

BAB IV

HASIL DAN PEMBAHASAN

4.1 Analisa Sistem (*System Analysis*)

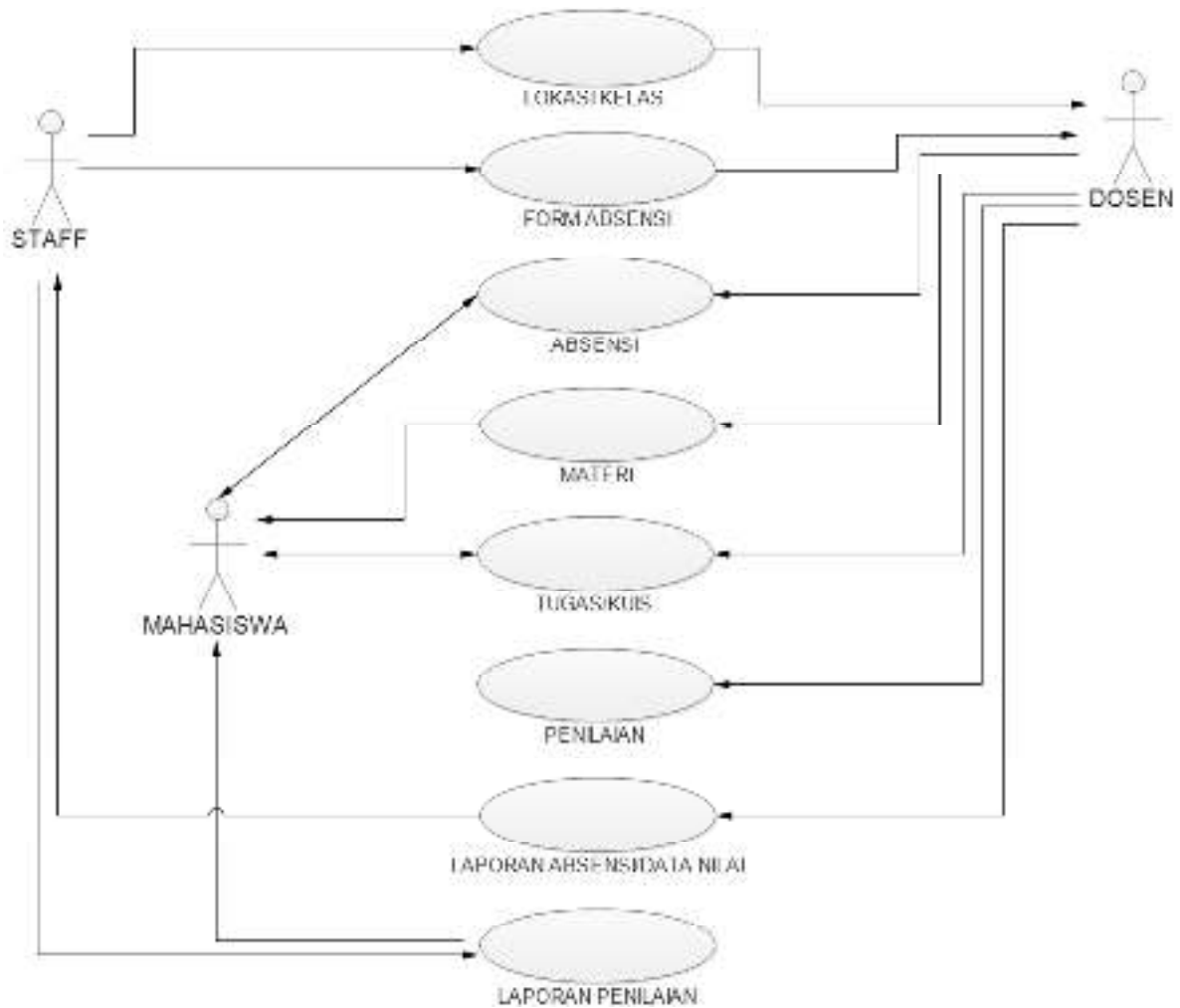
Dalam bab empat ini, dilakukan pembahasan yang mengacu pada identifikasi masalah yang ada. Permasalahan yang diangkat dalam penulisan skripsi ini adalah bagaimana membuat sistem informasi controlling sertifikasi awak pesawat yang bisa memudahkan staff controlling dalam mengontrol sertifikasi para aircrew.

4.1.1 Analisa Sistem Yang Berjalan

STMIK Muhammadiyah Jakarta menerapkan metode konvensional yaitu tatap muka dalam sistem perkuliahannya. Seluruh proses kegiatan belajar dilakukan oleh 2(dua) aktor yaitu Dosen dan Mahasiswa di dalam ruangan kelas, dengan Admin sebagai penentu lokasi kelas.

Berikut adalah sistem perkuliahan STMIK Muhammadiyah Jakarta yang disajikan dengan menggunakan *UML model* :

4.1.1.1 Use Case Diagram Sistem Perkuliahan



Gambar 4.1 Use Case Diagram Sistem Perkuliahan
STMIK Muhammadiyah Jakarta

Gambar 4.1 menjelaskan tentang peran setiap aktor terhadap sistem yang berjalan. Berikut penjelasannya:

Deskripsi Use Case (Lokasi Kelas)

<p>Aktor Utama: Staff</p> <p>Aktor Tambahan: Dosen, Mahasiswa</p> <p>Nama Use Case: Lokasi Kelas</p> <p>Tujuan Use Case: Menentukan kelas mana yang dapat digunakan untuk tiap sesi mata kuliah dan menginfokan kepada dosen dan mahasiswa agar menggunakan kelas yang ditentukan.</p> <p>Kondisi Awal: Lokasi kelas belum ditentukan. Mahasiswa belum mengetahui kelas mana yang harus dihadiri.</p> <p>Kondisi Akhir: Dosen dan mahasiswa melakukan perkuliahan pada kelas yang telah ditentukan.</p>	<p>Optimistic Flow:</p> <ol style="list-style-type: none"> 1. Staff melakukan survey terhadap kelas mana yang dapat digunakan untuk sebuah sesi mata kuliah. 2. Staff mengambil keputusan berdasarkan status pada tiap kelas. 3. Staff menginfokan kepada dosen dan mahasiswa bersangkutan untuk menggunakan kelas yang ditentukan. 4. Dosen dan mahasiswa melakukan perkuliahan pada kelas tersebut. <p>Pesimistic Flow:</p> <ol style="list-style-type: none"> 1. Tidak ada kelas kosong tersedia atau ada kendala pada ruangan kelas yang tersedia.
---	---

Deskripsi Use Case (Form Absensi)

<p>Aktor Utama: Staff</p> <p>Aktor Tambahan: Dosen</p> <p>Nama Use Case: Form Absensi</p> <p>Tujuan Use Case: Menentukan kelas mana yang dapat digunakan untuk tiap sesi mata kuliah dan menginfokan kepada dosen dan mahasiswa agar menggunakan kelas yang ditentukan.</p> <p>Kondisi Awal: Data form absensi belum terisi.</p> <p>Kondisi Akhir: Data form absensi sudah terisi.</p>	<p>Optimistic Flow:</p> <ol style="list-style-type: none"> 1. Staff memberikan form absensi kepada dosen. 2. Dosen mengisi informasi umum pada form seperti tanggal, bulan, tahun, dan mata kuliah. Dosen juga melakukan tanda tangan pada form absensi. 3. Dosen membawa form tersebut ke dalam kelas sebagai media pengisian data kehadiran mahasiswa.
--	--

