

**Laporan Tugas Kelompok
Rekayasa Perangkat Lunak - Ganjil 2023/2024**

Software Requirements Specification & Design Document

OpeNotes

Aplikasi Berbagi Catatan Pacil

Nama Kelompok

2106752180 - Alvaro Austin

2106629963 - Muflih Naufal Maxi

2106629995 - Fernando Nathaniel Sutanto

2106630044 - Christopher Nathanael Wijaya

2106706691 - Febrian Dwi Kimhan

Asisten Dosen: Achmad Fikri Adidharma

**Fakultas Ilmu Komputer
Universitas Indonesia
2023**

Daftar Isi

Riwayat Revisi	3
I. Latar Belakang	4
II. System Requirements	6
III. Requirements Modeling	15
A. Use Case Diagram	15
B. Use Case Specifications (UCS)	16
1. UCS 1: Manage Mata Kuliah (CRUD)	16
2. UCS 2: View Notes (Read)	19
3. UCS 3: View Detail Notes	20
4. UCS 4: Post Notes	22
5. UCS 5: Manage Favorite	23
6. UCS 6: Save to Favorite	25
7. UCS 7: Report Notes	26
8. UCS 8: Read User Reports	28
9. UCS 9: Remove Notes	29
10. UCS 10: Vote Notes	30
11. UCS 11: Reply Notes	32
C. Use Case Activity Diagrams	34
1. Activity Diagram for UCS 1: Manage Mata Kuliah (Alvaro)	34
2. Activity Diagram for UCS 2: View Notes (Alvaro)	36
3. Activity Diagram for UCS 3: View Detail Notes (Christo)	37
4. Activity Diagram for UCS 4: Post Notes (Christo)	38
5. Activity Diagram for UCS 5: Manage Favorite (Febrian)	38
6. Activity Diagram for UCS 6: Save to Favorite (Febrian)	41
7. Activity Diagram for UCS 7: Report Notes (Muflih)	41
8. Activity Diagram for UCS 8: Read User Reports (Muflih)	42
9. Activity Diagram for UCS 9: Remove Notes (Muflih)	42
10. Activity Diagram for UCS 10: Vote Notes (Nando)	43
11. Activity Diagram for UCS 11: Reply Notes (Nando)	44
D. [TK 3: 25 /100] Analysis Class Diagram	44
IV. System Design	45
A. [TK 4: 10/100] Design Class Diagram	45
B. [TK 4: 15/100] Tabel Class Diagram	45
C. [TK 4: 75/100] Sequence Diagrams	45
1. Sequence Diagram [Nama Use Case 1]	45
2. Sequence Diagram [Nama Use Case 2]	45

Riwayat Revisi

Tanggal	Ringkasan Revisi	Penulis Revisi
Feb 15 th , 2022	Template Preparation	lis Solichah
Sep 7 th 2023	Mengerjakan Draft TK1 (Latar Belakang)	Semua Anggota OpeNotes
Sep 8 th 2023	Menyelesaikan Draft TK1 yang direncanakan akan dikumpulkan tanggal 8 September 2023.	Semua Anggota OpeNotes
Sep 15 th 2023	Melakukan revisi TK1 yang telah disediakan asisten dosen pada tanggal 15 September	Semua Anggota OpeNotes
Sep 21 st 2023	Membuat UCD untuk TK2	Semua Anggota OpeNotes
Sep 29 th 2023	Membuat UCS untuk TK2	Semua Anggota OpeNotes
Sep 6 th 2023	Membuat AD untuk TK3	Semua Anggota OpeNotes

I. Latar Belakang

Perguruan tinggi merupakan lingkungan pendidikan serta sarana di mana mahasiswa dapat mengakses serta berbagi pengetahuan dalam rangka mencapai tujuan akademik mereka. Universitas Indonesia (UI), merupakan salah satu perguruan tinggi di Indonesia, masih terdapat permasalahan dalam akses fasilitas pertukaran informasi yang efektif dan efisien di antara mahasiswanya. Fakultas Ilmu Komputer (Fasilkom), sebagai salah satu fakultas di Universitas Indonesia yang memiliki banyak mahasiswa yang berdedikasi, masih mengalami permasalahan serupa dalam konteks pengorganisasian dan pengelolaan informasi yang berkaitan dengan materi perkuliahan dan pengalaman akademik.

