penelitian serta sistematika penulisan.

**BAB II : TINJAUAN PUSTAKA**

Memaparkan teori-teori dasar yang digunakan sebagai referensi untuk memahami permasalahan terkait dalam melakukan penelitian mengenai Rancang Bangun Aplikasi Pengelolaan Aktivitas Perkuliahan Tatap Muka Berbasis *Mobile* Menggunakan *Framework* Flutter.

**BAB III : METODE PENELITIAN**

Membahas mengenai metodologi penelitian yang digunakan dalam Rancang Bangun Aplikasi Pengelolaan Aktivitas Perkuliahan Tatap Muka Berbasis *Mobile* Menggunakan *Framework* Flutter.

**BAB IV : HASIL DAN PEMBAHASAN**

Membahas mengenai pembahasan yang dihasilkan pada penelitian Rancang Bangun Aplikasi Pengelolaan Aktivitas Perkuliahan Tatap Muka Berbasis *Mobile* Menggunakan *Framework* Flutter.

**BAB V : KESIMPULAN DAN SARAN**

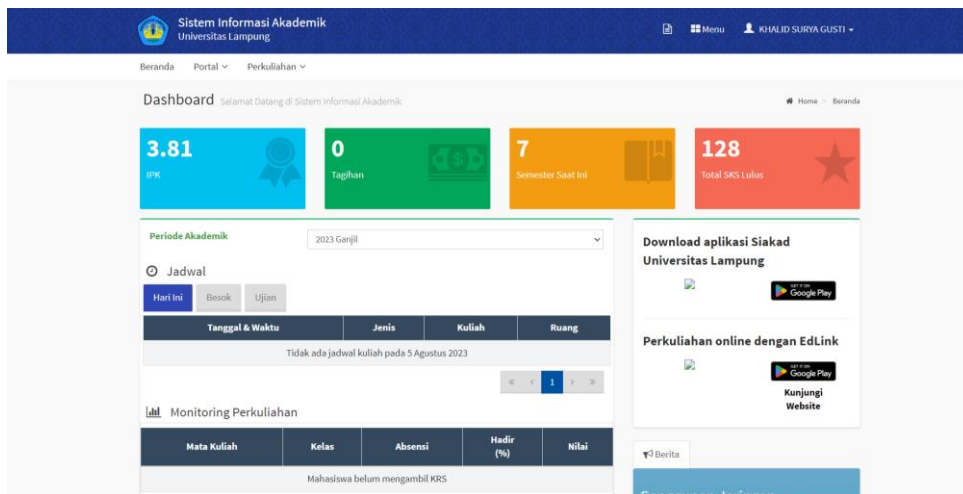
Memuat kesimpulan dari hasil penelitian yang telah dilaksanakan dan saran-saran untuk penelitian lebih lanjut.

**DAFTAR PUSTAKA**

## II. TINJAUAN PUSTAKA

### 2.1 SIAKADU

Sistem informasi akademik Universitas Lampung (SIAKADU) adalah sistem informasi yang dirancang dengan tujuan membantu pengelolaan data dalam bidang akademik dan administratif mahasiswa serta memberikan kemudahan dalam proses perkuliahan di Universitas Lampung. Pengelolaan data tersebut berupa pengisian KRS (Kartu Rencana Studi), informasi terkait penyusunan jadwal perkuliahan dan kurikulum yang digunakan, pengisian nilai mahasiswa, pencetakan kartu hasil studi, pembayaran UKT (Uang Kuliah Tunggal), dan melakukan pendaftaran wisuda. Selain itu, SIAKADU juga menyediakan fitur untuk mengajukan permohonan administrasi seperti cuti kuliah, perpanjangan masa studi, atau pengajuan kertas kerja akhir [10]. Gambar 2.1 di bawah ini menggambarkan tampilan halaman utama SIAKADU.



Gambar 2.1 Halaman Utama SIAKADU

Sumber : <https://siakadu.unila.ac.id/>

### 2.2 Android

Android merupakan sistem operasi yang banyak digunakan pada perangkat bergerak yang saat ini sangat populer digunakan pada ponsel cerdas. Android dikembangkan oleh Google untuk ponsel cerdas dan perangkat seluler lainnya, misalnya tablet. Android dapat berjalan pada berbagai macam perangkat yang dikembangkan oleh banyak vendor ponsel cerdas yang berbeda. Android

menyertakan paket pengembangan perangkat lunak untuk penulisan kode asli dan. Perakitan modul perangkat lunak dalam membuat aplikasi bagi pengembang Android. Selain menyediakan paket pengembangan aplikasi Android, Android juga menyediakan pasar untuk mendistribusikan aplikasi yang telah selesai dikembangkan [11]. Gambar 2.2 di bawah ini menggambarkan logo dari android.



Gambar 2.2 Logo Android

Sumber : *E-book* Pemrograman Aplikasi Android dengan Android Studio, Photoshop, dan Audition.

### **2.3 Unified Modelling Language (UML)**











*Unified Modeling Language (UML)* adalah bahasa pemodelan standar yang umum di bidang rekayasa perangkat lunak, yang meliputi teknik notasi grafik untuk membuat model abstrak dari suatu sistem. UML disebut juga sebagai bahasa grafis untuk mendokumentasikan, menspesifikasikan, dan membangun sistem. UML memberikan cara standar untuk membuat *blueprint* sistem, termasuk komponen konseptual seperti aktor, proses bisnis, komponen sistem dan aktivitas. Penggunaan UML pada suatu sistem bertujuan untuk memvisualisasikan, membuat, dan mendokumentasikan artefak sistem perangkat lunak yang sedang dikembangkan dalam suatu diagram. Adapun diagram tersebut diantaranya adalah *class*, *use case*, *sequence*, *deployment*, *communication*, *activity*, dan *state diagram* [12].

#### *a. Use case Diagram*

*Use case diagram* adalah diagram yang menggambarkan dan merepresntasikan aktor, *use case* dan *depedencies*. *Use case* mendefinisikan “apa” yang dilakukan oleh sistem dan elemen-elemennya, bukan “bagaimana” sistem dan elemen-



elemennya saling berinteraksi. *Use case* bekerja dengan menggunakan “*scenario*”, yaitu deskripsi urutan langkah yang menerangkan apa yang dilakukan pengguna terhadap sistem maupun sebaliknya [13]. Gambar 2.3 di bawah ini menjelaskan komponen – komponen yang ada pada *use case diagram*.

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).
3		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya ( <i>sinergi</i> ).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

Gambar 2.3 Komponen *Use Case Diagram*

Sumber : *E-book Metode Analisis dan Perancangan*

#### b. *Activity diagram*

*Activity diagram* merupakan bentuk visual dari alir kerja yang berisi aktivitas dan tindakan, yang juga dapat berisi pilihan, pengulangan, dan *concurrency*. *Activity diagram* menjelaskan behavioral/perilaku yang penting di dalam diagram UML untuk menggambarkan aspek dinamis dari sistem, yang dimulai dari pandangan *business level* hingga ke *operational level*. Pada dasarnya, *activity diagram*

merupakan variasi dari *statechart diagram*. *Activity diagram* mempunyai peran seperti halnya *flowchart*, perbedaannya adalah *activity diagram* dapat mendukung perilaku parallel sedangkan *flowchart* tidak bisa [14]. Gambar 2.4 di bawah ini menjelaskan komponen komponen yang ada pada *activity diagram*.






NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran
6		<i>Join Node</i>	Banyak aliran yang pada tahap tertentu berubah menjadi beberapa satu aliran
7		Decision	Pilihan untuk mengambil keputusan

Gambar 2.4 Komponen *Activity Diagram*

Sumber : *E-book Diagram UML Dalam Membuat Aplikasi Android Firebase*

## 2.4 User Flow

*User flow* adalah urutan langkah atau proses yang harus dilakukan oleh pengguna dalam mengerjakan suatu *task*. *User flow* dapat dikatakan sebagai proses untuk mengidentifikasi langkah yang harus dilakukan pengguna saat menggunakan suatu sistem atau aplikasi untuk mencapai suatu tujuan. Penggunaan *user flow* dapat memvisualisasikan langkah yang diambil pengguna, mengidentifikasi masalah dan kesulitan pengguna dalam penggunaan aplikasi, serta menentukan bagaimana mengoptimalkan alur interaksi untuk efisiensi dan kenyamanan dari pengguna agar aplikasi yang dibuat memiliki alur yang baik serta mudah digunakan oleh pengguna [15]. Gambar 2.5 di bawah ini menjelaskan komponen-komponen yang ada pada *user flow*.

Simbol	Deskripsi
	Mulai dan Akhir
	Proses/Kegiatan
	Kondisional dan Keputusan
	Arah aliran
	Masukan dan Keluaran

Gambar 2.5 Komponen *User Flow*

Sumber : *E-book* Kecanduan Gadget Dengan Pendekatan Lean UX

## 2.5 Dart

Dart adalah bahasa pemrograman yang dikembangkan oleh Google sejak tahun 2012 yang dipimpin oleh Lars Bak dan Kasper Lund. Bahasa Dart menggunakan prinsip pemrograman berorientasi objek (OOP) yang berfokus untuk optimalisasi *client*. Dart termasuk bahasa tingkat tinggi dengan manajemen memori otomatis (*garbage-collection*) yang membuat aplikasi berjalan lebih cepat pada banyak *platform*. Dart menggunakan sintak kode bergaya bahasa C, yang mana ketika membangun aplikasi, dart meng-*compile* ke dalam Bahasa *native* agar dapat berjalan pada *mobile* dan di *transpile* ke *Javascript* untuk dapat berjalan di *browser* [16].

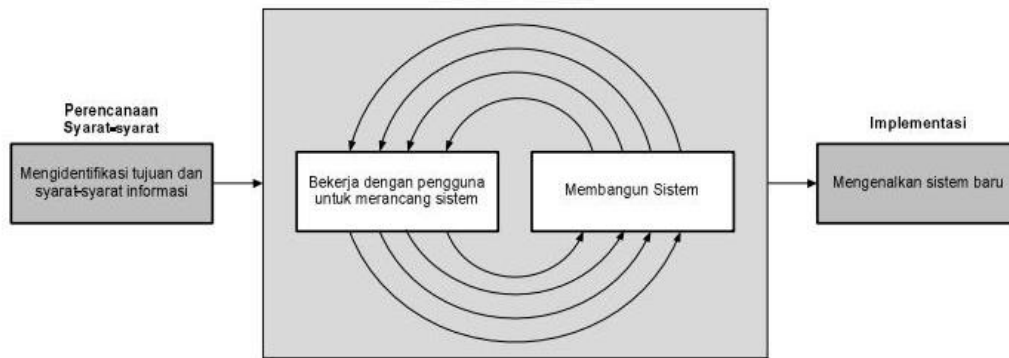
## 2.6 Flutter

Flutter adalah *framework* yang dirilis oleh Google pada tahun 2016 yang digunakan untuk mengembangkan aplikasi seluler dengan performa tinggi. Flutter dapat berjalan di iOS, Android, dan juga Fuschia yang merupakan sistem operasi terbaru dari Google. Berbeda dengan menggunakan *web views* atau mengandalkan perangkat OEM untuk *render* setiap komponen tampilan, flutter menggunakan mesin *rendering* sendiri yang memiliki performa tinggi. Hal ini memungkinkan pengembangan aplikasi dengan performa yang setara dengan aplikasi *native*.

Selain itu, flutter juga mendukung *hot reload* yang *stateful* saat dalam proses pengembangan, yang menjadikan faktor utama dalam meningkatkan siklus pengembangan. Selain itu Flutter memiliki *toolkit* tampilan yang indah, Flutter di buat berbeda dengan menggunakan desain material dari google yang *out of the box*, juga mempermudah *developer* untuk mengaplikasikan material *cupertino* untuk mendesain tampilan di *iOS*. Tampilannya dapat disesuaikan dalam semua *platform* sehingga aplikasi yang dibuat masi terkesan seperti produk sistem operasinya. Kemudian Flutter mendukung akses *libraries* berbagai bahasa pemrograman, Flutter mendukung interoperabilitas bahasa C dan C++ serta *platform channel* yang baik untuk menghubungkan ke kotlin atau java untuk android dan *swift* atau *objective-C* untuk *iOS*. Kelebihan Flutter selanjutnya ialah Flutter memiliki animasi dan transisi yang indah, *developer* juga dapat membuat *custom widget*, karena *widget* tersebut dapat disusun sehingga *developer* dapat berkereasi dengan UI sesuai keinginan.[17].

## **2.7 Rapid Application Development (RAD)**

*Rapid Application Development (RAD)* adalah sebuah model proses perkembangan *software* sekuensial linier yang menekankan siklus perkembangan yang sangat pendek serta menekankan pada kecepatan dan fleksibilitas dalam proses pengembangan perangkat lunak. Model RAD digunakan untuk proyek-proyek yang memerlukan pengembangan perangkat lunak yang cepat dan mempunyai kebutuhan yang berubah-ubah [18]. Pemilihan metode *RAD* pada penelitian ini dikarenakan *RAD* memungkinkan keterlibatan langsung pengguna (dosen) dalam proses pengembangan aplikasi. Jadi, pengguna (dosen) dapat memberikan saran serta *feedback* secara langsung selama tahap-tahap pengembangan, sehingga aplikasi yang dibuat dapat sesuai dengan kebutuhan dan harapan pengguna. Gambar 2.6 di bawah ini menggambarkan tahapan pada *Rapid Application Development (RAD)*.



Gambar 2.6 Tahapan *Rapid Application Development*

Sumber : *E-book Analisis dan Perancangan Sistem*

Metode *RAD* terdiri dari empat tahapan, yaitu :

- a. *Requirements Planning* : Tahapan awal dalam Model *RAD* adalah tahapan perencanaan kebutuhan. Tahapan ini dilakukan untuk mengidentifikasi dan memahami kebutuhan fungsional dan non-fungsional pengguna.
- b. *User Design* : Tahapan kedua adalah tahap perancangan arsitektur sistem. Tahapan ini melibatkan desain teknis tentang bagaimana komponen sistem bekerja satu sama lain dan bagaimana perangkat lunak bekerja.
- c. *Construction* : Tahapan ketiga adalah tahap konstruksi atau membangun sistem. Dalam tahap ini, pengembang perangkat lunak membangun prototipe atau model awal dari perangkat lunak. Model ini digunakan untuk mendapatkan umpan balik dari pelanggan dan memperbaiki perangkat lunak sesuai dengan umpan balik tersebut.
- d. *Cutover* : Tahapan terakhir dalam Model *RAD* adalah tahap implementasi. Tahapan ini merupakan peralihan sistem dari prototipe ke perangkat lunak yang lebih baik. Perangkat lunak yang telah selesai dibangun, selanjutnya siap untuk diimplementasikan. [19].

Menurut [20] RAD memiliki kelebihan serta kekurangan, yaitu :

Tabel 2.1 Kelebihan dan kekurangan RAD

No.	Kelebihan	Kekurangan
1.	Mudah untuk diamati karena menggunakan model prototipe, sehingga pengguna lebih mengerti tentang sistem yang dikembangkan.	Metode RAD mengharuskan pengembang dan pelanggan memiliki komitmen pada aktivitas <i>rapid-fire</i> yang dibutuhkan untuk melengkapi suatu sistem, pada jangka waktu yang sangat singkat. Apabila tidak memiliki komitmen, maka proyek RAD akan gagal.
2.	Lebih cepat daripada model <i>waterfall</i> yang dapat dimodulasi jika kebutuhan atau kendala proyek diketahui dengan baik.	Tidak semua proyek dapat menggunakan RAD, jadi tidak semua proyek dapat diuraikan dimodulasi.
3.	Keterlibatan pengguna semakin meningkat karena merupakan bagian dari tim secara keseluruhan.	Proyek ini dibagi menjadi beberapa bagian, sehingga banyak orang membentuk tim untuk mengerjakan bagian tim masing-masing.
4.	Memiliki kemampuan untuk menggunakan kembali komponen yang ada, sehingga pengembang tidak perlu membuat ulang aplikasi dari awal. Hal ini dapat meningkatkan waktu pengembangan.	Jika terjadi perubahan selama pengerjaan berlangsung, maka harus membuat komitmen baru antara pengembang dan pelanggan.
5.		Memerlukan tenaga kerja yang banyak, apabila mengerjakan sebuah proyek dalam skala besar.

## 2.8 *Blackbox Testing*

*Blackbox Testing* merupakan teknik pengujian fungsionalitas tanpa melihat struktur kode internal, detail implementasi, dan pengetahuan mengenai jalur internal perangkat lunak. Jenis pengujian ini sepenuhnya didasarkan pada persyaratan perangkat lunak dan spesifikasi. Pengujian *Blackbox* dilakukan dengan menjalankan program dan menguji fungsionalitas yang ditampilkan secara langsung. Fokus utama pengujian ini adalah pada masukan dan keluaran dari sistem perangkat lunak. *Blackbox Testing* dapat menjadi sistem perangkat lunak apa pun yang diuji. Misalnya, sistem operasi seperti *Windows*, situs web seperti Google, basis data seperti *Oracle*, atau bahkan aplikasi kustom. [21].