Deskripsi Use Case (Absensi)

<p>Aktor Utama: Dosen, Mahasiswa</p> <p>Aktor Tambahan: -</p> <p>Nama Use Case: Lokasi Absensi</p> <p>Tujuan Use Case: Melakukan proses absensi demi mendapatkan data kehadiran mahasiswa.</p> <p>Kondisi Awal: Dosen menyebutkan nama mahasiswa satu per satu.</p> <p>Kondisi Akhir: Mahasiswa memberikan <i>feedback</i> kepada dosen menunjukkan kehadirannya.</p>	<p>Optimistic Flow:</p> <ol style="list-style-type: none"> 1. Staff memberikan form absensi kepada dosen. 2. Dosen mengisi informasi umum pada form seperti tanggal, bulan, tahun, dan mata kuliah. Dosen juga melakukan tanda tangan pada form absensi. 3. Dosen membawa form tersebut ke dalam kelas sebagai media pengisian data kehadiran mahasiswa.
---	--

Deskripsi Use Case (Materi)

<p>Aktor Utama:</p> <p>Dosen</p> <p>Aktor Tambahan:</p> <p>Mahasiswa</p> <p>Nama Use Case:</p> <p>Materi</p> <p>Tujuan Use Case:</p> <p>Melakukan penyampaian materi kuliah oleh dosen kepada mahasiswa.</p> <p>Kondisi Awal:</p> <p>Adanya proses pembukaan mata kuliah, dilanjutkan dengan proses penyampaian materi.</p> <p>Kondisi Akhir:</p> <p>Adanya proses Tanya jawab, diskusi kelompok dan penjelasan mengenai pertanyaan yang diajukan.</p>	<p>Optimistic Flow:</p> <ol style="list-style-type: none"> 1. Staff memberikan form absensi kepada dosen. 2. Dosen mengisi informasi umum pada form seperti tanggal, bulan, tahun, dan mata kuliah. Dosen juga melakukan tanda tangan pada form absensi. 3. Dosen membawa form tersebut ke dalam kelas sebagai media pengisian data kehadiran mahasiswa.
--	--

Deskripsi Use Case (Tugas/Kuis)

<p>Aktor Utama: Dosen, Mahasiswa</p> <p>Aktor Tambahan: -</p> <p>Nama Use Case: Tugas/Kuis</p> <p>Tujuan Use Case: Dosen memberikan tugas maupun kuis kepada mahasiswa, dan mahasiswa memberikan <i>feedback</i> terhadap apa yang diberikan.</p> <p>Kondisi Awal: Dosen memberikan tugas atau kuis kepada mahasiswa.</p> <p>Kondisi Akhir: Mahasiswa menyerahkan tugas atau menjawab kuis yang telah diberikan.</p>	<p>Optimistic Flow:</p> <ol style="list-style-type: none"> 1. Dosen memberikan tugas maupun kuis kepada mahasiswa berdasarkan materi yang telah disampaikan. 2. Mahasiswa mengerjakan atau menjawab tugas/kuis. 3. Mahasiswa menyerahkan kembali tugas/kuis yang telah dikerjakan sebagai bahan penilaian.
--	--

Deskripsi Use Case (Penilaian)

<p>Aktor Utama: Dosen</p> <p>Aktor Tambahan: -</p> <p>Nama Use Case: Penilaian</p> <p>Tujuan Use Case: Mengarsipkan data nilai mahasiswa berdasarkan absensi, tugas, dan beberapa aspek penilaian lainnya.</p> <p>Kondisi Awal: Dosen melakukan penilaian kepada mahasiswa.</p> <p>Kondisi Akhir: Data nilai diarsipkan.</p>	<p>Optimistic Flow:</p> <ol style="list-style-type: none"> 1. Dosen melakukan pemasukkan data nilai dalam sebuah arsip berdasarkan aspek penilaian yang telah ditentukan.
--	---

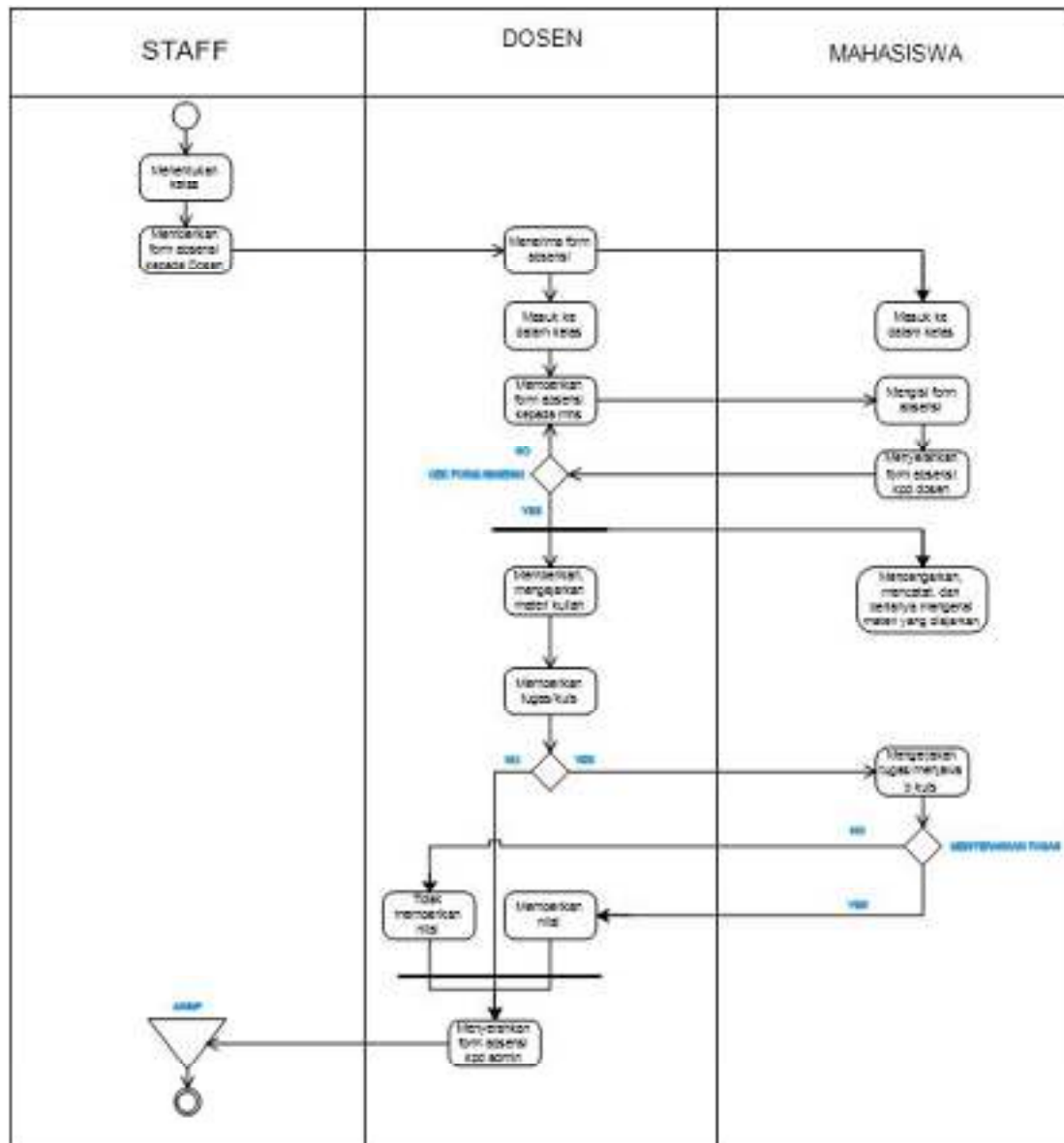
Deskripsi Use Case (Laporan absensi/data nilai)