Pada saat ini, masih terdapat beberapa kendala yang perlu diatasi untuk memenuhi kebutuhan informasi dan komunikasi di Fasilkom UI. Meskipun terdapat berbagai *platform* komunikasi seperti WhatsApp, Facebook, Twitter, keterbatasan *platform* tersebut dalam menyajikan, mengorganisasi, dan menyimpan informasi secara sistematis telah menjadi kendala yang signifikan. Informasi perkuliahan, seperti catatan, sumber daya, atau pengalaman akademik, seringkali hanya tercecer di berbagai grup dan forum, menyulitkan akses dan manajemen informasi yang efisien.

Sampai saat ini, belum ada *platform* yang benar-benar menyatukan seluruh mahasiswa Fasilkom UI. Keterbatasan ini menghambat pertukaran pengetahuan, pengalaman, dan sumber daya yang berpotensi memberikan manfaat yang signifikan dalam pengembangan akademik tiap mahasiswa. Sejumlah mahasiswa mungkin menghadapi kendala aksesibilitas terhadap sumber daya digital seperti Google Drive atau *platform* serupa lainnya yang digunakan untuk berbagai catatan dan materi perkuliahan. Hal ini mengakibatkan adanya kesenjangan dalam kemampuan mereka untuk mengikuti perkembangan dalam kurikulum perkuliahan, mengakses materi yang relevan, dan mengambil langkah-langkah proaktif dalam perkuliahan mereka.

OpeNotes dirancang untuk menjadi *platform* situs jejaring sosial berbasis edukasi dan diharapkan dapat menjadi penunjang sumber daya perkuliahan. OpeNotes menghadirkan fitur yang mendukung mahasiswa untuk saling berbagi catatan perkuliahan, pengalaman, memberikan tips, memberikan testimonial seputar mata kuliah tertentu, dan hal seputar perkuliahan Fasilkom lainnya. OpeNotes memungkinkan adanya interaksi sesama Mahasiswa Fasilkom untuk saling berbagi serta menanggapi pendapat satu sama lain yang dikemas dalam bentuk "Notes". OpeNotes berfokus pada kemudahan akses bagi seluruh mahasiswa Fasilkom UI serta memungkinkan terjadinya perolehan akses informasi secara luas bagi Mahasiswa Fasilkom UI

yang terhubung satu sama lain dalam suatu hubungan jejaring sosial edukasi serta . Mahasiswa Fasilkom diharapkan dapat memanfaatkan OpeNotes untuk berbagi pengalaman, memberikan tips, dan memberikan testimonial seputar mata kuliah tertentu. Hal ini memberikan peluang untuk memperluas cakupan *platform* dan menjadikannya lebih relevan dengan kebutuhan dan ekspektasi mahasiswa.

Dengan memahami kendala dan pertimbangan perluasan konsep OpeNotes, kami bermaksud untuk menciptakan *platform* yang dapat memberikan solusi yang efektif. OpeNotes akan menjadi wadah yang mampu memfasilitasi komunikasi serta kolaborasi antara mahasiswa Fasilkom UI, sehingga diharapkan dapat membantu mengatasi kendala-kendala yang dihadapi dan memperkuat komunitas akademik mahasiswa.