## 2.9 *System Usability Scale (SUS)*

*System Usability Scale (SUS)* merupakan salah satu teknik evaluasi *usability* yang dilakukan langsung terhadap pemberi nilai atau pengguna (*end user*). Penggunaan *SUS* dalam menentukan *usability* aplikasi dapat digunakan dengan mudah, karena memiliki rentang skor 0-100, sehingga dapat dihitung dengan mudah dan tidak memerlukan biaya. Selain itu, penggunaan *SUS* terbukti *valid* dan *reliable*, meskipun menggunakan sampel yang kecil. Pengujian *SUS* dilakukan dengan menggunakan kuisioner dengan 5 poin skala *Likert*. Pada pengujian *SUS*, responden dapat memberikan penilaian berupa “Sangat tidak setuju”, “Tidak setuju”, “Ragu-ragu”, “Setuju”, dan “Sangat setuju” dari 10 pertanyaan *SUS* yang sesuai dengan penilaian subyektif. Gambar 2.7 di bawah ini merupakan 10 pernyataan *SUS*.

Kode	Item Pernyataan
R1	Saya akan sering menggunakan/mengunjungi situs ini
R2	Saya menilai situs ini terlalu kompleks (memuat banyak hal yang tidak perlu)
R3	Saya menilai situs ini mudah dijelajahi
R4	Saya membutuhkan bantuan teknis untuk menggunakan/menjelajahi situs ini
R5	Saya menilai fungsi/fitur yang disediakan pada situs ini dirancang dan disiapkan dengan baik
R6	Saya menilai terlalu banyak inkonsistensi pada situs ini
R7	Saya merasa kebanyakan orang akan mudah menggunakan/menjelajahi situs ini dengan cepat
R8	Saya menilai situs ini sangat rumit untuk dijelajahi
R9	Saya merasa sangat percaya diri menjelajahi situs ini
R10	Saya perlu belajar banyak hal sebelum saya dapat menjelajahi situs ini dengan baik

Gambar 2.7 Item Pertanyaan *System Usability Scale*

Dari setiap pertanyaan yang diajukan, mempunyai skor kontribusi dengan rentang nilai dari 0-4. Pada pernyataan dengan nomor ganjil, kontribusi terhadap skor adalah posisi skala dikurangi 1. Sedangkan pada pernyataan dengan nomor genap, skor kontribusi dihitung dengan mengurangkan posisi skala dari 5. Berikut rumus perhitungan skor *SUS*.

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

Skor dari *System Usability Scale (SUS)* menggambarkan tingkat penerimaan pengguna. Skor *SUS* dianggap *acceptable* apabila mencapai nilai minimal 70. Untuk kategori *Good*, skor harus memiliki nilai lebih dari 70.4. Sedangkan, predikat A diberikan untuk nilai yang melebihi skor 90 [22]. Gambar 2.8 di bawah ini merupakan contoh jawaban *SUS*.



<i>Item</i> Pertanyaan	Jawaban	Nilai $Q_i$	<i>Item</i> Pertanyaan	Jawaban	Nilai $Q_i$
Q <sub>1</sub>	Netral	3	Q <sub>6</sub>	Netral	3
Q <sub>2</sub>	Netral	3	Q <sub>7</sub>	Setuju	4
Q <sub>3</sub>	Setuju	4	Q <sub>8</sub>	Netral	3
Q <sub>4</sub>	Tidak setuju	2	Q <sub>9</sub>	Netral	3
Q <sub>5</sub>	Netral	3	Q <sub>10</sub>	Tidak Setuju	2
Skor SUS = 60					

Gambar 2.8 Contoh Jawaban *System Usability Scale*

Sumber : *E-book* Referensi Mengukur *Usability* Perangkat Lunak

## 2.10 Penelitian Terkait

Penelitian yang dilakukan oleh Indah Utami Ilyas dan Hari Setiaji, yang berjudul “Pengembangan *Dashboard* Sistem Informasi Manajemen Perkuliahan (Studi Kasus: Fakultas Teknologi Industri Universitas Islam Indonesia)” membahas tentang pembangunan sistem informasi manajemen perkuliahan yang menyajikan informasi untuk mengetahui perkembangan perkuliahan selama beberapa periode, serta mengolah data dalam bentuk informasi yang berguna dalam kegiatan *monitoring* perkuliahan. Penelitian dikembangkan dengan menggunakan metode *pureshare* serta penerapan konsep *dashboard Drill Down*. Hasil dari penelitian ini ialah visualisasi data dalam bentuk *dashboard* untuk membantu pihak manajemen FTI UII mengolah informasi yang bermanfaat [23].

Penelitian yang dilakukan oleh Ardian Riftha Dhuha, Fajar Pradana, dan Bayu Priyambadha berjudul “Pengembangan Sistem Aplikasi Manajemen Proyek Berbasis Web (Studi Kasus: PT. Swadaya Graha)” membahas tentang pembangunan sistem aplikasi manajemen proyek yang memiliki beberapa fungsi, yaitu membantu perencanaan, pengawasan, dan memberikan laporan data secara *real-time* kepada para pemangku kepentingan. Sistem ini dikembangkan menggunakan model *Rapid Application Development*, dengan metode pemrograman berbasis *object oriented programming* dan menerapkan konsep model *view controller* [24].

Penelitian yang dilakukan oleh Helidorus Tugil, yang berjudul “Perancangan Sistem Informasi Manajemen Ekstrakurikuler berbasis Web pada Sekolah

Menengah Pertama Swasta Muhammadiyah Kupang”, membahas mengenai pengembangan sistem informasi pada sekolah menengah pertama swasta Muhammadiyah Kupang yang dibuat guna membantu manajemen pelaksanaan ekstrakurikuler agar pengelolaan kegiatan dapat terkelola dan terdata dengan baik. Sistem ini dibuat dengan menggunakan bahasa pemrograman PHP dengan menggunakan metode *waterfall*. Hasil dari penelitian ini ialah suatu sistem informasi yang berguna untuk memudahkan dan mengontrol pengajar, anggota dan kepala sekolah dalam melaksanakan kegiatan ekstrakurikuler [25].

Penelitian yang dilakukan oleh M. Yudha Putra dan Dwi Ely Kurniawan, yang berjudul “Implementasi Sistem *Reminder* Jadwal pada *eLearning Moodle* Berbasis API Menggunakan *Framework* Flutter”, membahas penerapan sistem *reminder* jadwal pada *platform eLearning Moodle* menggunakan *framework* Flutter dan menggunakan metode pengembangan dengan model *Rapid Application Development*. Tujuan dari penelitian ini adalah membantu mahasiswa mengelola jadwal perkuliahan, tugas, dan kegiatan lainnya dengan lebih efisien, sehingga waktu belajar dapat dioptimalkan. Penelitian ini juga bertujuan meningkatkan efisiensi pengelolaan waktu belajar dan prestasi akademik mahasiswa [26].

Penelitian yang dilakukan oleh Grinaldi Wisnu Tri Prasetyo, Fajar Pradana, dan Bondan Sapta Prakoso, yang berjudul “Pengembangan Aplikasi *Point of Sales* Warung dan UMKM ‘WarunkQu’ menggunakan *framework* Flutter”, membahas tentang pengembangan aplikasi *point of sale* bernama WarunkQu menggunakan *framework* Flutter dan menggunakan metode *Rapid Application Development*. Aplikasi yang dibuat memiliki tujuan untuk membantu usaha mikro, kecil, dan menengah (UMKM) dalam mencatat dan menghasilkan laporan penjualan. Hasil penelitian menunjukkan tingkat keberhasilan yang tinggi dan penerimaan yang baik dari pengguna. Penelitian ini menyimpulkan dengan menekankan pentingnya digitalisasi bagi UMKM dan manfaat penggunaan aplikasi WarunkQu [27].

Penelitian yang dilakukan oleh Ahmad Fauzi dan Eko Harli, yang berjudul

“Rancang Bangun Sistem Informasi Akademik SMK Negeri 1 Depok Berbasis Android dengan Pendekatan *Rapid Application Development*”, membahas mengenai pengembangan sistem informasi akademik pada SMA 1 Depok yang digunakan untuk membantu manajemen akademik dengan memberikan informasi tentang *profile* siswa, nilai siswa, berita akademik dan jadwal Pelajaran. Sistem ini dibuat dengan menerapkan metodologi pengembangan model *Rapid Application Development (RAD)*. Hasil penelitian menunjukkan penggunaan metode RAD dapat digunakan dalam membangun sistem informasi dengan baik dan sesuai dengan kebutuhan fungsional sistem [28].

### III. METODOLOGI PENELITIAN

#### 3.1 Waktu dan Tempat Penelitian

Waktu dan tempat pelaksanaan penelitian dilakukan pada :

1. Waktu penelitian : Juni 2023 sampai dengan November 2023
2. Tempat Penelitian : Universitas Lampung

Adapun jadwal kegiatan penelitian ini adalah sebagai berikut:

Tabel 3.1 Jadwal Penelitian

No	Aktivitas	Juni 2023	Juli 2023	Agustus 2023	September 2023	Oktober 2023	November 2023
1	<i>Requirement Planing</i>						
2	Iterasi 1						
3	Iterasi 2						
4.	<i>Cutover</i>						
5.	Pengerjaan Laporan						

#### 3.2 Alat dan Bahan Penelitian

Adapun alat dan bahan yang dibutuhkan pada penelitian ini adalah sebagai berikut:

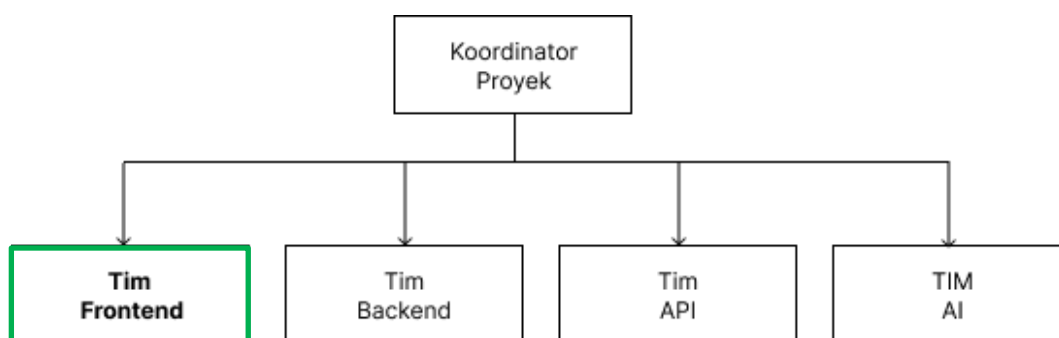
Tabel 3.2 Alat (*Hardware* dan *Software*) pengembangan sistem

No	Perangkat	Spesifikasi	Kegunaan
1	Laptop	AMD athlon <i>silver</i> 3050u	Perangkat yang digunakan untuk membuat aplikasi.
2	<i>Smartphone (Android)</i>	Android 10	Perangkat yang digunakan untuk melakukan pengujian Aplikasi.

3	<i>Visual Studio Code</i>	<i>Version 1.73.1</i>	Perangkat Lunak yang digunakan sebagai <i>text editor</i> dart.
4	Dart	<i>Dart version 2.17.6</i>	Bahasa Pemrograman yang digunakan untuk menjalankan <i>Framework</i> Flutter.
5	Flutter	<i>Flutter version 3.0.5</i>	<i>Framework</i> yang digunakan untuk membangun aplikasi.
6	<i>Design Requirement</i>	<i>Draw.io</i>	Perangkat lunak yang digunakan untuk mendesain UML.
7	<i>Mockup</i>	Figma	Perangkat lunak yang digunakan untuk desain <i>mockup</i> aplikasi.

Bahan penelitian yang diperlukan pada penelitian ini adalah data hasil wawancara, data *user* (dosen) dan data mahasiswa. Untuk data dosen, informasi yang diperlukan adalah nama lengkap dosen, NIP (Nomor Induk Pegawai), dan Mata Kuliah yang diampu. Selain itu, diperlukan juga data mahasiswa, meliputi nama lengkap mahasiswa, NPM (Nomor Pokok Mahasiswa).

### 3.3 Struktur Tim *Capstone*



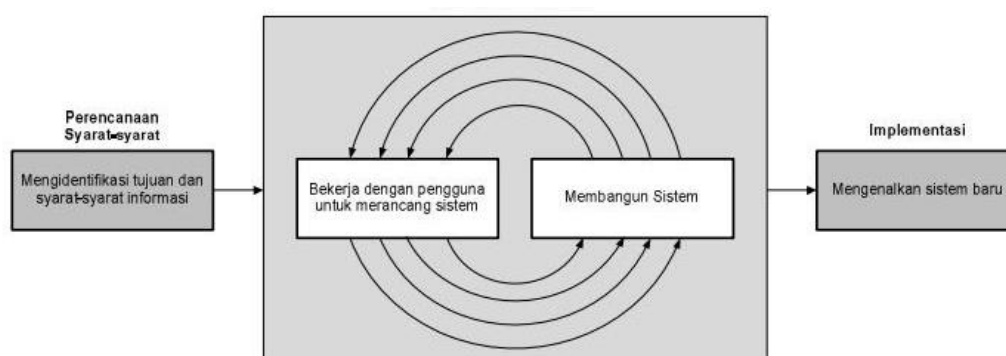
Gambar 3.1 Struktur Tim *Capstone*

Gambar 3.1 merupakan gambar struktur tim *capstone*. Tim *capstone* terdiri dari Koordinator Proyek yang bertugas dalam mengelola, mengkoordinasi tim, dan

memantau kemajuan aplikasi. Pada pembuatan aplikasi, Tim *Front-End* bertugas mengembangkan antarmuka pengguna, mengambil dan menampilkan data dari API pada antarmuka, serta melakukan integrasi dengan *back-end* agar perubahan-perubahan yang terjadi dapat tampil pada aplikasi. Tim *Front-End* juga memiliki peran dalam melakukan pengujian terhadap antarmuka pengguna. Selain itu, terdapat Tim *Back-End* yang bertugas mengembangkan infrastruktur *back-end*, Tim API yang bertugas untuk mengembangkan API yang menghubungkan *frontend* dan *backend*, serta Tim AI yang mengintegrasikan komponen kecerdasan buatan (AI).

### 3.4 Tahapan Penelitian

Tahapan yang dilakukan pada penelitian ini mengikuti model pengembangan perangkat lunak *Rapid Application Development (RAD)*. Metode RAD memungkinkan perubahan dan penyesuaian yang cepat selama proses pengembangan. Metode ini juga melibatkan kolaborasi yang erat antara pengembang dan pengguna, dimana melibatkan dosen sebagai pengguna aplikasi. Hal ini untuk memastikan bahwa kebutuhan dosen terpenuhi sehingga aplikasi dapat diadaptasi dengan baik. Gambar 3.1 di bawah ini menggambarkan tahapan penelitian.



Gambar 3.2 Tahapan Penelitian.

Sumber : *E-book Analisis dan Perancangan Sistem*

### 3.4.1 *Requirements Planning*

*Requirements planning* atau perencanaan kebutuhan adalah tahap dilakukannya identifikasi terhadap tujuan pembuatan aplikasi dan kebutuhan pengguna bertujuan untuk mengetahui permasalahan serta pengumpulan data dan informasi mengenai penelitian. Pada penelitian ini, dilakukan perencanaan dengan cara sebagai berikut :

#### 3.4.1.1 Studi Literatur

Tahap Studi literatur dilakukan untuk memberikan referensi teoritis yang diperlukan. Tahap ini dilakukan dengan membaca dan menganalisis literatur-literatur sebelumnya yang berkaitan dengan topik yang sedang diteliti.

#### 3.4.1.2 Wawancara

Tahap Wawancara dilakukan dengan proses tanya jawab secara lisan untuk mengumpulkan informasi langsung dari pengguna tentang kebutuhan dan masalah yang dihadapi oleh pengguna (dosen) terkait dengan pengelolaan aktivitas perkuliahan serta saran untuk perancangan aplikasi.

Tabel 3.3. Jumlah Informan

No	Informan	Jumlah Informan
1	Dosen Teknik Informatika Universitas Lampung	6

Pertanyaan wawancara ditentukan dengan mengikuti pendekatan yang terstruktur. Pertama, melakukan studi literatur untuk memahami permasalahan yang diteliti dan kebutuhan pengguna. Selanjutnya, mengidentifikasi area utama yang perlu dijelajahi dalam konteks pengelolaan aktivitas perkuliahan. Kemudian, merumuskan pertanyaan-pertanyaan yang relevan dan spesifik, yang dapat memberikan informasi mendalam tentang kebutuhan, harapan, masalah serta saran terkait pembuatan aplikasi. Adapun beberapa pertanyaan yang ditanyakan adalah sebagai berikut.

1. Bagaimana saat ini Bapak/Ibu mengelola aktivitas perkuliahan, seperti absensi, penilaian akademis seperti kuis dan tugas, serta mendapatkan laporan penilaian mahasiswa?
2. Apa kendala yang ditemui oleh Bapak/Ibu dalam mengelola aktivitas

perkuliahan saat ini?

3. Apa saja yang menjadi kebutuhan utama Bapak/Ibu dalam pengelolaan aktivitas perkuliahan?
4. Apa fitur dan fungsionalitas utama yang harus ada dalam aplikasi pengelolaan perkuliahan?
5. Apa aspek yang sangat penting untuk diperhatikan dalam proses pengembangan aplikasi pengelolaan perkuliahan?
6. Apakah Bapak/Ibu memiliki saran dan masukan mengenai fitur-fitur yang dapat meningkatkan aplikasi ini?
7. Bagaimana pandangan Bapak/Ibu terhadap pembuatan aplikasi ini?