<p>Aktor Utama: Dosen</p> <p>Aktor Tambahan: Staff</p> <p>Nama Use Case: Laporan Absensi/data nilai</p> <p>Tujuan Use Case: Melakukan proses komputerisasi terhadap data absensi dan nilai yang masih berbentuk <i>hard copy</i>.</p> <p>Kondisi Awal: Dosen memberikan data kepada staff di dalam sebuah form agar dapat didokumentasikan ke dalam komputer.</p> <p>Kondisi Akhir: Staff memasukkan data ke dalam <i>software</i> computer agar dapat diolah kembali.</p>	<p>Optimistic Flow:</p> <ol style="list-style-type: none"> 1. Dosen melakukan pemasukkan data nilai dalam sebuah arsip berdasarkan aspek penilaian yang telah ditentukan.
--	---

Deskripsi Use Case (Laporan penilaian)

<p>Aktor Utama: Staff</p> <p>Aktor Tambahan: Mahasiswa</p> <p>Nama Use Case: Laporan Penilaian</p> <p>Tujuan Use Case: Memberikan informasi penilaian kepada mahasiswa berdasarkan data nilai yang telah diolah sebagai bentuk laporan belajar.</p> <p>Kondisi Awal: Data yang telah diolah melalui proses komputerisasi, di <i>print out</i> dan siap untuk diserahkan kepada mahasiswa.</p> <p>Kondisi Akhir: Hasil <i>print out</i> diserahkan kepada mahasiswa sebagai laporan belajar.</p>	<p>Optimistic Flow:</p> <ol style="list-style-type: none"> 1. Staff melakukan <i>print out</i> data penilaian mahasiswa. 2. Staff menyerahkan hasil <i>print out</i> kepada mahasiswa
---	--

4.1.1.2 Activity Diagram Sistem Perkuliahan



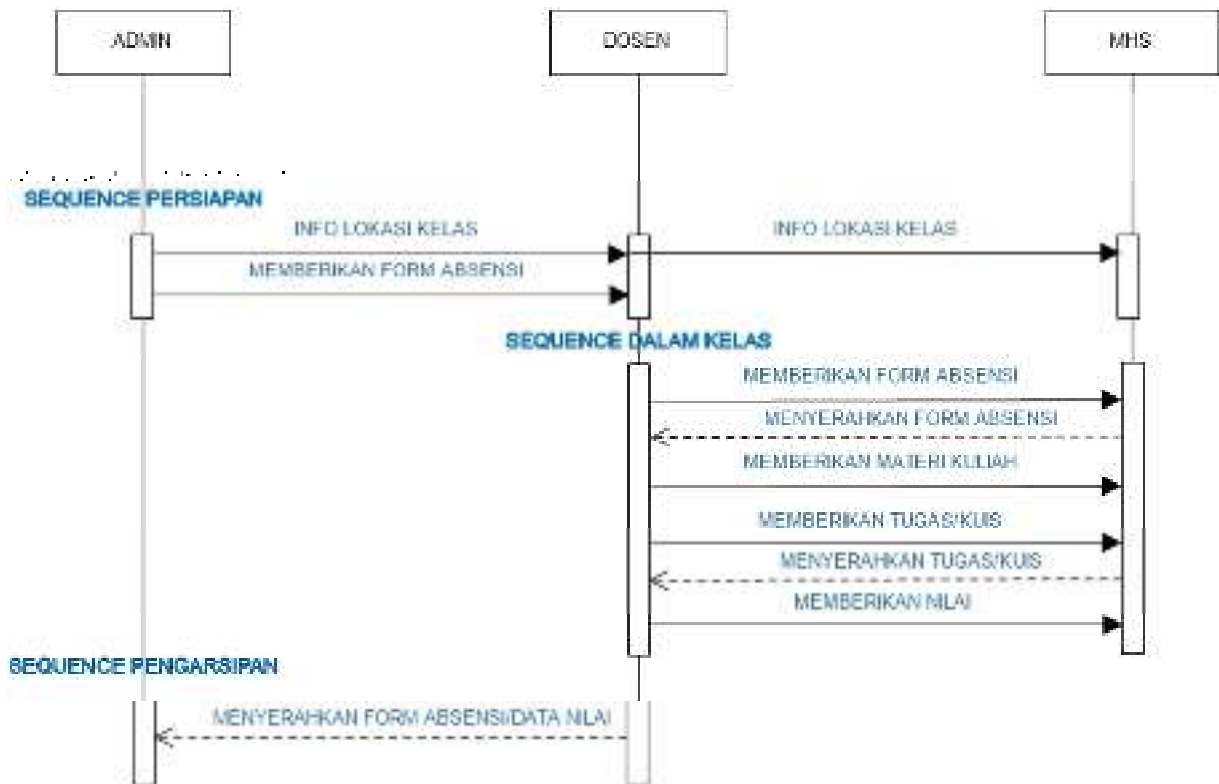
Gambar 4.2 *Activity Diagram Sistem Perkuliahan STMIK Muhammadiyah Jakarta*

Gambar 4.2 menjelaskan tentang aktivitas yang dilakukan antara 3(tiga) aktor, yaitu; Staff, Dosen dan Mahasiswa dalam

sistem perkuliahan STMIK Muhammadiyah Jakarta. Berikut adalah penjelasan bagaimana aktivitas itu dilakukan:

1. Pada awal persiapan sebelum dosen dan mahasiswa memasuki ruang kelas, staff bertugas untuk menentukan ruangan kelas yang akan digunakan dalam proses belajar. Staff juga memberikan form absensi kepada Dosen.
2. Setelah kelas ditentukan, Dosen dan Mahasiswa diwajibkan untuk memasuki kelas yang ditentukan.
3. Saat Dosen dan seluruh Mahasiswa siap untuk belajar, Dosen terlebih dahulu memberikan form absensi untuk ditandatangani oleh Mahasiswa. Kemudian form tersebut dikembalikan lagi kepada Dosen.
4. Kemudian Dosen memberikan materi kuliah, dan Mahasiswa bertugas untuk mendengarkan, mencatat, dan diperbolehkan bertanya tentang materi yang telah diberikan.
5. Pada umumnya Dosen melakukan sesi Tanya jawab atau memberikan tugas setelah materi diberikan, dan seluruh Mahasiswa dianggap tidak ada yang bertanya lagi. Kewajiban bagi Mahasiswa untuk menjawab pertanyaan yang diberikan maupun tugas, karena Dosen memiliki wewenang untuk memberikan nilai lewat media tersebut.
6. Saat jam perkuliahan selesai, Dosen memberikan kembali form absensi, untuk diarsipkan ke dalam computer.