II. System Requirements

Penjelasan Umum:

OpeNotes merupakan situs jejaring sosial berbasis edukasi yang ditujukan bagi mahasiswa Fasilkom UI di mana pengguna dapat membagikan posting singkat yang pada aplikasi ini dikenal sebagai "**Notes**". Notes dapat digunakan oleh mahasiswa untuk membagikan catatan/sumber pembelajaran, pengalaman belajar, *tips & tricks*, serta hal-hal lainnya seputar perkuliahan. Notes dapat mengandung gabungan dari teks, *file*, foto, ataupun *link*.

Notes yang akan dibagikan oleh pengguna memiliki beberapa komponen dan hal penting lain yang melingkupi:

- Pembuat Notes dapat membuat *tag* baru atau memilih dari *tag* yang sudah ada untuk mendeskripsikan secara singkat kategori umum dari notes tersebut, contoh *tag* suatu Notes: Materi, UAS, UTS, dll.
- Pengguna dapat me-*reply*, *report*, serta memberikan *upvote* atau *downvote* pada suatu Notes.
- Notes akan menampilkan profil singkat pembuatnya, tombol favorit, banyaknya *reply*, serta banyak nya *upvote* & *downvote* pada Notes tersebut.

Role:

1. **Mahasiswa:** sebuah *role* dari aplikasi OpeNotes yang didapatkan setelah diverifikasi identitas mahasiswanya (melalui SSO UI *Auth Service*). Hal yang menjadi garis besar dari adanya *role* mahasiswa adalah sebagai *role* utama pada website ini dengan aksesibilitas pada fitur membuat Notes, *reply* Notes, dll. Mahasiswa juga dapat membantu tugas admin untuk menjaga keamanan dan ketertiban website dengan bisa me-*report* Notes jika dirasa melanggar peraturan.

Beberapa fitur yang dapat dimiliki oleh Mahasiswa: *Logout*, *Read Notes*, *Post Notes*, *Bookmark Mata Kuliah*, *Save to Favorites*, *Reply Notes*, *Vote Notes*, *Search Notes By Tag*, *Manage Favorite*, *Report Notes*.

2. **Admin:** sebuah *role user* dari aplikasi OpeNotes yang bertanggung jawab untuk mengelola dan mengawasi aplikasi OpeNotes. Hal yang menjadi garis besar dari adanya admin adalah admin bertugas untuk menjaga ketertiban dan keteraturan hal-hal yang dibagikan dalam aplikasi OpeNotes. Apabila terdapat konten yang kurang berkenan dan relevan maka admin memiliki tanggung jawab untuk menghapus hal tersebut pada aplikasi OpeNotes ini. Selain itu Admin juga dapat mengelola daftar mata kuliah yang ada pada aplikasi OpeNotes. Admin juga tidak menutup kemungkinan dapat melakukan fitur yang role mahasiswa dapat lakukan.
3. **Guest:** sebuah *role* dari aplikasi OpeNotes yang belum melakukan autentikasi maupun otorisasi, tetapi ingin melihat atau menggunakan catatan dan latihan yang ada di aplikasi OpeNotes. *Guest* dapat melihat Notes yang bersifat publik pada aplikasi ini. *Guest* juga tidak dapat membuat atau mengedit Notes. *Guest* tidak dapat memberi *reply* ataupun memberi *upvote* atau *downvote* pada suatu Notes. *Guest* harus melakukan verifikasi (melalui autentikasi dan otorisasi) untuk mendapatkan akses lebih banyak ke aplikasi OpeNotes sebagai mahasiswa.

Beberapa fitur yang dapat dimiliki oleh *Guest*: *Login* (Autentikasi), *Find Mata Kuliah*, *Read Notes*, *Search Notes By Tag*.

Fitur:

1. Autentikasi (*Login & Logout*)

Pengguna akan diminta untuk login terlebih dahulu untuk melakukan validasi sebagai mahasiswa Fasilkom UI. Pengguna dapat melewati langkah ini dan akan diidentifikasi sebagai *guest*. Namun, *guest* tidak dapat menggunakan beberapa fitur yang dapat digunakan mahasiswa.