Dari hasil wawancara yang dilakukan, didapatkan kesimpulan bahwa :

1. Hasil wawancara dengan keenam dosen menunjukkan variasi dalam pengelolaan aktivitas perkuliahan. 2 dari 6 dosen memilih menggunakan SIAKADU, sementara 4 dosen lainnya memilih *Vclass* untuk beberapa aktivitas perkuliahan. Selain itu, terdapat 3 dosen yang masih menggunakan metode absensi dan penginputan data nilai secara manual.
2. Hasil rangkuman dari seluruh wawancara menunjukkan beberapa kendala yang dihadapi oleh dosen dalam mengelola aktivitas perkuliahan, yaitu kesulitan dalam memproses absensi dengan cepat, mengumpulkan dan menyimpan data penilaian, serta kesulitan untuk mengetahui hasil jawaban tugas dan kuis merupakan hasil pengerjaan sendiri.
3. Berdasarkan rangkuman seluruh wawancara, didapatkan kebutuhan dosen yaitu suatu aplikasi yang dapat mendukung proses perkuliahan. Aplikasi yang mampu melakukan penilaian akademis dan mencatat segala penilaian tersebut.
4. Berdasarkan hasil wawancara, didapatkan bahwa fitur dan fungsionalitas utama yang diperlukan pada aplikasi mencakup sistem absensi, integrasi modul pembelajaran, kemampuan penilaian tugas/kuis serta pelaporan aktivitas perkuliahan yang *up to date*.
5. Berdasarkan rangkuman dari hasil wawancara bersama enam dosen menunjukkan bahwa aspek-aspek yang perlu diperhatikan dalam proses pengembangan aplikasi yaitu aplikasi yang intuitif, mudah digunakan,



memiliki navigasi yang jelas, dan bahasa yang konsisten.

6. Berdasarkan hasil wawancara dengan keenam dosen, didapatkan saran oleh 5 dari 6 dosen berupa adanya fitur notifikasi sebagai pengingat jadwal, serta integrasi dengan kalender akademik. Sementara itu, satu dosen lainnya tidak memberikan saran ataupun masukan.
7. Berdasarkan hasil wawancara, semua dosen berpandangan bahwa pembuatan aplikasi ini penting dilakukan untuk menjawab kendala dan kebutuhan yang dihadapi oleh para dosen dalam pengelolaan aktivitas perkuliahan.

### 3.4.1.3 Kebutuhan Fungsional dan Non Fungsional

Setelah mengidentifikasi tujuan pembuatan aplikasi dan kebutuhan pengguna, ditemukan kebutuhan pengguna yang terdiri dari kebutuhan fungsional dan non fungsional. Kebutuhan fungsional adalah kebutuhan yang menjelaskan fungsi atau fitur yang diharapkan dari aplikasi. Sedangkan kebutuhan non fungsional menjelaskan fungsi dan fitur yang mempengaruhi kinerja, performa, dan pengalaman pengguna. Adapun kebutuhan fungsional dan non fungsional pada penelitian ini adalah sebagai berikut :

Tabel 3.4 Kebutuhan Fungsional

No	Kode Fungsi	Nama Fungsi	Deskripsi
1	SL – F- 001	<i>Login</i>	Dosen masuk ke aplikasi dengan menggunakan fungsi <i>login</i>
2.	SL – F- 002	<i>Logout</i>	Dosen keluar dari aplikasi dengan menggunakan fungsi <i>logout</i>
3.	SL – F- 003	Daftar Mata Kuliah	Dosen memilih daftar mata kuliah yang diampu.
4.	SL – F- 004	Daftar Pertemuan	Dosen memilih daftar pertemuan dari setiap mata kuliah.

5.	SL – F- 005	Detail Pertemuan	Dosen memilih opsi yang tersedia yaitu modul dan penilaian mahasiswa.
6.	SL – F- 006	Modul Pembelajaran	Dosen membaca modul yang tersedia untuk setiap pertemuan mata kuliah yang diampu.
7.	SL – F- 007	Penilaian Mahasiswa	Dosen memilih opsi penilaian mahasiswa yang tersedia pada setiap pertemuan .
8.	SL – F- 008	Absensi	Dosen melakukan absensi kepada mahasiswa dan mendapatkan daftar absensi.
9.	SL – F- 009	Keaktifan	Dosen menilai keaktifan serta mendapatkan daftar keaktifan.
10.	SL – F- 0010	Kuis	Dosen melakukan penilaian kuis serta mendapatkan daftar nilai.
11.	SL – F- 0011	Tugas	Dosen dapat menilai tugas dan melihat daftar nilai tugas.
12.	SL – F- 0012	UTS (Ujian Tengah Semester).	Dosen dapat melakukan penilaian UTS dan mendapatkan daftar nilai UTS.
13.	SL – F- 0013	UAS (Ujian Akhir Semester).	Dosen menilai UAS serta dapat melihat daftar nilai UAS.
14.	SL – F- 0014	Mengunduh Laporan Penilaian Mahasiswa	Dosen dapat mengakses dan mengunduh laporan penilaian mahasiswa untuk setiap pertemuan.

Tabel 3.5 Kebutuhan Non Fungsional

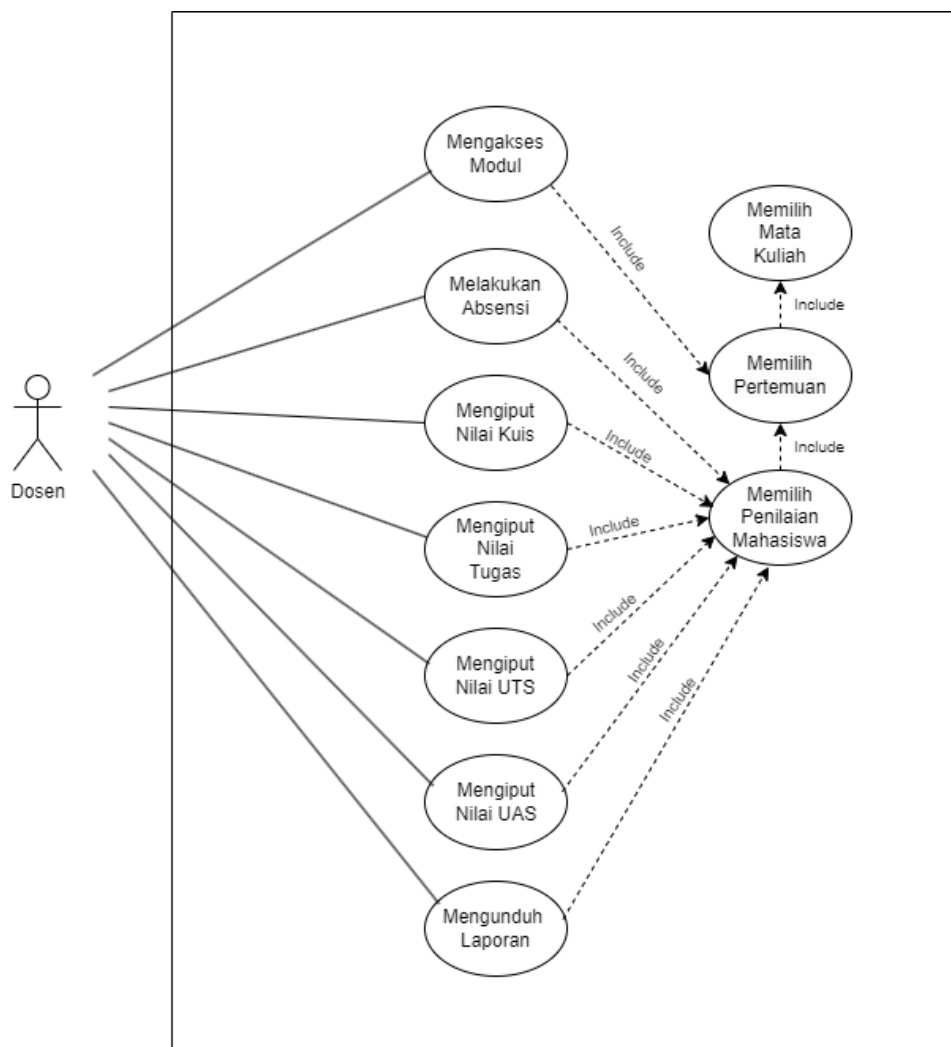
No	Kode Fungsi	Nama Fungsi	Deskripsi
1	SL – NF- 001	<i>Usability</i>	Aplikasi mudah digunakan, intuitif, dan menyediakan navigasi yang jelas bagi pengguna.
2	SL – NF- 002	<i>Performance</i>	Aplikasi memiliki kinerja yang cepat dan responsif.
3	SL – NF- 003	<i>Memorability</i>	Aplikasi memiliki alur yang mudah diingat oleh pengguna.
4	SL – NF- 004	<i>Language</i>	Aplikasi menggunakan bahasa Indonesia.

### 3.4.2 User Design

Desain pengguna atau *User design* merupakan tahapan dimana melakukan proses desain rancangan aplikasi dan perbaikan, apabila masih terdapat ketidaksesuaian desain antara pengguna dan pengembang. Adapun setelah mengidentifikasi kebutuhan pengguna, selanjutnya membuat *use case* bisnis untuk aplikasi aktivitas perkuliahan, terdapat beberapa diagram yang digunakan selama pengembangan yaitu :

#### 3.4.2.1 Use Case Diagram

Penggunaan *Use case diagram* pada penelitian ini bertujuan untuk menjelaskan interaksi antara *user* yang ada dengan sistem, serta untuk menjelaskan proses-proses yang berlangsung di dalam sebuah sistem. Gambar 3.3 di bawah ini menggambarkan hasil diagram *use case* dalam penggunaan aplikasi.



Gambar 3.3 Usecase Diagram

Pada *Use case Diagram* di atas menunjukkan bahwa aktor utama aplikasi ini yaitu dosen, Dosen dapat mengakses modul di tiap pertemuan dan dosen juga dapat melakukan absensi mahasiswa, penilaian terhadap keaktifan mahasiswa, penilaian tugas dan kuis yang diberikan kepada mahasiswa, penilaian UTS (Ujian Tengah Semester) dan UAS (Ujian Akhir Semester), serta mengunduh laporan penilaian pada pertemuan tersebut. Untuk melakukan aktivitas tersebut, dosen terlebih dahulu memilih penilaian mahasiswa dan memilih pertemuan dari suatu mata kuliah. *Use case diagram* tersebut memberikan gambaran tentang interaksi antara dosen dan aplikasi aktivitas perkuliahan mahasiswa. Tabel 3.5 di bawah ini mendeskripsikan *use case* pada aplikasi aktivitas perkuliahan mahasiswa.

Tabel 3.6 *Usecase diagram*

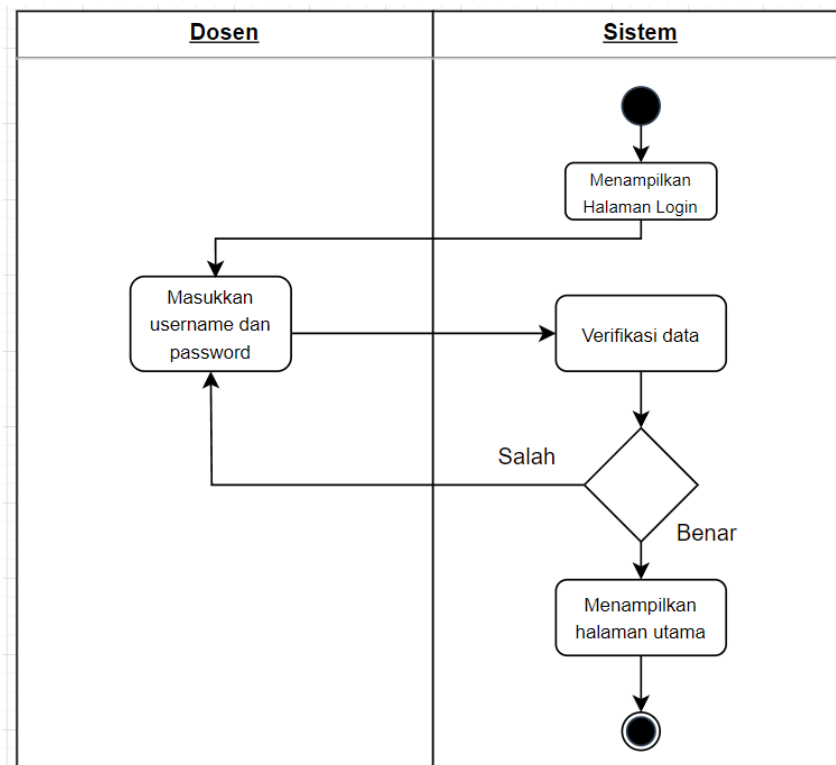
<b>No.</b>	<b><i>Use case</i></b>	<b>Deskripsi</b>
1.	Memilih mata kuliah	Dosen memilih mata kuliah yang telah terdaftar pada sistem ( <i>Pre-condition</i> : telah melalui proses <i>login</i> )
2.	Memilih pertemuan	Dosen memilih daftar pertemuan dari suatu mata kuliah untuk menilai aktivitas mahasiswa pada pertemuan tersebut. ( <i>Pre-condition</i> : telah melalui proses <i>login</i> )
3.	Memilih Penilaian Mahasiswa	Dosen memilih opsi penilaian mahasiswa untuk melihat daftar penilaian. ( <i>Pre-condition</i> : telah melalui proses <i>login</i> )
3.	Membaca Modul	Dosen membuka dan melihat isi dari modul yang telah disediakan. ( <i>Pre-condition</i> : telah melalui proses <i>login</i> )
4.	Melakukan Absensi	Dosen melakukan absensi untuk mencatat kehadiran mahasiswa. ( <i>Pre-condition</i> : telah melalui proses <i>login</i> )
5.	Menilai Keaktifan	Dosen menilai keaktifan mahasiswa pada suatu pertemuan. ( <i>Pre-condition</i> : telah melalui proses <i>login</i> )
6.	Menginput nilai kuis	Dosen menginputkan nilai pada kuis yang telah diikuti oleh mahasiswa. ( <i>Pre-condition</i> : telah melalui proses <i>login</i> )
7.	Menginput nilai Tugas	Dosen memasukkan nilai pada tugas yang telah dikumpulkan oleh mahasiswa. ( <i>Pre-condition</i> : telah melalui proses <i>login</i> )
8.	Menginput nilai UTS	Dosen melakukan penilaian UTS kepada mahasiswa yang mengikuti UTS ( <i>Pre-condition</i> : telah melalui proses <i>login</i> )

9.	Menginput nilai UAS	Dosen menginputkan nilai UAS kepada mahasiswa yang mengikuti UAS. ( <i>Pre-condition</i> : telah melalui proses <i>login</i> )
10.	Mengunduh laporan penilaian	Dosen mengunduh laporan penilaian mahasiswa yang berisi hasil penilaian aktivitas mahasiswa pada suatu pertemuan. ( <i>Pre-condition</i> : telah melalui proses <i>login</i> )

### 3.4.2.2 Activity Diagram

Penggunaan *activity diagram* pada penelitian ini adalah untuk menjelaskan lebih lanjut dari *Use case diagram* di atas dengan mendeskripsikan aktor, tindakan yang perlu dilakukan, serta kapan harus terjadi. Berikut merupakan *activity diagram* penggunaan aplikasi.

#### a. Activity Diagram Masuk Aplikasi

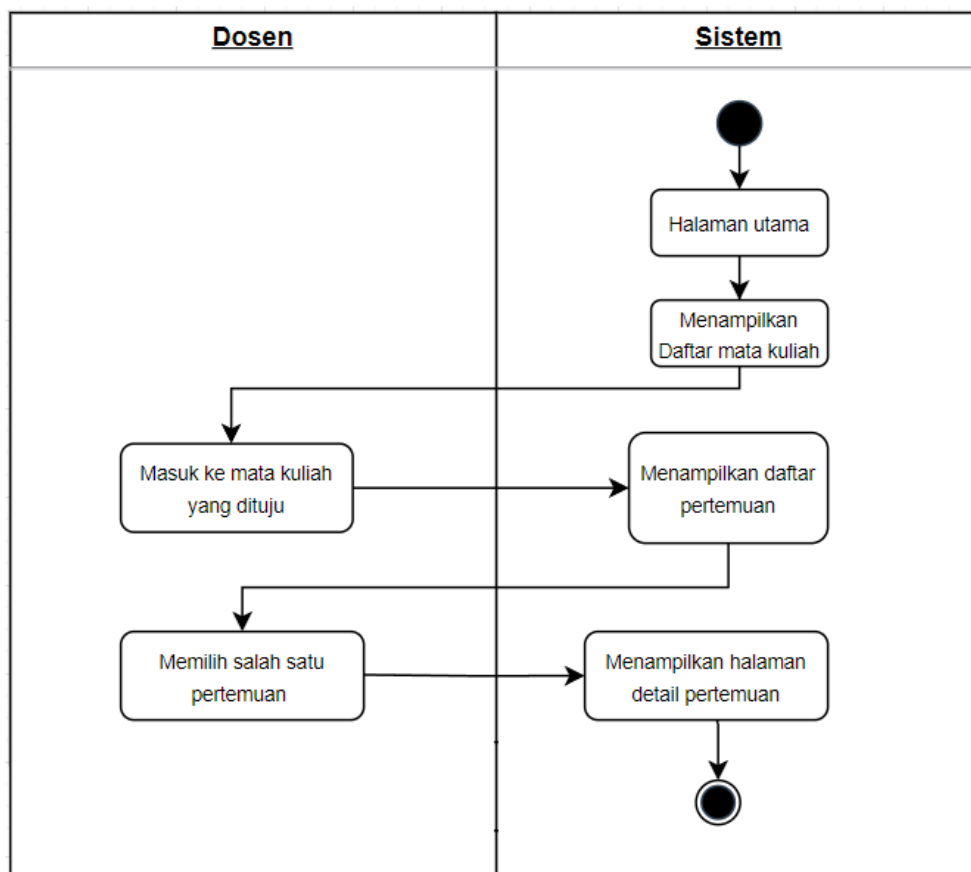


Gambar 3.4 Activity Diagram Masuk Aplikasi

Gambar 3.4 merupakan gambaran *activity diagram* dari proses dosen masuk ke aplikasi. Proses ini diawali oleh sistem yang menampilkan halaman *login*. Pengguna kemudian dapat memasukkan *username* serta *password* pada *form* yang

tersedia, dan menekan tombol masuk. kemudian sistem melakukan verifikasi data inputan, apabila data inputan yang dimasukkan benar, maka sistem mengarahkan pengguna menuju halaman utama aplikasi. Namun, apabila data inputan yang dimasukkan pengguna salah, maka pengguna dapat mengulang proses *login*.