4.1.1.3 Sequence Diagram Sistem Perkuliahan



Gambar 4.3 Sequence Diagram Sistem Perkuliahan
STMIK Muhammadiyah Jakarta

Gambar 4.3 menjelaskan tentang aktivitas yang dikelompokkan ke dalam sebuah *sequence*. Penjelasannya adalah sebagai berikut:

Deskripsi Sequence (Persipan)

Nama Activity.	Aktor	Optimistic Flow
Info Lokasi Kelas	Aktor Utama: Staff Aktor Tambahan: Dosen, Mahasiswa	1. Staff melakukan survey dan menginfokan Lokasi kelas yang dapat digunakan untuk proses perkuliahan kepada dosen. 2. Dosen membagikan info tersebut kepada mahasiswa agar mahasiswa dapat hadir pada kelas yang ditentukan Staff.
Memberikan Form Absensi	Aktor Utama: Staff Aktor Tambahan: Dosen	1. Staff memberikan form absensi kepada dosen agar dapat diisi.

Deskripsi Sequence (Dalam Kelas)

Nama Activity.	Aktor	Optimistic Flow
Memberikan Form Absensi Feedback: Menyerahkan Form Absensi	Aktor Utama: Dosen, Mahasiswa	<ol style="list-style-type: none"> 1. Dosen memberikan form absensi untuk ditandatangani oleh mahasiswa. 2. Mahasiswa menandatangani form absensi yang diberikan. 3. Mahasiswa menyerahkan kembali form absensi kepada dosen.
Memberikan Materi Kuliah	Aktor Utama: Dosen Aktor Tambahan: Mahasiswa	<ol style="list-style-type: none"> 1. Dosen memberikan materi kuliah kepada mahasiswa 2. <i>Feedback</i> dapat berupa pertanyaan dari mahasiswa kepada dosen.
Memberikan Tugas/Kuis Feedback: Menyerahkan Tugas/Kuis	Aktor Utama: Dosen, Mahasiswa	<ol style="list-style-type: none"> 1. Dosen memberikan tugas atau kuis kepada mahasiswa berdasarkan materi kuliah yang telah

		<p>disampaikan.</p> <p>2. Mahasiswa mengerjakan tugas atau menjawab kuis yang diberikan oleh dosen.</p> <p>3. Mahasiswa menyerahkan tugas atau kuis yang telah mereka kerjakan.</p>
Memberikan Nilai	<p>Aktor Utama: Dosen</p> <p>Aktor Tambahan: Mahasiswa</p>	<p>1. Dosen memberikan nilai dan melakukan arsip terhadap nilai tersebut.</p>

Deskripsi Sequence (Pengarsipan)

Nama Activity.	Aktor	Optimistic Flow
Menyerahkan form absensi/data nilai	Aktor Utama: Dosen Aktor Tambahan: Staff	3. Dosen menyerahkan hasil absensi atau data nilai kepada staff, untuk dimasukkan ke dalam computer dan diolah menjadi sebuah laporan.

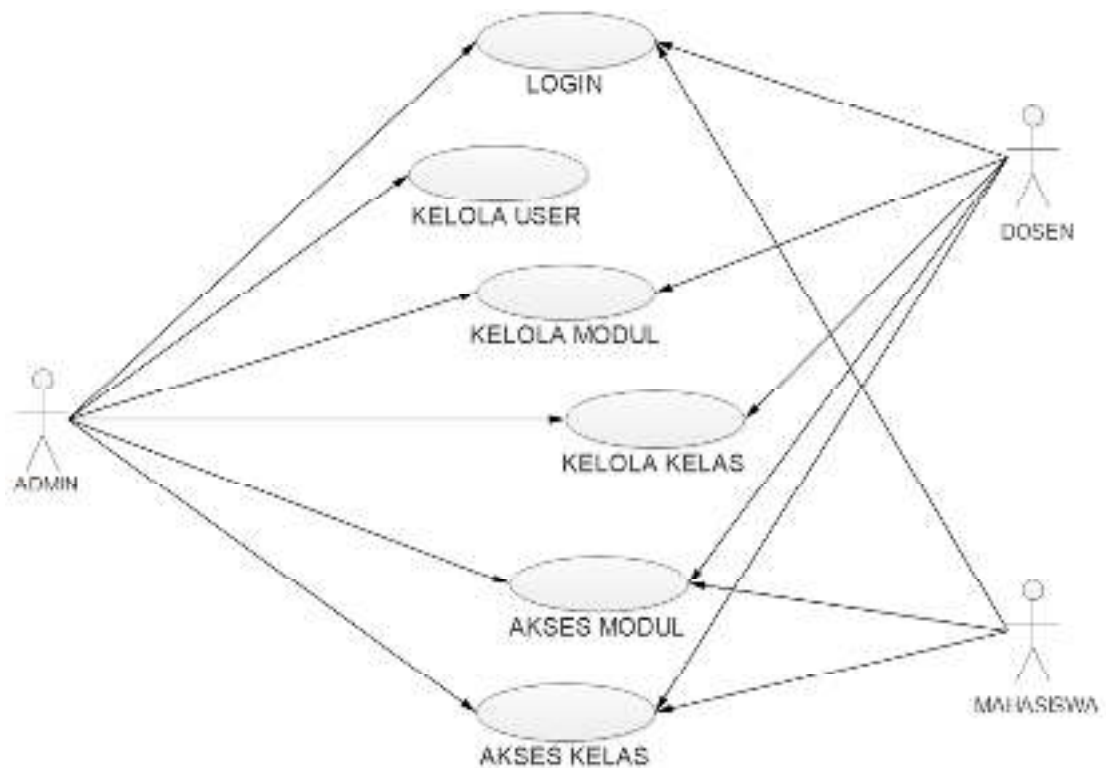
4.1.1.4 Identifikasi Masalah Sistem Yang Berjalan

Berikut adalah identifikasi kekurangan dari sistem yang berjalan pada STMIK Muhammadiyah saat ini:

1. Kegiatan belajar mengajar antara mahasiswa dan dosen masih dilakukan dengan cara konvensional, tanpa memiliki sistem alternative lain. Hal ini akan menyulitkan sebagian mahasiswa yang memiliki kewajiban bekerja.
2. Belum ada tempat penyimpanan yang mengintegrasikan seluruh bahan ajar dosen agar dapat di akses kembali oleh mahasiswa, layaknya repository seluruh modul dan materi.
3. Adanya kemungkinan mahasiswa tidak mendapatkan materi apabila dosen tidak masuk perkuliahan, hal ini sangat merugikan bagi mahasiswa yang mana wajib membayar sks untuk tiap mata kuliah.
4. Kegiatan belajar bagi mahasiswa tidak efektif dan efisien apabila materi yang dibagikan oleh dosen hanya melalui presentasi, catatan, maupun file. Karena ada kemungkinan materi tersebut tersimpan pada masing-masing device mahasiswa dalam kondisi acak.