Alur:

1. Pengguna membuka aplikasi OpeNotes. Apabila pengguna belum terautentikasi, pengguna dapat menekan tombol *login*.
2. Setelah menekan tombol *login*, pengguna akan diarahkan pada laman SSO UI untuk mengisi kredensial.

3. Sebelum terautentikasi, akan dilakukan pengecekan terlebih dahulu terhadap kredensial yang telah dimasukan. Apabila kredensial yang dimasukan benar akan dilakukan pengecekan kembali apakah email yang telah dimasukan teridentifikasi sebagai Admin.
4. Pengguna yang telah berhasil terautentikasi akan diarahkan menuju *homepage* sebagai mahasiswa.
5. Pengguna dapat menekan tombol *logout* untuk mengakhiri *session*.
6. Sistem akan melakukan penghapusan terhadap *cookie* yang disimpan server sebagai indikasi autentikasi.
7. Pengguna tetap dapat menggunakan aplikasi, namun sebagai *guest*. Jika ingin menggunakan aplikasi sebagai mahasiswa, pengguna harus *login* kembali.

2. View Notes

Setelah pengguna memilih suatu mata kuliah, pengguna dapat melihat daftar Notes awal yang ada dalam mata kuliah tersebut.

Alur:

1. Pengguna memilih mata kuliah yang diinginkan terlebih dahulu.
2. Setelah pengguna memilih mata kuliah, pengguna dapat melihat daftar Notes yang telah di-*post* oleh pengguna lain terkait mata kuliah yang telah dipilih sebelumnya.

3. View Detail Notes

Pengguna dapat memilih suatu Notes dari kumpulan Notes awal yang ditampilkan setelah tahap *View Notes* untuk dilihat secara lebih detail. Notes yang telah dipilih akan menampilkan kumpulan *reply* Notes yang telah dipilih sebelumnya.

Alur:

1. Pengguna memilih suatu Notes tertentu dari kumpulan Notes awal yang ditampilkan sistem setelah memilih mata kuliah tertentu
2. Pengguna dapat melihat Notes yang telah dipilih pada tahap ke-1 secara lebih detail. Pengguna dapat melihat kumpulan *reply* pada Notes yang telah dipilih tersebut.

4. Post Notes

Pengguna dapat mem-*post* Notes yang berisikan teks/gambar/*file*. Selain itu, pengguna dapat menambahkan *tag* yang relevan dengan Notes tsb.

Alur:

1. Setelah memilih mata kuliah, pengguna menekan tombol untuk mem-posting notes baru
2. Pengguna menentukan judul dan isi dari Notes
3. Pengguna dapat menambahkan gambar atau *file* pada Notes tersebut
4. Pengguna dapat membuat atau menambahkan beberapa *tag* pada Notes
5. Setelah melakukan konfirmasi, notes baru buatan pengguna dapat dilihat pengguna-pengguna lain dalam mata kuliah yang telah dipilih sebelumnya.
6. Suatu Notes yang telah dibuat dapat dihapus dengan menekan tombol *delete* yang ada pada Notes tersebut.

5. **Bookmark Mata Kuliah**

Untuk memudahkan mahasiswa dalam mencari Notes suatu matkul, mahasiswa dapat meng-*bookmark* mata kuliah (matkul) yang ingin diikutinya. Tidak hanya itu, mahasiswa dapat melihat *bookmark* yang dibuatnya melalui menu “Lihat Bookmark Saya”, dimana saat kita menekan menu tersebut, kita akan dinavigasikan menuju *bookmark* .

Alur:

1. Pada laman cari mata kuliah, mahasiswa dapat memberikan bintang yang menandakan *bookmark* suatu matkul. Setelah itu, seluruh mata kuliah yang diberi bintang akan muncul di baris paling atas.
2. Jika ingin menghapus *bookmark* dari suatu matkul, mahasiswa dapat menekan kembali tombol bintang tersebut.