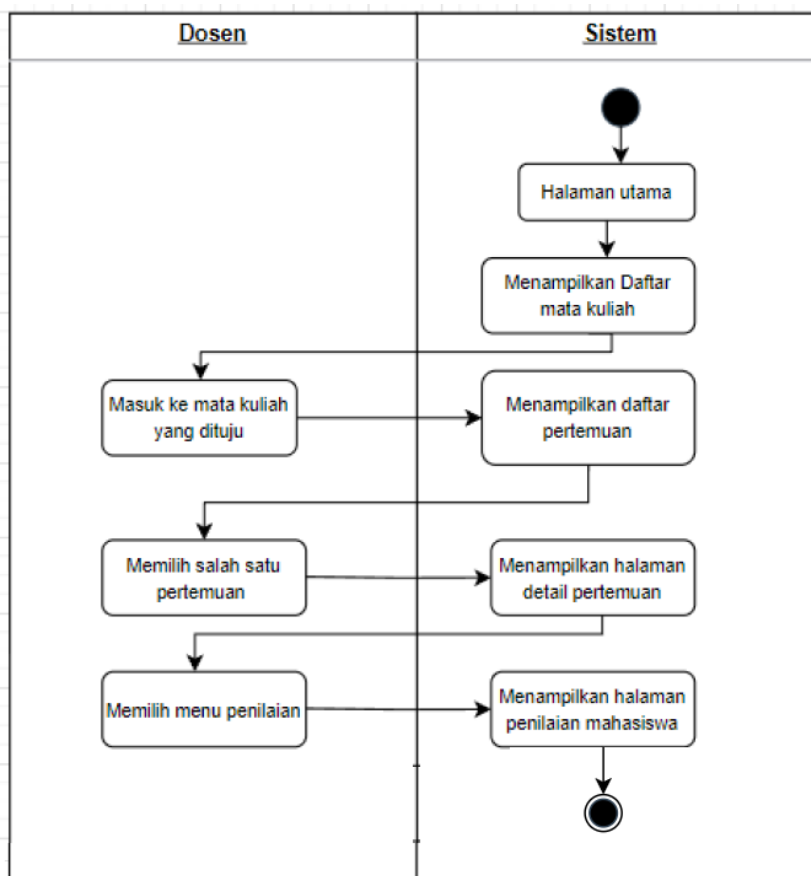
b. *Activity Diagram* Memilih Daftar Pertemuan



Gambar 3.5 *Activity Diagram* Memilih Daftar Pertemuan

Gambar 3.5 menggambarkan *activity diagram* dari proses dosen memilih daftar pertemuan. Aktivitas memilih pertemuan dimulai oleh sistem yang menampilkan halaman utama aplikasi. Pada halaman utama aplikasi ditampilkan pilihan daftar mata kuliah yang terdaftar pada sistem dan juga informasi tentang pengguna. Aksi pengguna pada halaman ini ialah memilih salah satu mata kuliah, yang mana sistem merespon aksi tersebut dengan mengarahkan pengguna menuju ke halaman daftar pertemuan pada mata kuliah tersebut. Pada halaman daftar pertemuan, pengguna memilih salah satu pertemuan yang tersedia dan sistem mengarahkan pengguna menuju halaman detail pertemuan tersebut.

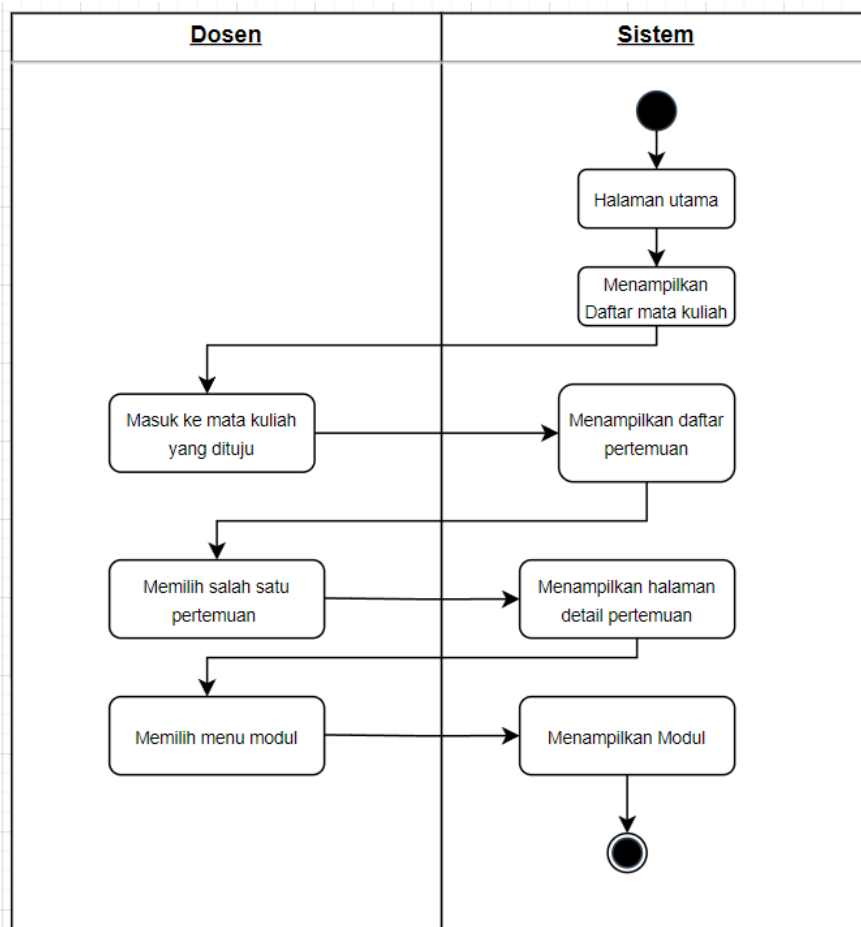
c. *Activity Diagram* Memilih Penilaian Mahasiswa



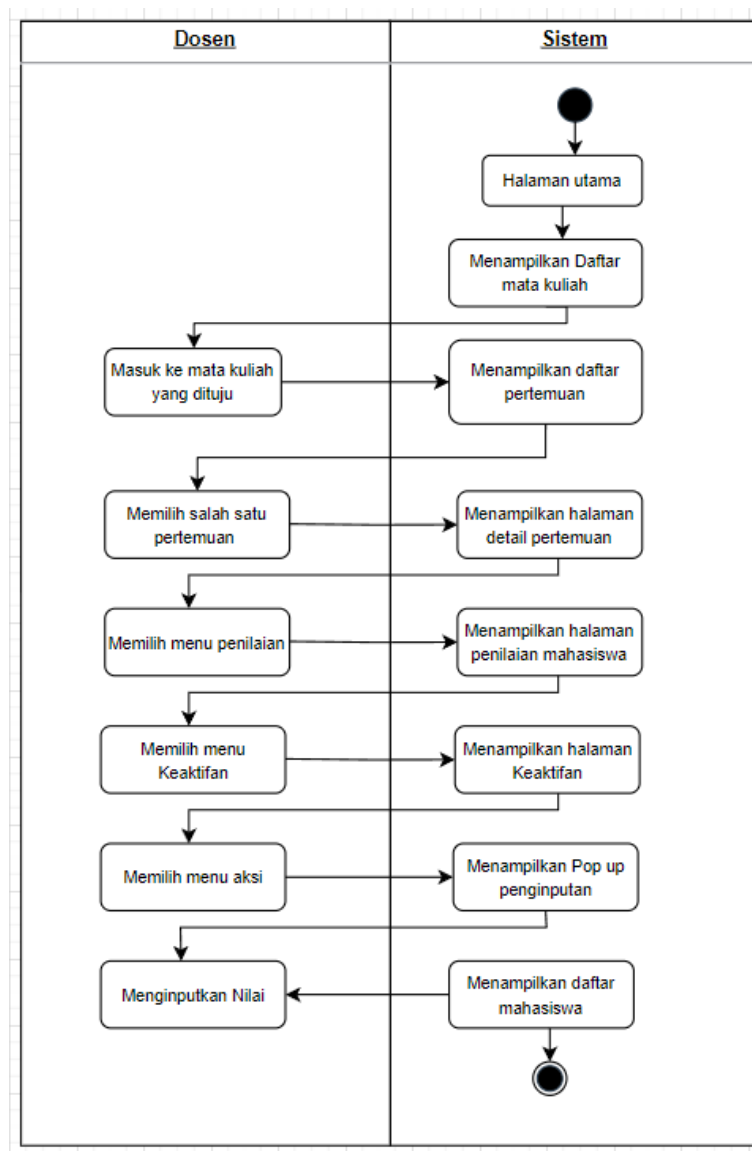
Gambar 3.6 *Activity Diagram* Memilih Penilaian Mahasiswa

Gambar 3.6 menggambarkan *activity diagram* dari proses dosen memilih Penilaian Mahasiswa. Untuk masuk pada halaman penilaian mahasiswa, sistem terlebih dahulu menampilkan halaman utama aplikasi. Pada halaman utama aplikasi, terdapat pilihan daftar mata kuliah yang terdaftar pada sistem. Pengguna dapat memilih salah satu mata kuliah, dan sistem mengarahkan pengguna menuju ke halaman daftar pertemuan pada mata kuliah tersebut. Pada halaman daftar pertemuan, pengguna memilih salah satu pertemuan yang tersedia untuk melihat detail dari pertemuan tersebut. Sistem merespon pilihan pengguna dengan menampilkan halaman detail pertemuan yang memuat opsi modul pembelajaran dan penilaian mahasiswa. Dengan memilih menu penilaian mahasiswa, maka sistem menampilkan halaman penilaian mahasiswa yang berisi beberapa opsi penilaian mahasiswa.



d. *Activity Diagram* Mengakses ModulGambar 3.7 *Activity Diagram* mengakses modul

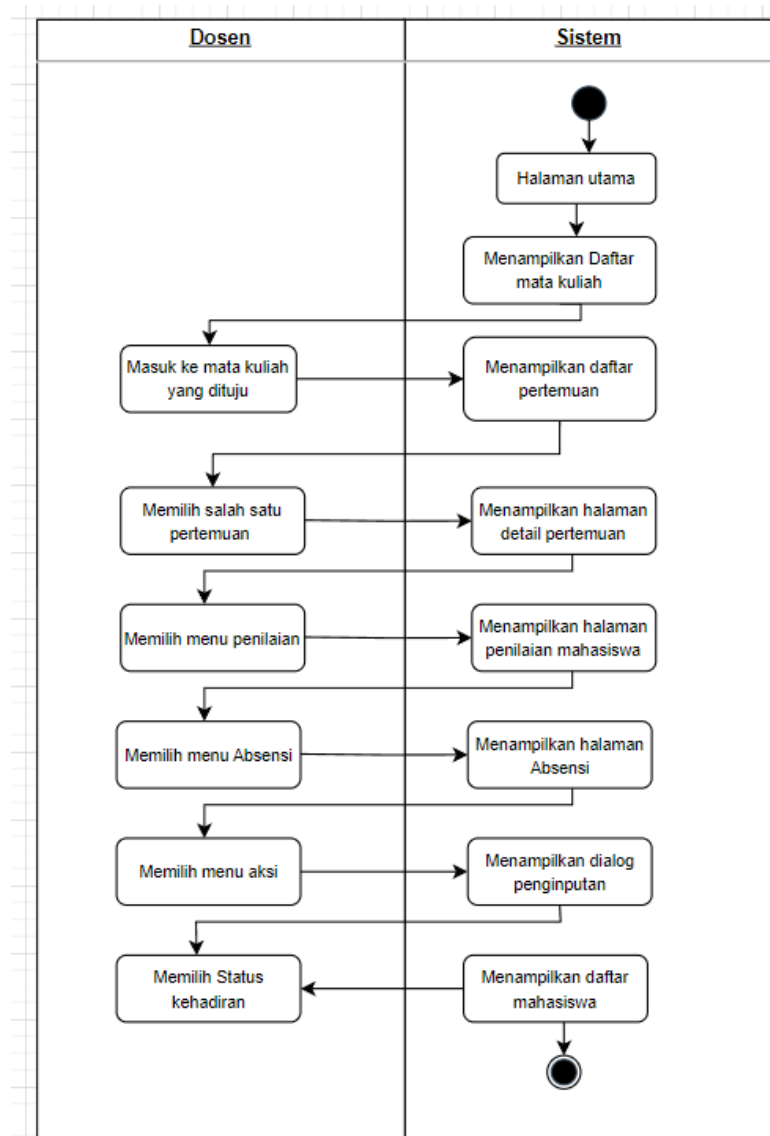
Gambar 3.7 menggambarkan *activity diagram* dari proses dosen mengakses modul. Alur *activity* ini dimulai dari sistem yang menampilkan halaman utama. Pada halaman utama, sistem menampilkan daftar mata kuliah yang dosen ampu dan pengguna dapat memilih mata kuliah yang dituju. Setelah itu, sistem menampilkan halaman daftar pertemuan pada mata kuliah tersebut. Pada halaman ini, pengguna dapat memilih salah satu pertemuan yang diakses. Setelah memilih salah satu pertemuan, pengguna diarahkan menuju ke halaman detail pertemuan yang menyediakan dua pilihan, yaitu modul pembelajaran dan penilaian mahasiswa. Untuk melihat modul pembelajaran, pengguna dapat memilih menu modul dan sistem mengarahkan pengguna menuju halaman modul yang dapat diakses oleh dosen. Pada halaman tersebut, pengguna ditampilkan modul pembelajaran yang menjadi bahan ajar pengguna pada pertemuan tersebut.

e. *Activity Diagram* Penilaian KeaktifanGambar 3.8 *Activity Diagram* Penilaian Keaktifan

Gambar 3.8 merupakan gambaran dari *activity diagram* dari aktivitas dosen menilai keaktifan mahasiswa. Untuk menilai keaktifan mahasiswa, proses diawali pada tampilan halaman utama yang menampilkan informasi pengguna dan mata kuliah yang diampu. Selanjutnya, pengguna dapat memilih mata kuliah yang dituju, lalu sistem menampilkan halaman daftar pertemuan. Pada halaman ini, pengguna dapat memilih salah satu pertemuan untuk melihat detail dari pertemuan tersebut. Untuk dapat melihat penilaian mahasiswa, pengguna dapat memilih opsi “penilaian mahasiswa” dari dua opsi yang tersedia, yaitu modul pembelajaran dan penilaian mahasiswa. Sistem kemudian menampilkan halaman penilaian

mahasiswa yang berisikan beberapa pilihan penilaian. Selanjutnya, pengguna dapat memilih menu keaktifan. Pada halaman ini, pengguna ditampilkan daftar nama mahasiswa yang mengikuti perkuliahan, dan pengguna dapat memilih menu aksi. Sebagai respon, sistem menampilkan *pop up* yang berisi *form* inputan nilai yang dapat diinputkan oleh pengguna. Setelah nilai keaktifan tersimpan, pengguna diarahkan kembali menuju halaman daftar keaktifan.