4.1.2 Analisa Sistem Yang Diusulkan

4.1.2.1 Use Case Diagram Sistem Yang Diusulkan



Gambar 4.4 Use Case Diagram Sistem Yang Diusulkan

Gambar 4.4 menjelaskan tentang peran setiap aktor terhadap sistem yang diusulkan. Use Case di atas hanya menjelaskan tentang garis besar dari sistem yang diusulkan. Berikut penjelasannya:

Deskripsi Use Case (Login)

<p>Aktor Utama: Admin, Dosen, Mahasiswa</p> <p>Aktor Tambahan: -</p> <p>Nama Use Case: Login</p> <p>Tujuan Use Case: Sistem melakukan pengecekan credentials terhadap <i>user</i> yang masuk berdasarkan informasi pada pangkalan data.</p> <p>Kondisi Awal: Menampilkan halaman login pada website dengan 2(dua) input, yaitu email/username dan password.</p> <p>Kondisi Akhir: Success: <i>User</i> akan dialihkan pada halaman beranda user. Failure: Ada notifikasi yang menampilkan pesan error, berdasarkan kesalahan yang dialami.</p>	<p>Optimistic Flow:</p> <ol style="list-style-type: none"> 1. <i>User</i> memasukkan identitas user pada kolom tersedia. 2. <i>User</i> melakukan submit pada website. 3. <i>User</i> akan dialihkan ke halaman beranda. <p>Pesimistic Flow:</p> <ol style="list-style-type: none"> 1. <i>User</i> memasukkan <i>credentials</i> yang tidak dikenali oleh website. 2. Muncul notifikasi error berdasarkan kesalahan yang dialami.
--	--

Deskripsi Use Case (Kelola User)

<p>Aktor Utama: Admin</p> <p>Aktor Tambahan: -</p> <p>Nama Use Case: Kelola User</p> <p>Tujuan Use Case: Admin melakukan pengelolaan berupa tambah, edit, maupun delete terhadap user.</p> <p>Kondisi Awal: Admin mengakses menu <i>activities</i>. Diberikan dua jenis pilihan user yaitu, dosen dan mahasiswa.</p> <p>Kondisi Akhir: Admin akan dialihkan pada halaman user berdasarkan user yang dipilih.</p>	<p>Optimistic Flow:</p> <ol style="list-style-type: none"> 1. Admin melakukan pengelolaan data pada halaman user. 2. Pada halaman tersebut tersedia berbagai fitur seperti tambah, edit dan delete. <p>Pesimistic Flow:</p> <ol style="list-style-type: none"> 1. Terjadi kesalahan pada server sehingga laman tidak dapat diakses.
--	--

Deskripsi Use Case (Kelola Modul)

<p>Aktor Utama: Admin, Dosen</p> <p>Aktor Tambahan: -</p> <p>Nama Use Case: Kelola Modul</p> <p>Tujuan Use Case: Admin atau Dosen melakukan pengelolaan modul, berupa tambah, edit, delete dan publish.</p> <p>Kondisi Awal: Admin atau Dosen menekan menu modul. Kemudian akan dialihkan pada laman administrasi modul.</p> <p>Kondisi Akhir: Admin atau Dosen akan dialihkan pada laman administrasi modul.</p>	<p>Optimistic Flow:</p> <ol style="list-style-type: none"> 1. Website menampilkan laman administrasi modul. 2. Admin atau dosen melakukan pengelolaan modul pada laman modul. 3. Pada halaman tersebut tersedia berbagai fitur seperti tambah, edit, delete dan publish. <p>Pesimistic Flow:</p> <ol style="list-style-type: none"> 1. Terjadi kesalahan pada server sehingga laman tidak dapat diakses.
---	--

Deskripsi Use Case (Kelola Kelas)

<p>Aktor Utama: Admin, Dosen</p> <p>Aktor Tambahan: -</p> <p>Nama Use Case: Kelola User</p> <p>Tujuan Use Case: Admin atau Dosen melakukan pengelolaan kelas, berupa tambah, edit, delete dan publish.</p> <p>Kondisi Awal: Admin atau Dosen menekan menu modul. Kemudian akan dialihkan pada laman administrasi kelas.</p> <p>Kondisi Akhir: Admin atau Dosen akan dialihkan pada laman administrasi kelas.</p>	<p>Optimistic Flow:</p> <ol style="list-style-type: none"> 1. Website menampilkan laman administrasi kelas. 2. Admin atau dosen melakukan pengelolaan kelas pada laman kelas. 3. Pada halaman tersebut tersedia berbagai fitur seperti tambah, edit, delete dan publish. <p>Pesimistic Flow:</p> <ol style="list-style-type: none"> 1. Terjadi kesalahan pada server sehingga laman tidak dapat diakses.
--	--

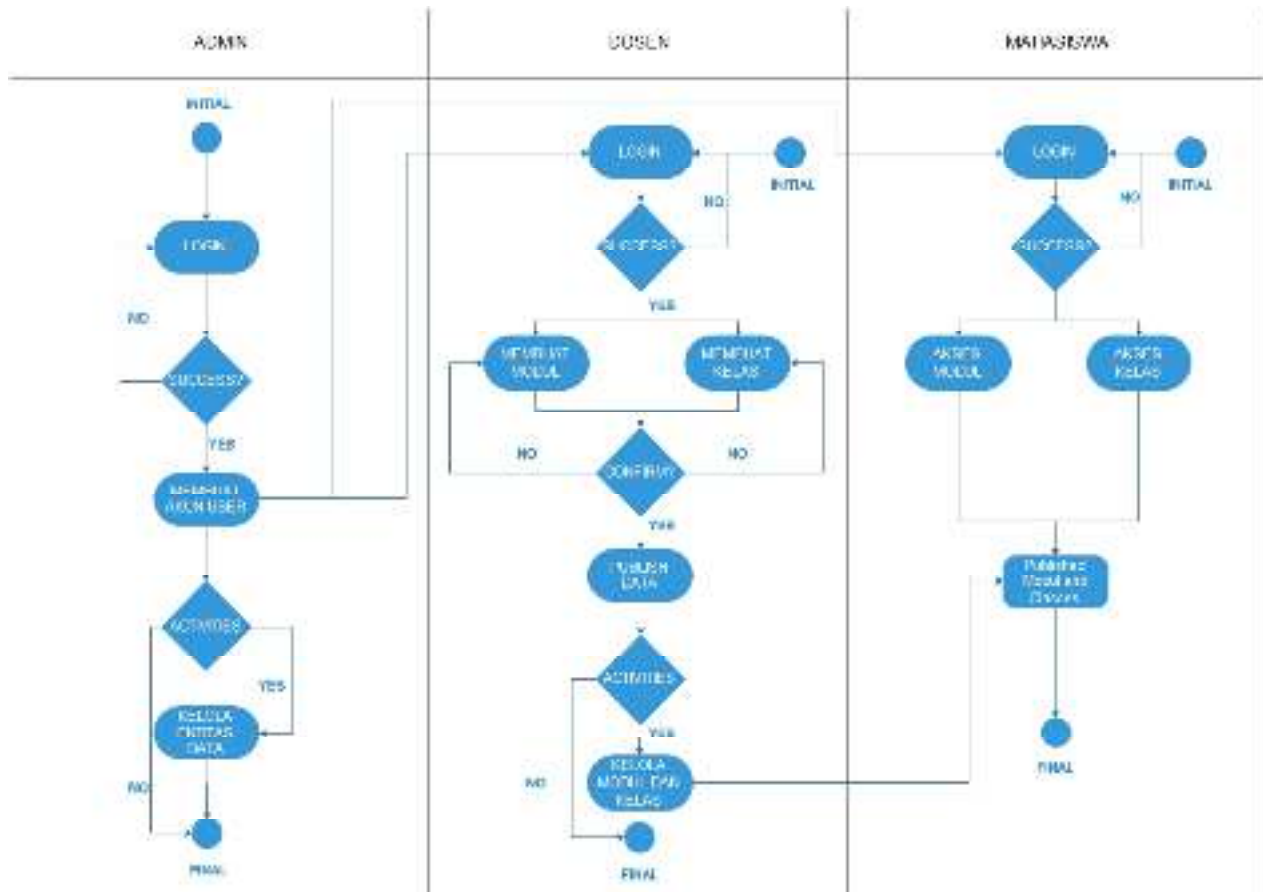
Deskripsi Use Case (Akses Modul)