6. **Save to Favorites**

Mahasiswa dapat menyimpan beberapa Notes favoritnya ke dalam suatu folder.

Alur:

1. Pada suatu Notes, mahasiswa dapat mengklik tombol *kebab* dan mengklik tombol *Save*.
2. Pengguna dapat memilih satu folder untuk menyimpan Notes tersebut. Pengguna juga bisa membuat folder baru untuk menyimpan Notes tersebut.

3. Notes yang disimpan tersebut dapat dilihat mahasiswa di menu favorit.
4. Jika ingin membatalkan penyimpanan Notes dari suatu folder, mahasiswa bisa menekan kembali tombol *unsave* di Notes tersebut.

7. Reply Notes

Suatu Notes dapat di-*reply* dengan suatu Notes lain. Admin dan Mahasiswa dapat melakukan *reply* terhadap suatu Notes yang ada, sedangkan *guest* **tidak** dapat melakukan *reply* pada suatu Notes.

Alur:

1. Pengguna menekan Notes yang ingin di-*reply*.
2. Pengguna menekan tombol *reply* yang terdapat pada Notes.
3. Pengguna kemudian membuat Notes yang akan menanggapi Notes terkait yang telah dipilih sebelumnya.
4. Notes tanggapan yang telah berhasil dibuat akan ditampilkan di bawah Notes yang telah dipilih pada langkah pertama.

8. Vote Notes

Suatu Notes dapat di-*upvote* atau di-*downvote* oleh pengguna lain. Admin dan mahasiswa dapat melakukan *vote* terhadap suatu Notes yang ada, sedangkan *guest* **tidak** dapat melakukan *vote* pada suatu Notes.

Alur:

1. Pengguna dapat menekan tombol *upvote* atau *downvote* pada suatu Notes. Pengguna tidak dapat menekan keduanya sekaligus.
2. Pengguna hanya dapat menekan *thumbs up* atau *thumbs down* (sekali saja), sehingga fungsi ini opsional bagi pengguna yang terautentikasi (mahasiswa/admin).
3. Jika ingin membatalkan *vote* tersebut, pengguna dapat mengklik sekali lagi tombol *vote* tersebut.

9. Search Notes by Tag

Pengguna dapat mencari suatu Notes dengan menggunakan *tag* yang telah dicantumkan pada Notes yang dibuat. Tentu saja suatu Notes bisa saja tidak memiliki *tag*, namun fitur

pencarian ini dapat membantu kita dalam mencari Notes yang sesuai berdasarkan *tag* pada Notes yang ada.

Alur:

1. Pengguna menekan *toolbar search* untuk melakukan *search*.
2. Pengguna menuliskan *tag* yang ingin dicari. Sistem menyediakan fitur *autocomplete* agar pengguna dapat mencari *tag* dengan lebih cepat.
3. Proses ini dapat diulangi secara terus-menerus agar pengguna dapat melakukan *search* untuk Notes yang memiliki *tag-tag* yang dipilih pengguna.
4. Pengguna menekan tombol *search*
5. Hasil *query* ditampilkan ke pengguna

10. **Manage Favorite**

Mahasiswa dapat menambahkan, mengganti nama, atau menghapus folder di menu *Favorite*.

Alur:

1. Mahasiswa mengklik menu *Favorite*.
2. Pada menu *Favorite*, mahasiswa dapat mengklik tombol untuk menambahkan folder.
3. Jika ingin mengganti nama suatu folder, mahasiswa bisa mengklik tombol *kebab* dan mengklik tombol *ganti nama*.
4. Jika ingin menghapus suatu folder, mahasiswa bisa mengklik tombol *kebab* dan mengklik tombol *hapus*.

11. **Report Notes**

Mahasiswa dapat melakukan *report* pada suatu Notes.

Alur:

1. Mahasiswa dapat menekan tombol *report* pada Notes.
2. Mahasiswa dapat memilih jenis *report* yang akan diberikan, contoh: Mengandung kata-kata kasar, Mengandung unsur sara, dll.