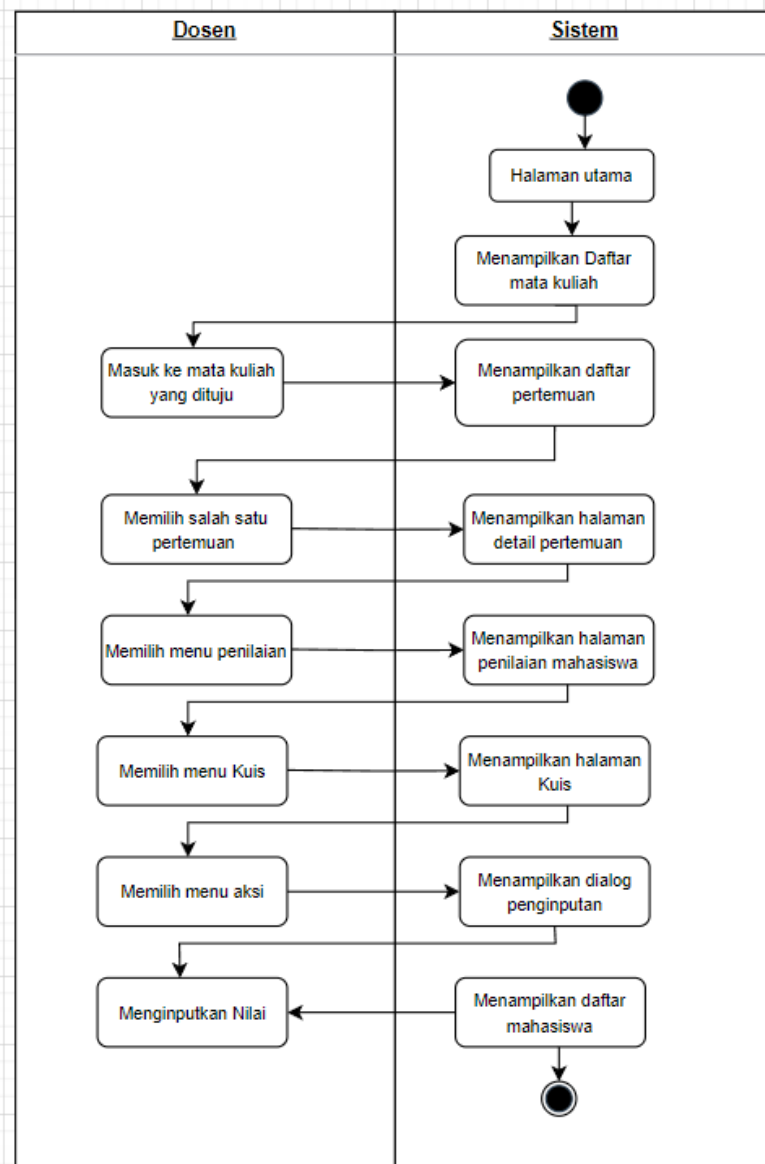
f. *Activity Diagram* Melakukan Absensi



Gambar 3.9 *Activity Diagram* Melakukan Absensi

Gambar 3.9 menggambarkan *activity diagram* dari aktivitas pengguna melakukan absensi. Alur *activity* ini dimulai dari sistem yang menampilkan tampilan halaman utama yang memuat informasi pengguna dan mata kuliah yang diampu.

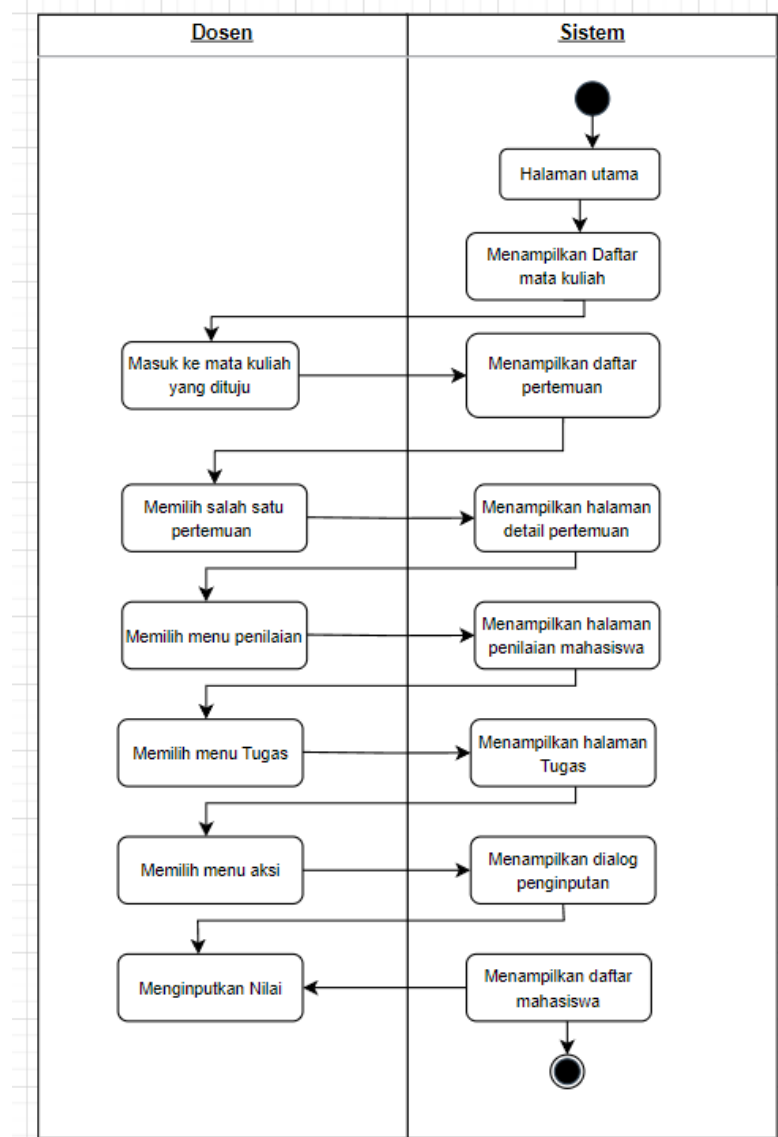
Selanjutnya, pengguna memilih salah satu daftar mata kuliah dan sistem mengarahkan pengguna menuju halaman daftar pertemuan. Pada halaman daftar pertemuan, pengguna dapat memilih salah satu pertemuan untuk melihat detail dari pertemuan tersebut. Sistem merespon pilihan pengguna dengan menampilkan halaman detail pertemuan yang memuat opsi modul pembelajaran dan penilaian mahasiswa. Dengan memilih menu penilaian mahasiswa, maka sistem menampilkan halaman penilaian mahasiswa yang berisi beberapa menu penilaian, salah satunya adalah absensi. Pengguna dapat memilih menu absensi yang mengarahkan pengguna menuju halaman absensi. Pada halaman ini, pengguna ditampilkan daftar nama mahasiswa yang mengikuti perkuliahan. Dengan memilih menu aksi disalah satu nama mahasiswa, pengguna dapat melakukan absensi untuk mahasiswa tersebut. Sistem menampilkan *pop up* yang berisi pilihan status kehadiran yang dapat dipilih oleh pengguna. Setelah menyimpan pilihan tersebut, maka sistem mengarahkan pengguna kembali pada halaman daftar absensi.

g. *Activity Diagram* Penilaian KuisGambar 3.10 *Activity Diagram* Penilaian Kuis

Gambar 3.10 merupakan gambaran dari *activity diagram* dari proses dosen menilai kuis mahasiswa. Untuk melakukan penilaian kuis, pengguna dapat memilih mata kuliah yang dituju pada halaman utama aplikasi. Selanjutnya sistem menampilkan halaman daftar pertemuan untuk mata kuliah terkait. Pada halaman daftar pertemuan, pengguna memiliki pilihan untuk memilih salah satu pertemuan dan melihat detailnya. Setelah berada pada halaman detail pertemuan, pengguna dapat memilih menu penilaian untuk melihat opsi penilaian kuis. Pada halaman penilaian, sistem menampilkan opsi penilaian yang terdapat menu kuis dan

pengguna dapat memilih menu kuis. Pada halaman kuis, pengguna ditampilkan daftar nama mahasiswa yang mengikuti perkuliahan dan pengguna dapat memilih menu aksi pada salah satu mahasiswa. Sistem merespon aksi pengguna dengan menampilkan *pop up* yang berisi *form* inputan nilai. Pengguna dapat menginput nilai berdasarkan nilai kuis mahasiswa pada pertemuan tersebut, dan menyimpan nilai tersebut. Kemudian tampilan kembali pada daftar mahasiswa

#### h. Activity Diagram Penilaian Tugas

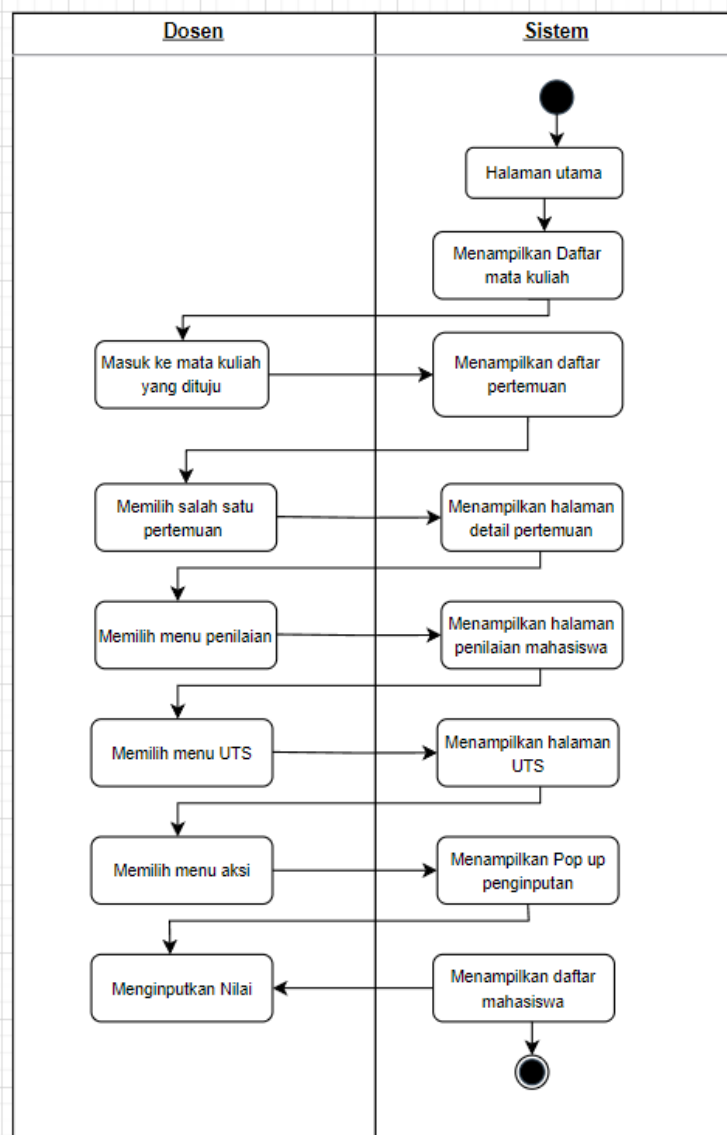


Gambar 3.11 Activity Diagram Penilaian Tugas

Gambar 3.11 menggambarkan *activity diagram* dari proses dosen menilai tugas mahasiswa. Proses dimulai dengan sistem yang menampilkan halaman utama dengan memuat informasi pengguna dan mata kuliah yang diajarkan. Pengguna

dapat memilih salah satu mata kuliah tujuannya. Selanjutnya, sistem menampilkan halaman daftar pertemuan berisi pertemuan-pertemuan mata kuliah terkait, dan pengguna dapat memilih pertemuan yang dituju. Aksi tersebut mengarahkan pengguna pada halaman detail pertemuan yang memuat pilihan modul pembelajaran dan penilaian mahasiswa. Dengan memilih menu penilaian, maka sistem menampilkan halaman penilaian mahasiswa yang mencakup pilihan penilaian, salah satunya adalah penilaian tugas. Pengguna dapat memilih menu tugas untuk melakukan penilaian tugas. Pada halaman tugas, pengguna dapat melihat daftar nama mahasiswa, dan pengguna dapat memilih menu aksi pada salah satu mahasiswa untuk menilai tugas mahasiswa tersebut. Setelah aksi pengguna tersebut, sistem menampilkan *pop up* yang berisikan *form* penilaian yang dapat diisi oleh pengguna. Setelah pengguna memasukkan nilai, pengguna dapat menyimpan nilai tersebut dan sistem mengarahkan pengguna menuju halaman daftar nilai mahasiswa.

i. Activity Diagram Penilaian UTS



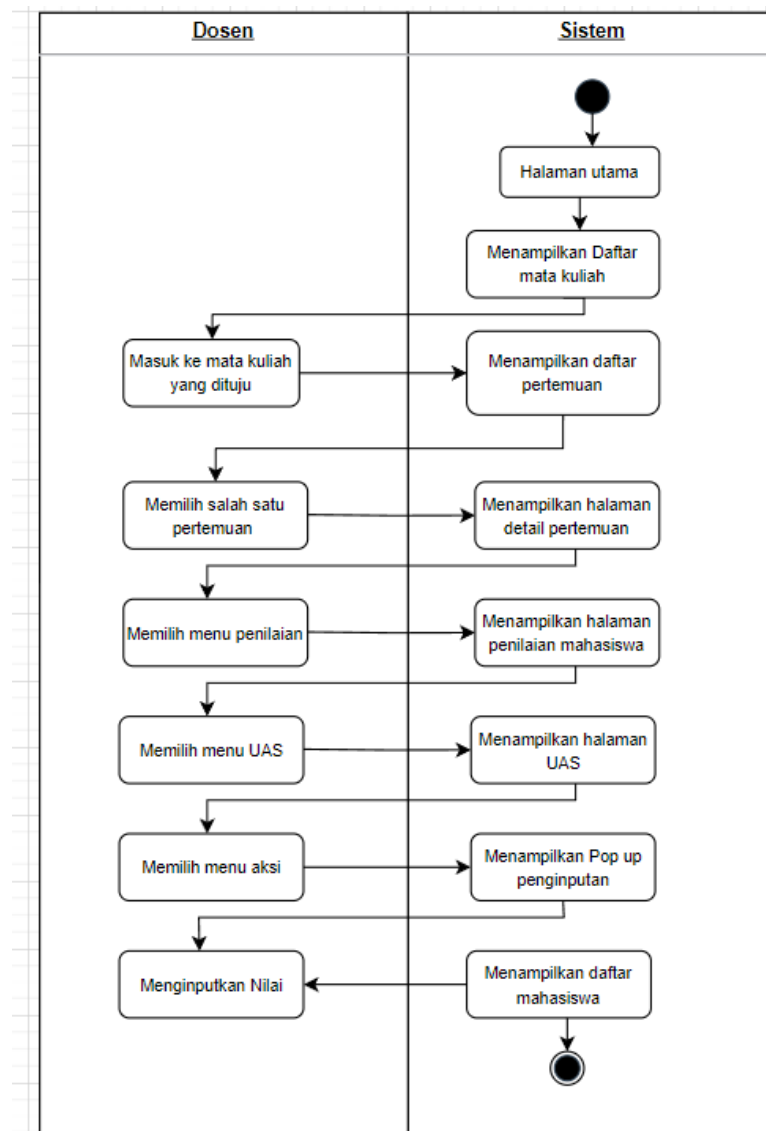
Gambar 3.12 Activity Diagram Penilaian UTS

Gambar 3.12 merupakan gambaran *activity diagram* proses pengguna menilai UTS (Ujian Tengah Semester) mahasiswa. Aktivitas menilai UTS diawali tampilan halaman utama dengan informasi tentang pengguna serta mata kuliah yang diajarkan. Aksi pengguna ialah memilih salah satu mata kuliah yang menjadi tujuannya. Selanjutnya, tampil halaman daftar pertemuan memuat daftar pertemuan mata kuliah terkait yang dapat dipilih oleh pengguna. Aksi tersebut mengarahkan pengguna menuju halaman detail pertemuan yang berisi opsi modul pembelajaran dan penilaian mahasiswa. Pengguna memilih opsi penilaian mahasiswa yang mengarahkan pengguna pada halaman penilaian mahasiswa yang memuat pilihan



penilaian, dengan salah satu penilaiannya adalah penilaian UTS. Langkah Pengguna berikutnya ialah memilih menu UTS untuk dapat menilai UTS mahasiswa. Pada halaman UTS, pengguna ditampilkan daftar nama mahasiswa yang mengikuti perkuliahan dan pengguna memilih menu aksi pada salah satu mahasiswa. Setelah aksi pengguna tersebut, pengguna ditampilkan suatu *pop up* berupa *form* penilaian yang dapat diisi oleh pengguna. Pengguna dapat menginputkan nilai pada *form*, kemudian menyimpan nilai tersebut. Setelah nilai tersimpan, sistem mengarahkan pengguna menuju halaman daftar nilai mahasiswa.