<p>Aktor Utama: Admin, Dosen, Mahasiswa</p> <p>Aktor Tambahan: -</p> <p>Nama Use Case: Akses Modul</p> <p>Tujuan Use Case: Admin, Dosen, maupun mahasiswa melakukan interaksi terhadap website pada laman modul. Interaksi yang dilakukan adalah membaca modul-modul yang telah dipublish sebelumnya.</p> <p>Kondisi Awal: Pada laman beranda, admin, dosen maupun mahasiswa dapat menekan link yang mengarahkan mereka pada detail lengkap modul yang diinginkan.</p> <p>Kondisi Akhir: Akan muncul laman detail modul yang diklik sebelumnya. <i>User</i> dapat membaca atau <i>me-review</i> modul yang telah dipublish</p>	<p>Optimistic Flow:</p> <ol style="list-style-type: none"> 1. Website menampilkan laman detail modul. 2. Seluruh aktor dapat membaca seluruh isi modul. <p>Pesimistic Flow:</p> <ol style="list-style-type: none"> 1. Terjadi kesalahan pada server sehingga laman tidak dapat diakses.
--	--

Deskripsi Use Case (Akses Kelas)

<p>Aktor Utama: Admin, Dosen, Mahasiswa</p> <p>Aktor Tambahan: -</p> <p>Nama Use Case: Akses Kelas</p> <p>Tujuan Use Case: Admin, Dosen, maupun mahasiswa melakukan interaksi terhadap website pada laman kelas. Interaksi yang dilakukan dapat berupa absen, <i>chatting</i>, <i>posting</i>, dan <i>upload</i> berkas.</p> <p>Kondisi Awal: Aktor dapat menekan link kelas yang akan mengarahkan mereka pada daftar kelas yang mereka ikuti. Dengan menekan link pada daftar kelas, maka aktor/user akan dialihkan pada detail kelas yang telah diklik sebelumnya.</p> <p>Kondisi Akhir: Akan muncul laman detail modul yang diklik sebelumnya. <i>User</i> dapat membaca atau <i>me-review</i> modul yang telah dipublish</p>	<p>Optimistic Flow:</p> <ol style="list-style-type: none"> 1. Website menampilkan laman detail kelas. 2. Seluruh aktor dapat melakukan absen, <i>posting</i>, <i>chatting</i>, dan <i>upload</i> berkas pada laman detail kelas. <p>Pesimistic Flow:</p> <ol style="list-style-type: none"> 1. Terjadi kesalahan pada server sehingga laman tidak dapat diakses.
--	---

4.1.2.2 Activity Diagram Sistem Yang Diusulkan



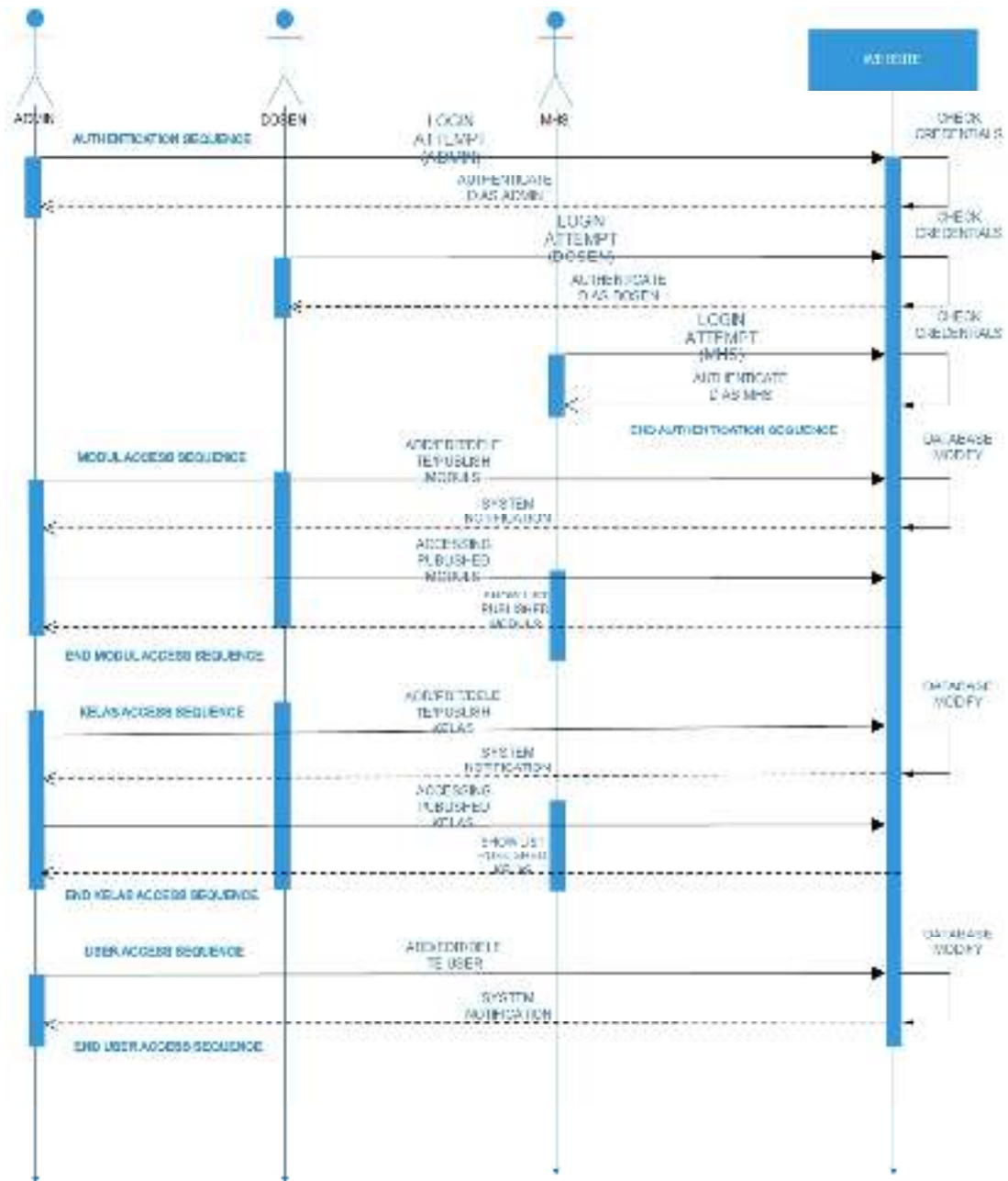
Gambar 4.5 Activity Diagram Sistem Yang Diusulkan

Gambar 4.5 menjelaskan alur sistem yang diusulkan. Alur tersebut dapat dijabarkan sebagai berikut:

1. Admin melakukan autentikasi melalui proses login. Data admin sendiri telah di *seeding* oleh Back-end Developer sebagai entitas awal.
2. Setelah berhasil login, admin dapat membuat user baru dengan 2(dua) *role* pilihan, yaitu Dosen dan Mahasiswa.

3. Admin diberikan pilihan untuk menjalankan activities atau tidak. Arti activities disini adalah menu administrative yang tersedia pada aplikasi, meliputi *CRUD*.
4. Setelah User berhasil dibuat, Dosen maupun Mahasiswadapat menggunakan informasi user yang dibutuhkan untuk login, yaitu email/*username* dan *password*.
5. Ada perbedaan *user interface* pada website tergantung *role* apa yang terautentikasi. Dosen memiliki *user interface* yang memungkinkan untuk membuat modul dan kelas, sedangkan mahasiswa tidak bisa membuat keduanya.
6. Dosen juga diberikan pilihan untuk mengelola modul dan kelas yang telah mereka buat. Admin juga diberikan otoritas ini.
7. Pada sistem ini, Admin memiliki wewenang tertinggi diikuti Dosen dan Mahasiswa.

4.1.2.3 Sequence Diagram Sistem Yang Diusulkan



Gambar 4.6 Sequence Diagram Sistem Yang Diusulkan

Gambar 4.6 menjelaskan pokok-pokok alur sistem yang dikelompokkan ke dalam beberapa *sequence*. Alur tersebut dapat dijabarkan sebagai berikut:

Deskripsi Sequence (Authentication)

Nama Activity.	Aktor	Optimistic Flow
Login Attempt (ADMIN)	<p>Aktor Utama: Admin</p> <p>Aktor Tambahan: -</p>	<ol style="list-style-type: none"> Admin memasukkan username/email dan password untuk melakukan autentikasi. Setelah melakukan submit, <i>back-end website</i> akan melakukan pengecekan <i>credentials</i> terhadap input user. Apabila sukses, website akan mengautentikasi user dengan role admin ke dalam website.
Login Attempt	Aktor Utama:	1. Dosen

(DOSEN)	<p>Dosen</p> <p>Aktor Tambahan:</p> <p>-</p>	<p>memasukkan username/email dan password untuk melakukan autentikasi.</p> <p>2. Setelah melakukan submit, <i>back-end website</i> akan melakukan pengecekan <i>credentials</i> terhadap input user.</p> <p>3. Apabila sukses, website akan mengautentikasi user dengan role dosen ke dalam website.</p>
Login Attempt (MHS)	<p>Aktor Utama:</p> <p>Mahasiswa</p> <p>Aktor Tambahan:</p> <p>-</p>	<p>1. Mahasiswa memasukkan username/email dan password untuk melakukan autentikasi.</p> <p>2. Setelah melakukan submit, <i>back-end website</i> akan</p>

		<p>melakukan pengecekan <i>credentials</i> terhadap input user.</p> <p>3. Apabila sukses, website akan mengautentikasi user dengan role mahasiswa ke dalam website.</p>
--	--	---

Deskripsi Sequence (Modul Access)

Nama Activity.	Aktor	Optimistic Flow
Add/ Edit/ Delete/ Publish Modul	Aktor Utama: Admin, Dosen Aktor Tambahan: -	1. Admin dan Dosen mengakses menu modul. 2. Pada menu tersebut diberikan beberapa fitur yang meliputi <i>CRUD</i> dan publish untuk modul. 3. Saat admin maupun dosen hendak menggunakan fitur tersebut, sistem secara otomatis memberikan notifikasi terhadap status proses pada aplikasi. 4. Apabila sukses, user akan diberikan pesan yang menunjukkan berhasilnya perubahan data. Data pada database

		juga telah dimanipulasi pada proses ini.
Accessing Published Modul	Aktor Utama: Admin, Dosen, Mahasiswa Aktor Tambahan: -	<ol style="list-style-type: none"> 1. Seluruh role user dapat mengakses modul pada beranda. 2. Apabila link modul pada beranda diklik, user akan <i>redirect</i> pada laman detail modul. 3. Modul-modul yang ada pada beranda, adalah modul-modul yang telah <i>publish</i> oleh admin maupun dosen.

Deskripsi Sequence (Kelas Access)

Nama Activity.	Aktor	Optimistic Flow
Add/ Edit/ Delete/ Publish Kelas	Aktor Utama: Admin, Dosen Aktor Tambahan: -	<ol style="list-style-type: none"> 1. Admin dan Dosen mengakses menu kelas. 2. Pada menu tersebut diberikan beberapa fitur yang meliputi <i>CRUD</i> dan publish untuk kelas. 3. Saat admin maupun dosen hendak menggunakan fitur tersebut, sistem secara otomatis memberikan notifikasi terhadap status proses pada aplikasi. 4. Apabila sukses, user akan diberikan pesan yang menunjukkan berhasilnya perubahan data. Data pada database

		juga telah dimanipulasi pada proses ini.
Accessing Published Kelas	Aktor Utama: Admin, Dosen, Mahasiswa Aktor Tambahan: -	<ol style="list-style-type: none"> 1. Seluruh role user dapat mengakses kelas yang telah mereka ikuti. 2. Pendaftaran kelas dapat dilakukan dengan mengakses menu gabung kelas. Menu ini tidak ada pada role admin, karena admin secara sistem dapat langsung mengikuti seluruh kelas yang telah dibuat oleh user. 3. Pada menu kelas, user dapat mengakses “detail kelas” dengan mengklik link pada kelas yang diinginkan. 4. Pada detail kelas

		terdapat berbagai fitur seperti absen, <i>comment</i> , <i>upload</i> , dan <i>posting</i> .
--	--	--

Deskripsi Sequence (User Access)

Nama Activity.	Aktor	Optimistic Flow
Add/ Edit/ Delete User	<p>Aktor Utama: Admin</p> <p>Aktor Tambahan: -</p>	<ol style="list-style-type: none"> 1. Admin mengakses menu dosen/mahasiswa pada <i>dropdown</i> “activities”. 2. Setelah memilih salah satu dari 2(dua) link tersebut, Admin akan dialihkan pada laman user tersebut. 3. Pada laman ini admin dapat melihat list data seluruh user yang terdaftar. 4. Pada halaman ini juga, admin dapat mengakses fitur CRUD yang tersedia. 5. Setelah melakukan submit pada salah satu fitur, sistem akan memberikan notifikasi terhadap

		<p>proses aplikasi.</p> <p>6. Apabila sukses, sistem akan menampilkan pesan “berhasil”.</p> <p>7. Apabila gagal, sistem akan menampilkan pesan <i>error</i> berdasarkan kesalahan yang dialami.</p>
--	--	---

4.2 Analisa Kebutuhan

4.2.1 Kebutuhan Hardware

Dalam penerapan rancangan sistem yang telah dijelaskan sebelumnya, dibutuhkan beberapa perangkat keras untuk melakukan *testing*. Adapun perangkat keras yang digunakan adalah:

- Perangkat keras yang akan digunakan dalam pembuatan sistem:

Pada proses perancangan sistem, dalam mayoritas waktunya, penulis menggunakan Laptop Acer e15-975g dengan spesifikasi sebagai berikut :

Tabel 4.1 Kebutuhan Hardware Dalam Perancangan

No.	Nama Hardware	Spesifikasi
1	Processor	Intel @i5-7200
2	RAM	4GB
3	HDD	1TB
4	VGA	NVIDIA GeForce 940MX

- Minimum Spesifikasi Perangkat keras yang digunakan dalam implementasi sistem:

Tabel 4.2 Minimum Spesifikasi Kebutuhan Hardware

No.	Nama Hardware	Min. Spesifikasi
1	Processor	Intel @Pentium 4
2	RAM	2GB
3	HDD	500GB
4	Monitor	-

4	Perangkat keras tambahan	-
---	--------------------------	---

4.2.2 Kebutuhan Software

- Perangkat lunak yang digunakan untuk pembuatan sistem:

Tabel 4.3 Perangkat Lunak Pembuatan Sistem

No.	Nama Software	Keterangan
1	Windows 10 64 bit	Sebagai Sistem Operasi
2	Visual Code Studio	Sebagai Text Editor
3	XAMPP	Sebagai Local Server
4	Google Chrome, Mozilla Firefox, Opera	Sebagai Web Browser
5	GitBash	Sebagai Terminal
6	GitLab	Sebagai Repository