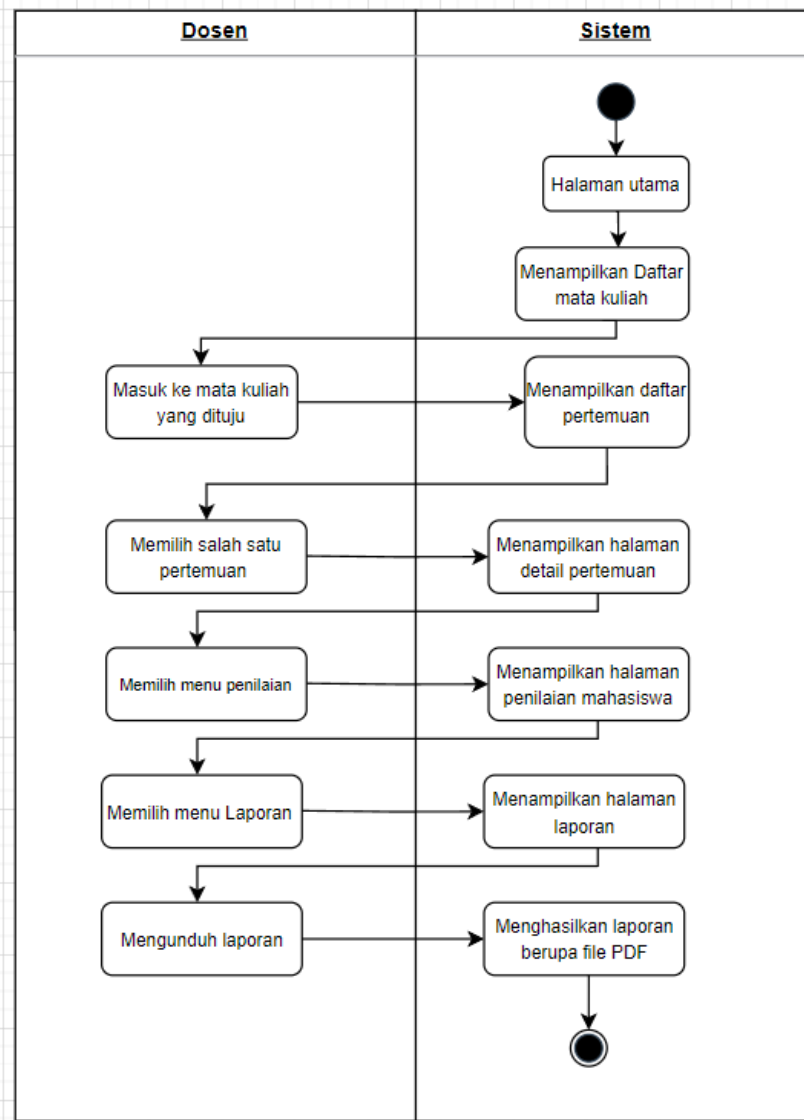
*j. Activity Diagram Penilaian UAS*



Gambar 3.13 Activity Diagram Penilaian UAS

Gambar 3.13 merupakan *activity diagram* proses dosen menilai UAS (Ujian Akhir Semester) mahasiswa. Langkah pertama aktivitas menilai UAS ialah sistem menampilkan halaman utama, pada halaman ini terdapat informasi dan mata kuliah yang diampu oleh pengguna. Pengguna dapat memilih salah tu mata kuliah untuk menuju pada halaman daftar pertemuan. Pada halaman daftar pertemuan terdapat pertemuan mata kuliah terkait yang dapat dipilih oleh pengguna. Setelah pengguna memilih salah satu pertemuan, sistem mengarahkan pengguna menuju halaman detail pertemuan yang tersedia pilihan menu modul dan penilaian mahasiswa. Langkah pengguna berikutnya ialah memilih menu penilaian mahasiswa yang mengarahkan pengguna pada halaman penilaian mahasiswa yang terdiri dari pilihan menu penilaian. Pengguna dapat memilih menu UAS untuk melakukan penilaian UAS. Pada halaman UAS, sistem menampilkan daftar nama mahasiswa yang mengikuti perkuliahan dan setiap nama mahasiswa terdapat tombol aksi. Pengguna menekan tombol aksi yang akan direspon oleh sistem berupa *pop up form* penilaian. Pengguna dapat memasukkan nilai pada *form* dan menyimpan nilai tersebut. Selanjutnya pengguna diarahkan oleh sistem menuju halaman daftar nilai yang telah berisi nilai mahasiswa.

k. Activity Diagram Mengunduh Laporan

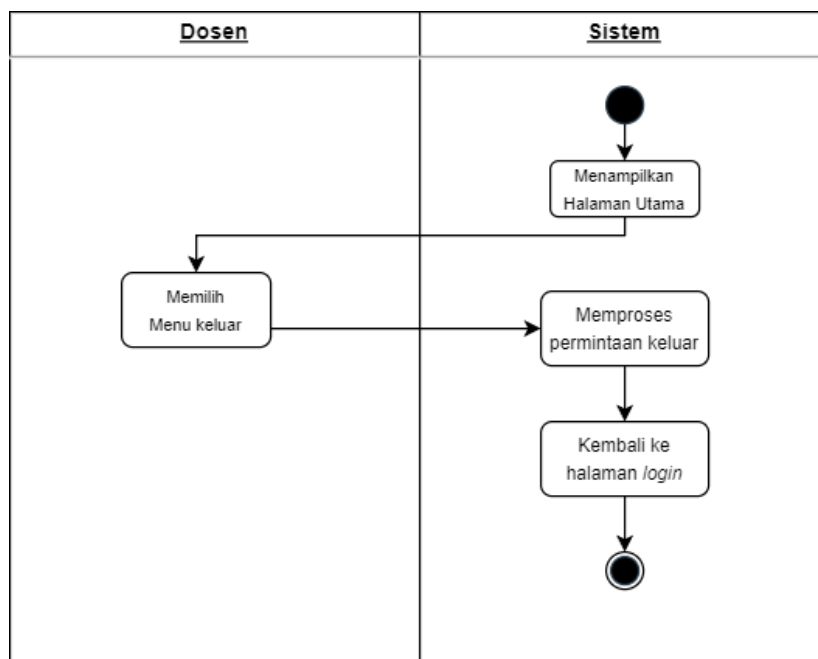


Gambar 3.14 Activity Diagram Mengunduh Laporan

Gambar 3.14 merupakan gambaran *activity diagram* untuk aktivitas dosen pada saat mengunduh laporan penilaian. Alur aktivitas ini diawali dari halaman utama aplikasi yang menampilkan informasi pengguna seperti nama, NIP dan mata kuliah yang diampu. Pengguna dapat memilih mata kuliah yang dituju, lalu sistem menampilkan halaman daftar pertemuan yang memuat pertemuan-pertemuan mata kuliah terkait. Dengan memilih salah pertemuan yang dituju, pengguna diarahkan sistem menuju halaman detail pertemuan. Pengguna dapat memilih menu penilaian untuk melihat pilihan menu penilaian mahasiswa. Aksi tersebut mengarahkan pengguna pada halaman penilaian mahasiswa yang mencakup menu

laporan. Pengguna dapat memilih menu laporan untuk dapat melihat laporan penilaian mahasiswa pada pertemuan tersebut. Pada halaman laporan, pengguna dapat mengunduh daftar laporan tersebut berupa *file* PDF dengan menekan tombol “unduh laporan”.

#### 1. Activity Diagram Keluar Aplikasi

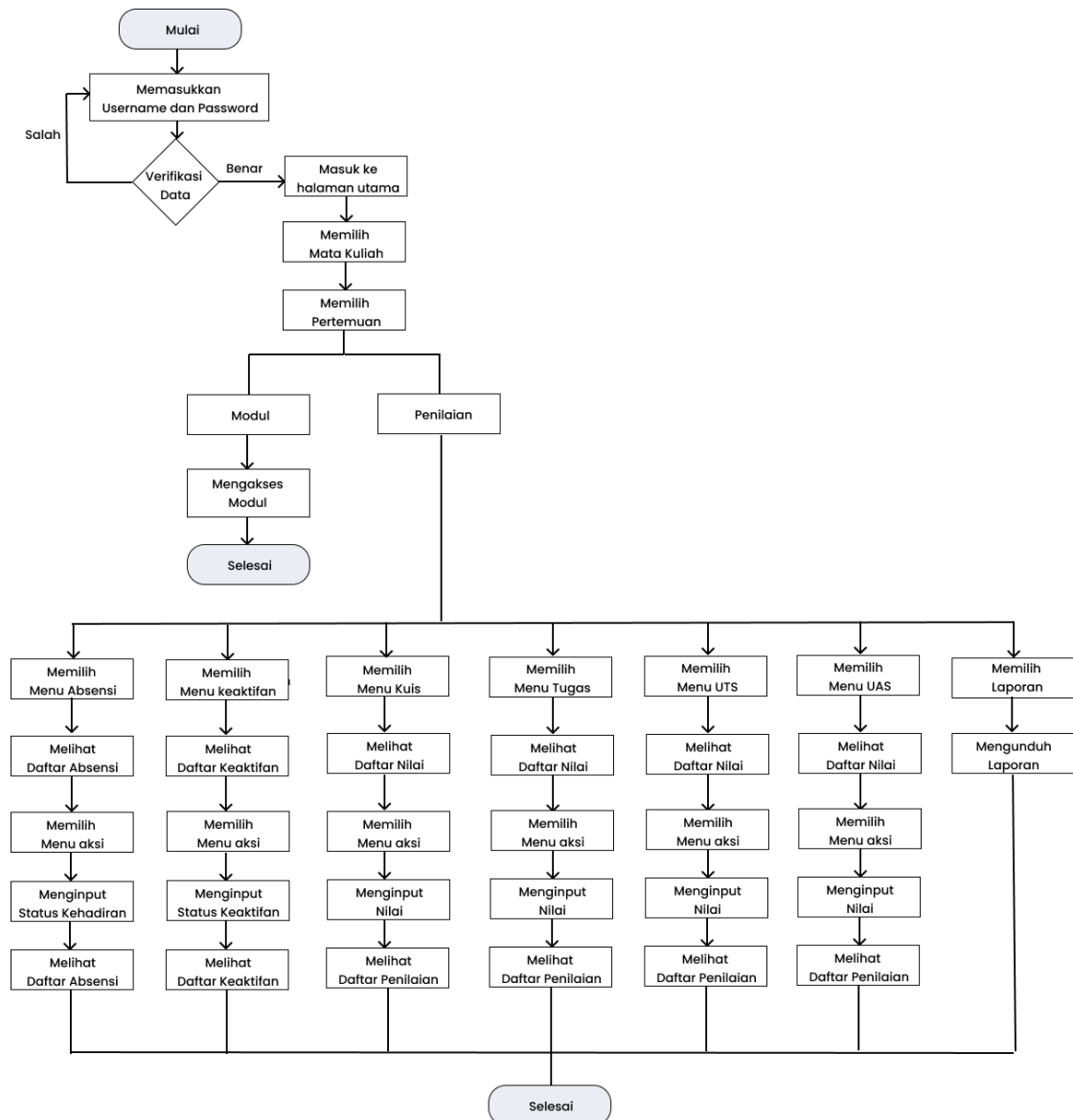


Gambar 3.15 Activity Diagram Keluar Aplikasi

Gambar 3.15 merupakan gambaran dari *activity diagram* dari proses dosen keluar aplikasi. Proses ini dimulai dengan sistem yang menampilkan halaman utama. Pengguna kemudian dapat memilih menu *logout* dan selanjutnya sistem memproses *request logout* pengguna dan mengarahkan pengguna kembali pada halaman *login*.

#### 3.4.2.3 User flow

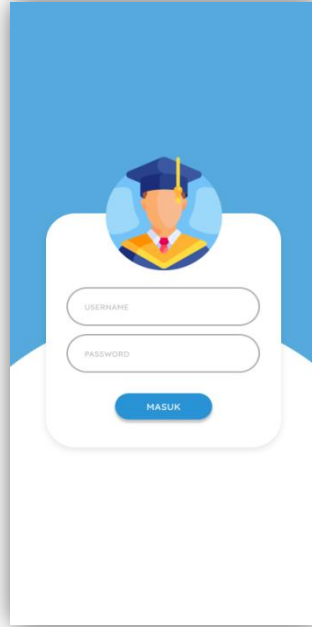
Penggunaan *User flow* pada penelitian ini bertujuan untuk memvisualisasikan alur interaksi antara *user* dengan aplikasi. Gambar 3.14 di bawah ini menggambarkan hasil *user flow* penggunaan aplikasi

Gambar 3.16 *User flow*

Gambar 3.15 merupakan *User flow* yang menggambarkan alur interaksi antara pengguna dengan aplikasi untuk semua kegiatan terkait. Hal ini mencakup langkah-langkah seperti masuk aplikasi, mengakses modul, melakukan absensi, keaktifan, penilaian tugas dan kuis, serta mengunduh laporan kegiatan pada pertemuan tersebut

### 3.4.2.4 Mockup Aplikasi

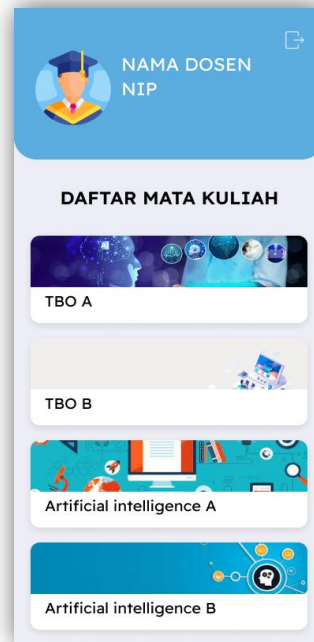
#### a. Halaman *Login*



Gambar 3.17 Halaman *Login*

Gambar 3.17 merupakan *mockup* halaman *login*. Pada halaman ini, pengguna dapat masuk ke aplikasi dengan mengisi *form* yang tersedia. *Form* tersebut berupa inputan *username* dan *password*. Setelah pengguna mengisi data inputan, pengguna dapat masuk ke aplikasi dengan menekan tombol ‘masuk.’ Pengguna dapat masuk apabila data yang diinputkan telah sesuai dengan informasi yang benar dan terdaftar. Ketika pengguna memasukkan data inputan yang salah, maka muncul pop up kesalahan berupa informasi kesalahan data inputan. Hal ini bertujuan untuk memastikan bahwa hanya pengguna yang memiliki akses yang benar-benar dapat masuk ke dalam aplikasi.

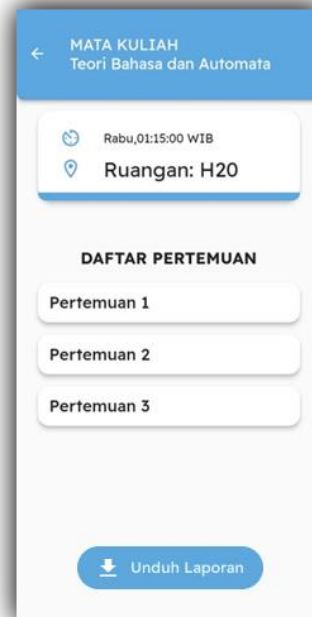
## b. Halaman Utama



Gambar 3.18 Halaman Utama

Gambar 3.18 merupakan *mockup* halaman utama aplikasi. Setelah pengguna masuk ke aplikasi, pengguna diarahkan pada halaman utama aplikasi. Pada halaman ini pengguna dapat melihat informasi-informasi pengguna berupa nama, NIP serta daftar mata kuliah yang diampu oleh pengguna. Pengguna dapat memilih salah satu mata kuliah untuk melihat detail informasi lebih lanjut mengenai mata kuliah tersebut. Selain itu, terdapat juga ikon *logout* yang dapat mengarahkan pengguna keluar dari aplikasi dan kembali menuju halaman *login*.

c. Halaman Pertemuan

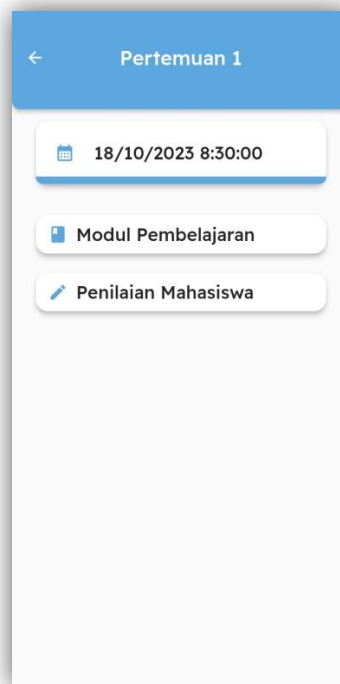


Gambar 3.19 Halaman Pertemuan

Gambar 3.19 merupakan *mockup* halaman pertemuan aplikasi. Halaman ini tampil apabila pengguna telah memilih salah satu mata kuliah pada halaman utama. Pada halaman ini, pengguna ditampilkan informasi berupa jadwal mata kuliah yang memuat waktu dan tempat perkuliahan berlangsung, serta terdapat beberapa daftar pertemuan mata kuliah terkait. Untuk melihat detail informasi terkait pertemuan pada mata kuliah tersebut, pengguna dapat memilih salah satu daftar pertemuan yang mengarahkan pengguna menuju halaman detail pertemuan. Pada halaman ini pengguna dapat mengunduh daftar laporan hasil rekaman semua pertemuan mata kuliah dalam format *file excel* dengan menekan tombol “unduh laporan”.



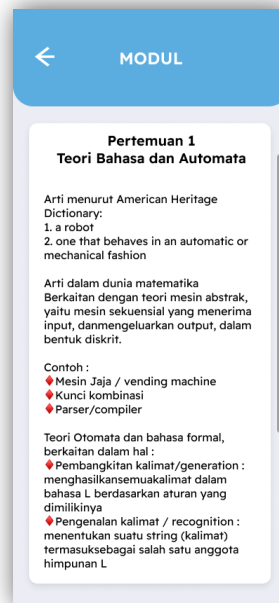
d. Halaman Detail Pertemuan



Gambar 3.20 Halaman Detail Pertemuan

Gambar 3.20 merupakan *mockup* halaman detail pertemuan. Untuk mengakses halaman detail pertemuan, pengguna terlebih dahulu memilih salah satu pertemuan pada halaman daftar pertemuan. Pada halaman detail pertemuan, terdapat dua pilihan menu yang dapat dipilih pengguna yaitu modul pembelajaran dan penilaian mahasiswa. Pengguna dapat memilih salah satu menu yang tersedia untuk melihat informasi yang dituju. Selain itu, pada halaman ini menampilkan informasi terkait pertemuan yaitu jadwal pertemuan, yang mencakup keterangan hari dan tanggal pertemuan.

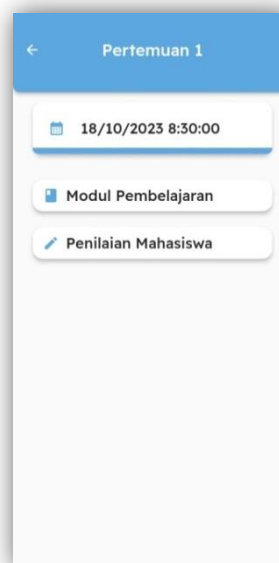
e. Halaman Modul



Gambar 3.21 Halaman Modul

Gambar 3.21 merupakan *mockup* halaman Modul. Pada halaman ini, pengguna ditampilkan modul pembelajaran yang tersedia dan pengguna dapat mengakses modul tersebut sebagai bahan ajar pengguna pada pertemuan tersebut.

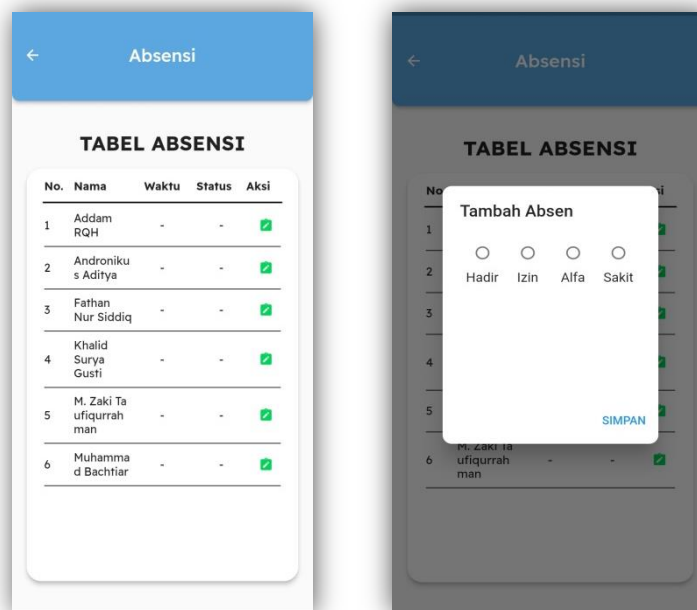
f. Halaman Penilaian Mahasiswa



Gambar 3.22 Halaman Penilaian Mahasiswa

Gambar 3.22 merupakan *mockup* halaman Penilaian Mahasiswa. Halaman ini tampil apabila pengguna memilih menu penilaian mahasiswa pada halaman detail pertemuan. Halaman penilaian mahasiswa berisikan beberapa menu penilaian, seperti absensi, keaktifan, kuis, tugas dan laporan perkuliahan pada pertemuan tersebut. Pengguna dapat memilih menu yang sesuai untuk melihat informasi dan penilaian yang berkaitan dengan kegiatan perkuliahan pada hari tersebut.

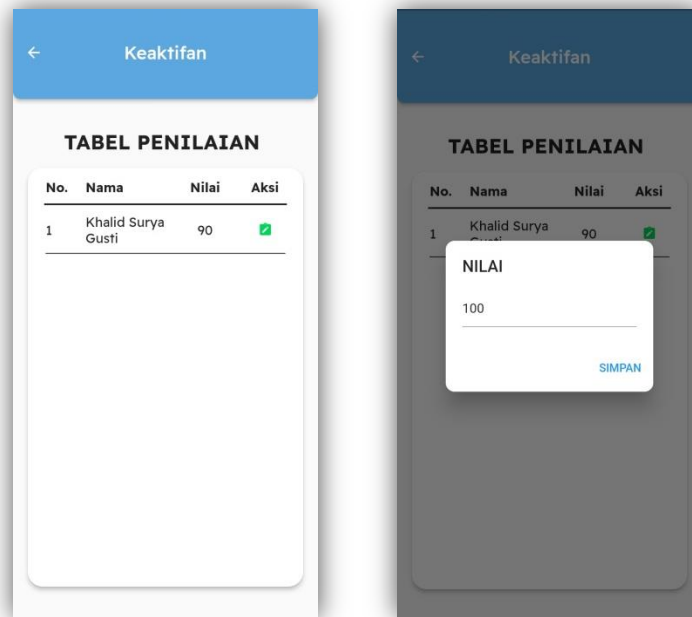
g. Halaman Absensi



Gambar 3.23 Halaman Absensi

Gambar 3.23 merupakan *mockup* Halaman Absensi. Pada halaman absensi, pengguna ditampilkan daftar nama mahasiswa yang mengikuti perkuliahan. Selanjutnya pada setiap nama mahasiswa, terdapat tombol ikon aksi yang dapat digunakan pengguna untuk melakukan absensi untuk mahasiswa tersebut. Aksi pengguna tersebut mengarahkan pengguna pada *pop up* dengan pilihan status kehadiran yaitu Hadir, Sakit, Izin dan Alfa. Pengguna dapat memilih salah satu status dan menyimpan pilihan tersebut. Setelah menyimpan pilihan tersebut, maka pengguna diarahkan kembali pada halaman daftar absensi yang telah berisi informasi mengenai status dan waktu absensi.

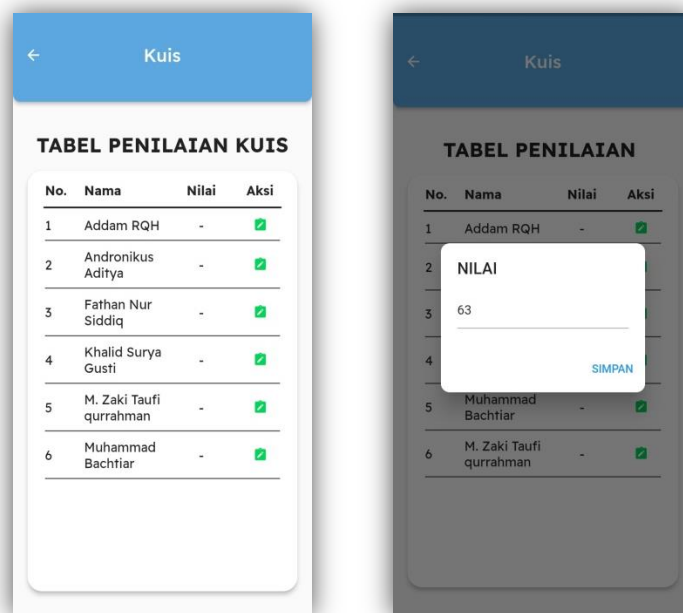
## h. Halaman Keaktifan



Gambar 3.24 Halaman Keaktifan

Gambar 3.24 merupakan *mockup* Halaman Keaktifan. Setelah memilih menu keaktifan pada halaman penilaian mahasiswa. Pengguna diarahkan pada halaman keaktifan yang menampilkan daftar nama mahasiswa yang aktif dalam kegiatan perkuliahan. Untuk menilai keaktifan mahasiswa, pengguna dapat menekan tombol ikon aksi yang kemudian mengarahkan pengguna pada *pop up* berupa formulir nilai yang dapat diinputkan oleh pengguna. Pengguna menginputkan nilai tersebut berdasarkan keaktifan mahasiswa pada pertemuan tersebut. Setelah itu, pengguna dapat menyimpan inputanya dan kembali pada daftar keaktifan mahasiswa yang telah memuat informasi nilai keaktifan mahasiswa pada perkuliahan tersebut.

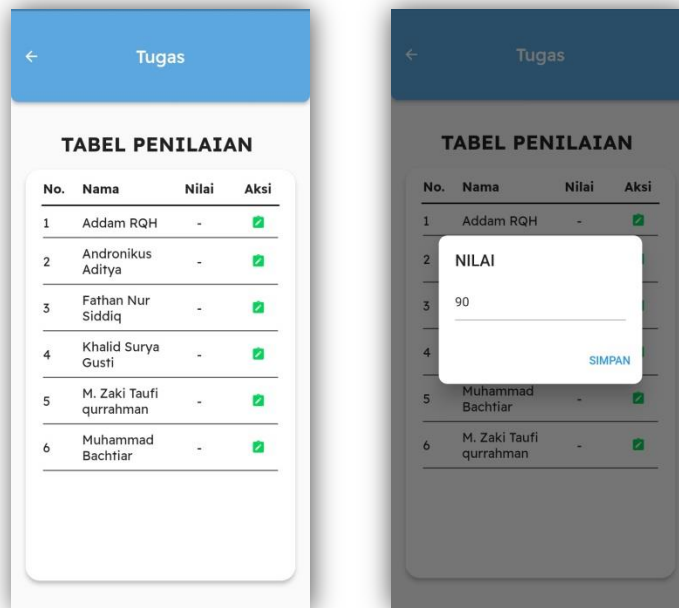
## i. Halaman Kuis



Gambar 3.25 Halaman Kuis

Gambar 3.25 merupakan *mockup* Halaman Kuis. Halaman kuis tampil setelah pengguna memilih menu kuis pada halaman penilaian mahasiswa. Pada halaman kuis, pengguna dapat melakukan penilaian dengan dapat menekan tombol ikon aksi yang terdapat pada daftar nama mahasiswa yang tersedia. Respon terhadap aksi tersebut adalah tampilan *pop up* berupa *form* untuk memasukkan nilai. Pengguna dapat menginputkan nilai pada *form* tersebut berdasarkan hasil kuis mahasiswa pada pertemuan tersebut. Setelah pengguna memasukkan nilai, pengguna dapat menyimpan nilai tersebut dan pengguna diarahkan menuju halaman daftar nilai mahasiswa.

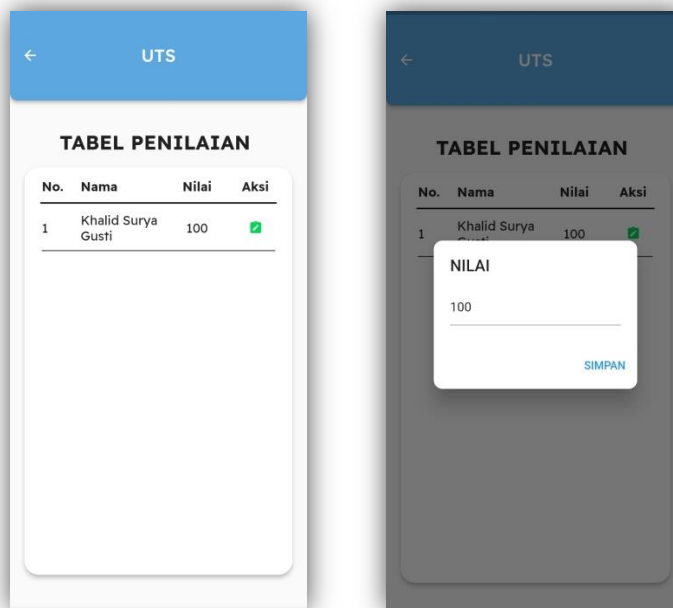
## j. Halaman Tugas



Gambar 3.26 Halaman Tugas

Gambar 3.26 merupakan *mockup* Halaman Tugas. Pengguna dapat melihat halaman tugas apabila memilih menu tugas pada halaman penilaian mahasiswa. Pada halaman ini, pengguna ditampilkan daftar nama mahasiswa yang mengikuti perkuliahan. Untuk melakukan penilaian tugas, pengguna dapat menekan tombol ikon aksi yang kemudian muncul jendela *pop up* berupa *form* untuk memasukkan nilai. Pengguna dapat menginputkan nilai pada *form* yang tersedia dan menyimpan nilai tersebut. Setelah itu, pengguna kembali ke halaman daftar nilai tugas mahasiswa yang telah memuat informasi nilai tugas mahasiswa pada pada pertemuan tersebut.

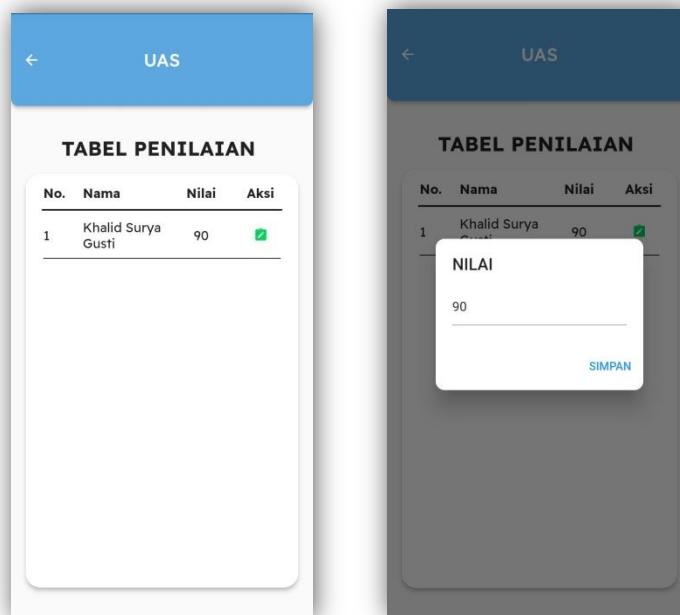
## k. Halaman UTS



Gambar 3.27 Halaman UTS

Gambar 3.27 merupakan *mockup* Halaman UTS (Ujian Tengah Semester).. Halaman UTS tampil hanya pada saat pengguna memilih pertemuan delapan pada halaman daftar pertemuan. Saat pengguna memilih menu UTS di halaman Penilaian, pengguna dapat melihat daftar nama mahasiswa yang mengikuti perkuliahan. Untuk dapat melakukan penilaian UTS, pengguna dapat menekan tombol ikon aksi, yang memunculkan jendela *pop up* berupa *form* yang dapat dimasukkan nilai oleh pengguna. Nilai dapat dimasukkan dan disimpan oleh pengguna. Setelah itu, pengguna kembali pada halaman daftar nilai UTS mahasiswa yang telah berisikan informasi nilai UTS mahasiswa.

## 1. Halaman UAS

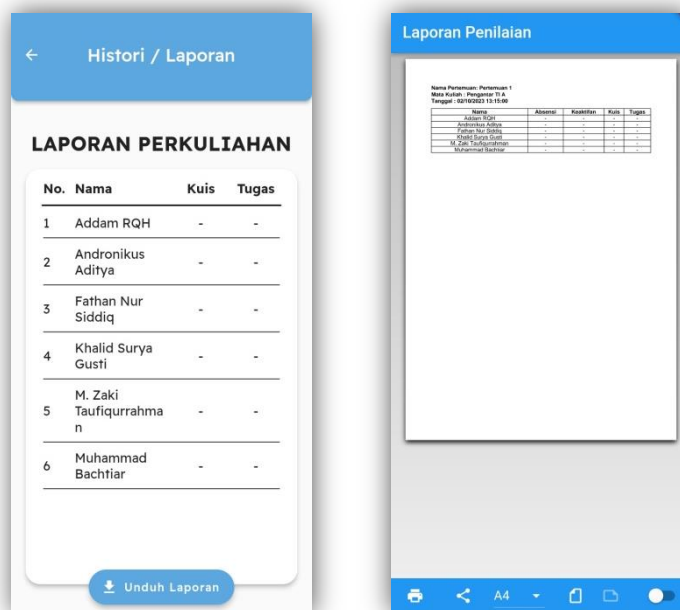


Gambar 3.28 Halaman UAS

Gambar 3.28 merupakan *mockup* Halaman UAS (Ujian Akhir Semester).. Agar dapat mengakses dan melakukan penilaian di halaman UAS, pengguna perlu memilih pertemuan 16 pada daftar pertemuan. Saat pengguna masuk pada halaman UAS, pengguna ditampilkan daftar nama mahasiswa yang mengikuti perkuliahan tersebut. Pengguna dapat menekan tombol ikon aksi yang terdapat pada setiap mahasiswa untuk melakukan penilaian. Tombol aksi ini membuka jendela *pop up* yang menampilkan *form* penilaian. Pengguna dapat menginputkan nilai pada *form* yang tersedia dan menyimpan nilai tersebut. Setelah nilai disimpan, pengguna diarahkan kembali menuju halaman daftar nilai UAS mahasiswa yang telah memuat nilai UAS mahasiswa tersebut.



### m. Halaman Laporan



Gambar 3.29 Halaman Laporan

Gambar 3.29 merupakan *mockup* halaman laporan. Untuk melihat informasi laporan penilaian mahasiswa, pengguna terlebih dahulu memilih menu laporan pada halaman penilaian mahasiswa. Halaman ini menampilkan laporan penilaian harian mahasiswa, yang berisikan laporan penilaian mahasiswa termasuk hasil penilaian tugas, kuis, dan aktivitas akademik lainnya yang dilakukan oleh mahasiswa dalam pertemuan tersebut. Pada halaman laporan, pengguna dapat mengunduh daftar laporan tersebut dalam format *file* PDF dengan menekan tombol “unduh laporan”.

#### 3.4.3 Construction

Pada tahapan *Construction* atau konstruksi, pengembang membangun dan mengembangkan sistem menjadi program. Pada tahap ini hasil desain dimasukkan ke dalam bentuk bahasa pemrograman yang digunakan agar dapat dijalankan dalam bentuk aplikasi. Pada penelitian ini aplikasi dikembangkan dengan berbasis *mobile* menggunakan bahasa pemrograman android yaitu Dart dan juga menggunakan *framework* Flutter.

### 3.4.3.1 Testing

#### a. *Blackbox Testing*

Penggunaan *Blackbox Testing* pada penelitian bertujuan untuk mengetahui bagaimana kinerja fungsionalitas dari aplikasi yang dikembangkan pada penelitian ini. Pengujian dilakukan dengan menjalankan aplikasi dan melihat keluarannya (*output*), rancangan pengujian yang dilakukan didasarkan pada kesesuaian antara fitur dan fungsionalitas yang diharapkan oleh pengguna (dosen) dengan *testcase* atau skenario pengujian yang telah dirancang sebelumnya. Setelah itu, dilakukan pengecekan apakah aplikasi telah sesuai dengan yang diharapkan.

#### b. *System Usability Scale (SUS)*

Penggunaan *System Usability Scale (SUS)* pada penelitian ini bertujuan untuk mengetahui kepuasan pengguna terhadap sistem yang sudah dibuat, Responden dalam pengujian ini adalah dosen yang menjadi pengguna utama dari aplikasi. Target dari pengujian adalah memastikan bahwa antarmuka, fungsi dan kinerja aplikasi sesuai dengan kebutuhan dosen. Pengujian ini memberikan pandangan langsung dari dosen tentang kebutuhan non fungsional yang harus dipenuhi agar aplikasi dapat memberikan manfaat maksimal dalam mendukung aktivitas. Pengujian ini mencakup berbagai aspek, seperti *usability*, *performance*, dan *memorability*.