- Perangkat lunak yang digunakan untuk pembuatan *user interface*:

Tabel 4.4 Perangkat Lunak Pembuatan Sistem

No.	Nama Software	Keterangan
1	Adobe Photoshop CS6	Sebagai Aplikasi Desain grafis dengan format pixel
2	Undraw.co	Website yang menyediakan <i>free open-source</i> desain dengan format .svg
3	Adobe Illustrator CS6	Sebagai aplikasi desain grafis dengan format vector

- Perangkat lunak yang digunakan untuk implementasi:

Tabel 4.5 Perangkat Lunak Implementasi Sistem

No.	Nama Software	Keterangan
1	Windows 10 64 bit	Sebagai Sistem Operasi
2	Google Chrome	Sebagai Web Browser
3	XAMPP	Sebagai Local Server

4.2.3 Kebutuhan Brainware

Kebutuhan Brainware adalah kebutuhan dimana aplikasi memerlukan operator untuk dapat berjalan, dan melakukan seluruh proses yang diinginkan. Kebutuhan Brainware memegang peranan penting dan dibutuhkan sumber daya manusia yang memenuhi kriteria karena sebuah kesalahan fatal dapat mengakibatkan seluruh sistem *down*. Sumber daya manusia yang dibutuhkan dalam pengoperasian *web* ini antara lain:

1. Sistem Analis

Analisis Sistem adalah penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya. Maka Sistem analis berarti orang yang bertugas merancang, *men-design*, dan manajemen seluruh kebutuhan perangkat lunak, untuk dibuat sebagai sistem informasi secara utuh.

Kegagalan dalam memahami kebutuhan sistem adalah kesalahan fatal dalam *project* pengembangan *software*. Maka dari itu, sistem analis haruslah merupakan orang yang paham betul mengenai seluruh bagian vital dalam perancangan sistem. Dan pada umumnya, seorang sistem analis harus mengetahui tentang *data flow*, *schema*, *structure database*, bahkan sedikit tentang *coding*. Untuk aplikasi ini, sistem analisis diperlukan saat adanya perubahan fitur, relasi data dan lain sebagainya.

2. Web Developer

Bertugas melakukan penulisan kode-kode pemrograman dan melakukan perancangan program sesuai dengan arahan sistem analis. Karena aplikasi ini merupakan *web-based* maka diperlukan 2(dua) jenis kemampuan yang harus dikuasai, antara lain:

a. Front-end Developer

Tugas pokoknya melakukan perancangan desain *user-interface*. Tidak hanya itu, Front-end Developer juga harus memiliki koordinasi yang baik dengan Back-end Developer, karena akan bertanggung jawab menangani data binding dan request form.

b. Back-end Developer

Bertugas untuk menerima request dari front-end, melakukan strukturisasi database sesuai arahan sistem analis, membuat *controller* ataupun *endpoint* sesuai dengan kebutuhan sistem dan melakukan *seeding* atau bahasa mudahnya, melakukan input data awal ke database untuk user admin.

3. Admin

Pada sistem *E-Learning* STMIK Muhammadiyah Jakarta ini, tugas admin adalah sebagai berikut:

a. Membuat User

Tugas dari seorang admin adalah membuatkan user bagi pengguna aplikasi agar dapat digunakan untuk autentikasi ke

dalam aplikasi. Ada 2(dua) jenis User yang harus dibuatkan oleh Admin, yaitu:

1. Dosen

Dosen adalah user yang memiliki kewenangan untuk membuat, mengedit, dan menghapus modul dan kelas.

2. Mahasiswa

Mahasiswa adalah user yang hanya memiliki kewenangan untuk mengikuti modul dan kelas.

- b. Mengelola Data

Tugas lainnya dari admin adalah mengelola entitas data. Admin diberikan wewenang untuk melakukan *CRUD*, yaitu *Create, Read, Update* dan *Delete*.

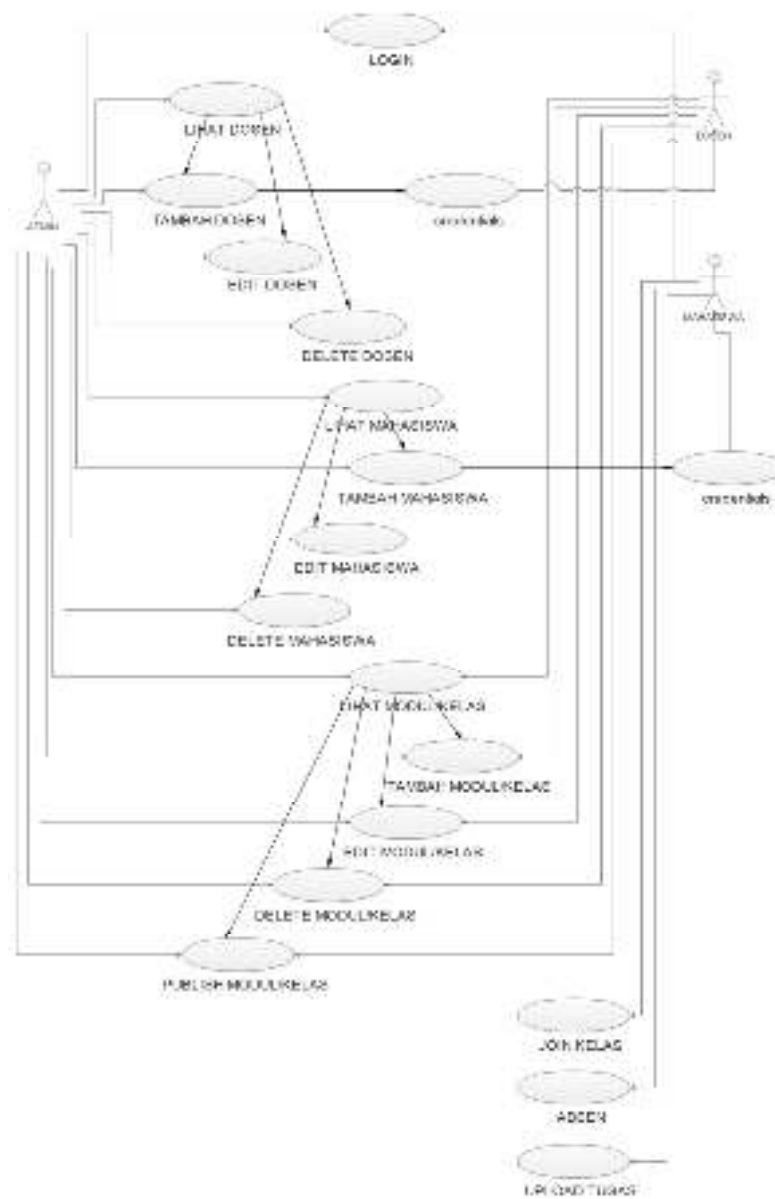
4. User

User atau pengguna, adalah orang yang melakukan interaksi di dalam aplikasi. User dibagi menjadi 3(tiga) *role*, yaitu; Admin, Dosen dan Mahasiswa.

4.3 Desain Sistem (*System Design*)

Untuk mendesain sistem disini digunakan UML diagram yang meliputi *use case diagram*, *class diagram*, *sequence diagram* dan *activity diagram*.

4.3.1 Use Case Diagram Sistem



4.3.2 Diagram Class Sistem