1. *Usability* : Pada tahap ini, aplikasi yang dibuat selanjutnya diuji terkait dengan sejauh mana kemudahan penggunaan dan interaksi dengan antarmuka pengguna.
2. *Performance* : Pada tahap ini dilakukan pengukuran kinerja aplikasi pada situasi yang berbeda-beda. Hal ini melibatkan pengujian kemampuan aplikasi untuk menangani beban kerja yang tinggi, respon waktu, serta stabilitas aplikasi ketika digunakan.
3. *Memorability* : Pada tahap ini, aplikasi diuji dengan mengukur seberapa mudah dosen dapat mengingat alur dan cara menggunakan aplikasi setelah beberapa waktu tidak menggunakannya.

#### **3.4.4 Cutover**

Pada tahapan *Cutover* atau implementasi, aplikasi yang telah dibuat selanjutnya diluncurkan dan dikeluarkan. Sebelum aplikasi diluncurkan, dilakukan evaluasi aplikasi dengan tujuan untuk menemukan kesalahan pada sistem dan mendapatkan umpan balik dari pengguna. Jika sistem belum sesuai, maka proses kembali pada tahapan pengembangan untuk dilakukan pengembangan kembali atau untuk diperbaiki oleh pengembang sebelum aplikasi diliris kembali.

## V. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Berdasarkan hasil dari penelitian yang telah dilakukan, terdapat kesimpulan yang dapat diambil diantaranya :

1. Penelitian telah berhasil membuat tahapan awal sebuah aplikasi menggunakan *framework* flutter yang dapat membantu dosen dalam proses perkuliahan yang terdigitalisasi, yang memiliki fitur absensi, keaktifan, kuis, tugas, UTS, UAS dan laporan penilaian.
2. Berdasarkan hasil dua kali pengujian menggunakan *Blackbox testing* terhadap fitur dan fungsionalitas aplikasi pada iterasi pertama dengan 10 skenario pengujian dan iterasi kedua dengan 3 skenario pengujian menunjukkan bahwa fitur-fitur yang dikembangkan dapat berjalan sesuai dengan yang diharapkan.
3. Berdasarkan hasil dua kali pengujian menggunakan SUS bersama 4 responden, diperoleh kesimpulan bahwa pada pengujian pertama, perhitungan nilai mencapai skor 70. Skor ini masuk dalam kategori *Acceptable* pada penilaian *Acceptability Range*, kategori OK pada penilaian *Adjective Range*, dan memperoleh predikat C pada penilaian *Grade Scale*. Selanjutnya pada pengujian kedua, nilai SUS meningkat menjadi 81.87, yang masuk dalam kategori *Acceptable* pada penilaian *Acceptability Range*, naik ke kategori *GOOD* pada penilaian *Adjective Range*, dan memperoleh predikat B pada penilaian *Grade Scale*.

### 5.2 Saran

Berdasarkan hasil dari penelitian yang telah dilakukan, terdapat beberapa saran untuk pengembangan selanjutnya, antara lain :

1. Melakukan pengembangan aplikasi lebih lanjut dengan menambahkan fitur sistem cerdas berupa *face recognition*, guna menciptakan sistem yang lebih akurat dan valid kepada dosen saat melakukan perkuliahan tatap muka.

2. Melakukan integrasi dengan akun SSO Unila, agar memberikan kemudahan akses kepada dosen, sehingga tidak perlu mengingat banyak nama pengguna dan kata sandi.

### DAFTAR PUSTAKA

- [1] R. D. Djojodibroto, Tradisi Kehidupan Akademik, 1st ed. Yogyakarta: Galang Press, 2004. [Online]. Available: [https://www.google.co.id/books/edition/Tradisi\\_kehidupan\\_akademik/DheS01j3X5oC?hl=id&gbpv=1&dq=Perkuliahan+adalah&pg=PA38&printsec=frontcover](https://www.google.co.id/books/edition/Tradisi_kehidupan_akademik/DheS01j3X5oC?hl=id&gbpv=1&dq=Perkuliahan+adalah&pg=PA38&printsec=frontcover).
- [2] Y. S. Wulandari, Tracer Study Unika Soegijapranata 2019. Semarang: SCU Knowledge Media, 2019. [Online]. Available: [https://www.google.co.id/books/edition/Tracer\\_Study\\_Unika\\_Soegijapranata\\_2019/EtcCEAAAQBAJ?hl=id&gbpv=1&dq=Perkuliahan+adalah&pg=PA65&printsec=frontcover](https://www.google.co.id/books/edition/Tracer_Study_Unika_Soegijapranata_2019/EtcCEAAAQBAJ?hl=id&gbpv=1&dq=Perkuliahan+adalah&pg=PA65&printsec=frontcover)
- [3] M. Amstrong, Pengembangan Sumber Daya Manusia: Handbook Manajemen SDM. Bandung: Nusamedia, 2021. [Online]. Available: [https://www.google.co.id/books/edition/Pengembangan\\_Sumber\\_Daya\\_Manusia\\_Handboo/ZLJsEAAAQBAJ?hl=id&gbpv=1](https://www.google.co.id/books/edition/Pengembangan_Sumber_Daya_Manusia_Handboo/ZLJsEAAAQBAJ?hl=id&gbpv=1)
- [4] M. M. Dr. Herningsih., Kinerja Dosen & Karyawan. Media Nusa Creative (MNC Publishing), 2022. [Online]. Available: [https://www.google.co.id/books/edition/Kinerja\\_Dosen\\_Karyawan/gFSeEAAAQBAJ?hl=id&gbpv=0](https://www.google.co.id/books/edition/Kinerja_Dosen_Karyawan/gFSeEAAAQBAJ?hl=id&gbpv=0).
- [5] M. Dr. Rufman Iman Akbar, Denny Ganjar Purnama., S.Si., M.T.I., Slamet Riyanto., S.T., Importance - Performance Analysis Matrix Penerapan untuk Penilaian Situs Pembelajaran SIAkad dan LMS. Makassar: Nas Media Pustaka, 2022. [Online]. Available: [https://www.google.co.id/books/edition/Importance\\_Performance\\_Analysis\\_Matrix/boOeEAAAQBAJ?hl=id&gbpv=0](https://www.google.co.id/books/edition/Importance_Performance_Analysis_Matrix/boOeEAAAQBAJ?hl=id&gbpv=0)
- [6] W. S. · I Gede Iwan Sudipa, Rakhmadi Rahman, Moch. Fauzi, Fien Pongpalilu, Zunan Setiawan, Miftakhul Huda, Aniek Suryanti Kusuma, Desak Made Dwi Utami Putra, Muhammad Ikhwan Burhan, Yunita Magrima Anzani, Siva Devi Azahra, PENERAPAN SISTEM INFORMASI DI BERBAGAI BIDANG. PT. Sonpedia Publishing Indonesia, 2023. [Online]. Available: [https://www.google.co.id/books/edition/PENERAPAN\\_SISTEM\\_INFOR](https://www.google.co.id/books/edition/PENERAPAN_SISTEM_INFOR)

MASI\_DI\_BERBAGAI\_B/xtKvEAAAQBAJ?hl=id&gbpv=1&dq=SIAKA  
D+adalah&pg=PA73&printsec=frontcover

- [7] M. P. Muh. Faisal Halim, S.Sos.I., Pelayanan Perubahan Data Pada Sistem Informasi Akademik Di Perguruan Tinggi. Bening Media Publishing. [Online]. Available: [https://www.google.co.id/books/edition/Pelayanan\\_Perubahan\\_Data\\_Pada\\_Sistem\\_Inf/EemnEAAAQBAJ?hl=id&gbpv=0](https://www.google.co.id/books/edition/Pelayanan_Perubahan_Data_Pada_Sistem_Inf/EemnEAAAQBAJ?hl=id&gbpv=0)
- [8] B. Sundari, “Sistem Business Intelligence Perguruan Tinggi Sebagai Pendukung Keputusan Akademik Universitas Lampung,” 2023.
- [9] Reporter Unila, “Unila Terapkan Uji Coba Sistem Informasi Akademik Versi 6.0,” 2019. <https://www.unila.ac.id/unila-terapkan-uji-coba-sistem-informasi-akademik-versi-6-0/>
- [10] Ismail, Efitra, Pengantar Framework Populer Mobile Apps. Jakarta : PT. Sonpedia Publishing Indonesia, 2023. [Online]. Available: [https://www.google.co.id/books/edition/Pengantar\\_Framework\\_Populer\\_Mobile\\_Apps/VSW4EAAAQBAJ?hl=id&gbpv=1](https://www.google.co.id/books/edition/Pengantar_Framework_Populer_Mobile_Apps/VSW4EAAAQBAJ?hl=id&gbpv=1)
- [11] M. . Prof. Dr. Ir. Hasriadi Mat Akin, Profil Umum Universitas lampung 2019. Lampung: BIRO PERENCANAAN DAN HUBUNGAN MASYARAKAT UNIVERSITAS LAMPUNG, 2019. [Online]. Available: <https://bphm.unila.ac.id/wp-content/uploads/2020/01/PROFIL-UMUM-UNILAweb-bphm.pdf>
- [12] Herlinah, S.Kom, M,Si., Pemrograman Aplikasi Android dengan Android Studio, Photoshop, dan Audition. Jakarta: Elex Media Komputindo, 2019. [Online]. Available: [https://www.google.co.id/books/edition/Pemrograman\\_Aplikasi\\_Android\\_dengan\\_Andr/pEyrDwAAQBAJ?hl=id&gbpv=0](https://www.google.co.id/books/edition/Pemrograman_Aplikasi_Android_dengan_Andr/pEyrDwAAQBAJ?hl=id&gbpv=0)
- [13] U. Ardiati , Rekayasa Perangkat Lunak Berorientasi Objek Menggunakan PHP. Daerah Istimewa Yogyakarta: Andi 2021. [Online]. Available: [https://www.google.co.id/books/edition/Rekayasa\\_Perangkat\\_Lunak\\_Beroirentasi\\_Ob/x8xEEAAAQBAJ?hl=id&gbpv=1](https://www.google.co.id/books/edition/Rekayasa_Perangkat_Lunak_Beroirentasi_Ob/x8xEEAAAQBAJ?hl=id&gbpv=1)
- [14] S. Mulyani , Metode Analisis dan Perancangan. Bandung: Abdi Sistematika 2016.[Online]. Available: [https://www.google.co.id/books/edition/Metode\\_Analisis\\_dan\\_Perancangan\\_Sistem/SbrPDgAAQBAJ?hl=id&gbpv=1](https://www.google.co.id/books/edition/Metode_Analisis_dan_Perancangan_Sistem/SbrPDgAAQBAJ?hl=id&gbpv=1)

- [15] R. Destriana, Diagram UML Dalam Membuat Aplikasi Android Firebase Sleman: CV. Budi Utama, 2021. [Online]. Available: [https://www.google.co.id/books/edition/Diagram\\_UML\\_Dalam\\_Membuat\\_Aplikasi\\_Andro/vmtYEAAAQBAJ?hl=id&gbpv=0](https://www.google.co.id/books/edition/Diagram_UML_Dalam_Membuat_Aplikasi_Andro/vmtYEAAAQBAJ?hl=id&gbpv=0)
- [16] V. Tasril, S.Kom., M.Kom, A. Rizka, S.Kom.,M.Kom, Munisa, S.Psi.,M.Psi, S. D. Murni ., Kecanduan Gadget Dengan Pendekatan Lean Ux. Sumatera Barat: CV. Azka Pustaka, 2023. [Online]. Available: [https://www.google.co.id/books/edition/KECANDUAN\\_GADGET\\_Dengan\\_Pendekatan\\_Lean/Q1LAEAAAQBAJ?hl=id&gbpv=0](https://www.google.co.id/books/edition/KECANDUAN_GADGET_Dengan_Pendekatan_Lean/Q1LAEAAAQBAJ?hl=id&gbpv=0).
- [17] M. Wali, Sepriano, T. A. Nengsih, , Pengantar 15 Bahasa Pemograman Terbaik Di Masa Depan (Referensi & Coding Untuk Pemula). Jambi: PT. Sonpedia Publishing Indonesia 2023. [Online]. Available: [https://www.google.co.id/books/edition/Pengantar\\_15\\_Bahasa\\_Pemograman\\_Terbaik/uIWxEAAAQBAJ?hl=id&gbpv=0](https://www.google.co.id/books/edition/Pengantar_15_Bahasa_Pemograman_Terbaik/uIWxEAAAQBAJ?hl=id&gbpv=0).
- [18] Ridwan and Bustami., Konsep dan Perancangan Aplikasi: Membangun Aplikasi Mobile Menggunakan Flutter. Surabaya: Syah Kuala Univeristy Press,2021.[Online].Available:[https://www.google.co.id/books/edition/Konsep\\_dan\\_Perancangan\\_Aplikasi\\_Membangu/qBFhEAAAQBAJ?hl=id&gbpv=0](https://www.google.co.id/books/edition/Konsep_dan_Perancangan_Aplikasi_Membangu/qBFhEAAAQBAJ?hl=id&gbpv=0)
- [19] M. K. Dedy Rahman Prehanto, S.Kom., Buku Ajar Konsep Sistem Informasi. Surabaya: Scopindo Media Pustaka, 2020. [Online]. Available:[https://www.google.co.id/books/edition/Buku\\_Ajar\\_Konsep\\_Sistem\\_Informasi/0OriDwAAQBAJ?hl=id&gbpv=0](https://www.google.co.id/books/edition/Buku_Ajar_Konsep_Sistem_Informasi/0OriDwAAQBAJ?hl=id&gbpv=0)
- [20] Kendall, Kenneth E, Analisis dan perancangan sistem. Jakarta: Indeks, 2010.[Online].Available:[http://perpustakaan.umsida.ac.id//index.php?p=show\\_detail&id=22313](http://perpustakaan.umsida.ac.id//index.php?p=show_detail&id=22313)
- [21] D. A. zein M.Kom et al., Konsep Dasar Rekaya Perangkat Lunak. Batam: yayasan cendekia mulia mandiri, 2023. [Online]. Available: [https://www.google.co.id/books/edition/Konsep\\_Dasar\\_Rekayasa\\_Perangkat\\_Lunak/7Om6EAAAQBAJ?hl=id&gbpv=1&dq=tahapan+RAD&pg=PA95&printsec=frontcover](https://www.google.co.id/books/edition/Konsep_Dasar_Rekayasa_Perangkat_Lunak/7Om6EAAAQBAJ?hl=id&gbpv=1&dq=tahapan+RAD&pg=PA95&printsec=frontcover)
- [22] A. Nordeen, Learn Software Testing in 24 Hours. New York: Guru99, 2020.[Online].Available:[https://www.google.co.id/books/edition/Learn\\_So](https://www.google.co.id/books/edition/Learn_So)



ftware\_Testing\_in\_24\_Hours/hRwGEAAAQBAJ?hl=id&gbpv=0.

- [23] T. Wahyuningrom, *Buku Referensi Mengukur Usability Perangkat Lunak*. Sleman: CV. Budi Utama, 2021. [Online]. Available: [https://www.google.co.id/books/edition/Buku\\_Rferensi\\_Mengukur\\_Usability\\_Perang/Pzk9EAAAQBAJ?hl=id&gbpv=0](https://www.google.co.id/books/edition/Buku_Rferensi_Mengukur_Usability_Perang/Pzk9EAAAQBAJ?hl=id&gbpv=0)
- [24] H. Muliandari and H. Setiaji, "Pengembangan Dashboard Sistem Informasi Manajemen Perkuliahan (Studi Kasus: Fakultas Teknologi Industri)," *Automata*, 2021, [Online]. Available: <https://journal.uui.ac.id/-AUTOMATA/article/view/19433>
- [25] A. R. Dhuha, F. Pradana, and B. Priyambadha, "Pengembangan Sistem Aplikasi Manajemen Proyek Berbasis Web (Studi Kasus : PT . Swadaya Graha)," *Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 11, pp. 1367–1375, 2018.
- [26] G. Manu and H. Tugil, "Perancangan Sistem Informasi Manajemen Ekstrakurikuler (Sime) Berbasis Web," *J. Pendidik. Teknol. Inf.*, vol. 3, no. 1, pp. 14–20, 2020, doi: 10.37792/jukanti.v3i1.91.
- [27] M. Y. Putra and D. E. Kurniawan, "Implementasi Sistem Reminder Jadwal pada eLearning Moodle Berbasis API Menggunakan Framework Flutter," *J. Appl. Comput. Sci. Technol.*, vol. 4, no. 1, pp. 7–11, 2023, doi: 10.52158/jacost.v4i1.490.
- [28] W. G. T. Prasetyo, F. Pradana, and B. S. Prakoso, "Pengembangan Aplikasi Point of Sales Warung dan UMKM 'WarunkQu' menggunakan Framework Flutter," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 10, pp. 4724–4730, 2022, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [29] F. Ahmad and H. Eko, "Rancang Bangun Sistem Informasi Akademik SMK Negeri 1 Depok Berbasis Android dengan Pendekatan Rapid Application Development," *J. Teknol. Inf.*, vol. 12, no. 2, pp. 129-139, 2019, doi: <http://dx.doi.org/10.15408/jti.v12i2.10939>