12. **Remove Notes**

Suatu Notes dapat dihapus oleh admin. Notes yang bisa dihapus oleh admin adalah seluruh Notes yang tidak terbatas siapapun pembuatnya. Notes yang akan dihapus juga dipertimbangkan melalui Notes yang di-*report* oleh pengguna pada fitur sebelumnya

Alur:

1. Admin dapat menekan tombol hapus pada suatu Notes yang dirasanya melanggar aturan.
2. Notes yang telah dihapus oleh admin akan dihilangkan dari tampilan pengguna.

13. Manage Mata Kuliah

Admin dapat menambahkan, mengubah, dan menghapus mata kuliah dari daftar mata kuliah yang dapat dilihat oleh pengguna di aplikasi OpeNotes.

Alur:

1. Jika admin ingin membuat mata kuliah baru, maka admin perlu memilih menu untuk membuat mata kuliah. Sistem lalu menampilkan *form* yang berisi *fields* yang terkait dengan mata kuliah. admin bisa memilih batal jika ingin membatalkan pembuatan mata kuliah, atau mengisi informasi terkait mata kuliah yang akan dibuka dan memilih menu simpan. Sistem melakukan validasi dari isian formulir. Contoh validasinya adalah apakah SKS yang dimasukkan merupakan angka atau tidak. Jika valid, sistem menyimpan data mata kuliah, menampilkan notifikasi sukses membuat mata kuliah dan kembali menampilkan daftar mata kuliah yang dibuat admin. Jika isian formulir tidak valid, tampilkan pesan gagal dan juga form isian yang gagal disimpan. Jika admin memilih menu batal, maka sistem menampilkan kembali halaman melihat daftar mata kuliah yang dibuat admin. Penyimpanan data mata kuliah tidak dipastikan selalu berhasil. Apabila gagal, maka sistem akan menampilkan formulir kembali.
2. Jika admin ingin mengubah mata kuliah yang sudah pernah dibuatnya sebelumnya, maka admin memilih menu untuk mengubah mata kuliah. Sistem lalu menampilkan *form (editable)* yang berisi data tentang mata kuliah tersebut. Admin bisa memilih batal jika ingin membatalkan perubahan mata kuliah, atau mengubah data pada *fields* sesuai kebutuhan dan memilih menu simpan. Sistem melakukan validasi dari isian formulir. Jika valid, sistem menyimpan data mata kuliah, menampilkan notifikasi sukses mengubah mata kuliah dan kembali menampilkan daftar mata kuliah yang

dibuat admin. Jika isian formulir tidak valid, tampilkan pesan gagal dan juga *form* yang berisi *fields* tadi. Jika admin memilih menu batal, maka sistem menampilkan kembali daftar mata kuliah yang dibuat admin.

3. Jika admin ingin menghapus mata kuliah yang sudah pernah dibuatnya sebelumnya, maka pilih menu hapus pada mata kuliah yang diinginkan. Sistem lalu menampilkan *form* konfirmasi apakah yakin akan menghapus mata kuliah. Pada *form* konfirmasi ini, admin bisa memilih batal atau hapus. Jika admin memilih hapus, maka sistem menghapus mata kuliah, menampilkan notifikasi sukses menghapus, dan menampilkan kembali halaman melihat daftar mata kuliah yang dibuat admin. Jika admin memilih batal, maka sistem hanya menampilkan kembali daftar mata kuliah yang dibuat admin tanpa melakukan perubahan data.

14. *Read User Reports*

Fitur ini memungkinkan admin untuk membaca laporan yang diadukan oleh pengguna atas Notes yang dilihat mereka. Melalui pembacaan laporan ini, mereka dapat memutuskan untuk melakukan tindakan lanjut atas Notes tersebut seperti menghapus Notes. Akan tetapi, laporan ini tidak menjadi persyaratan untuk menghapus Notes.

Alur:

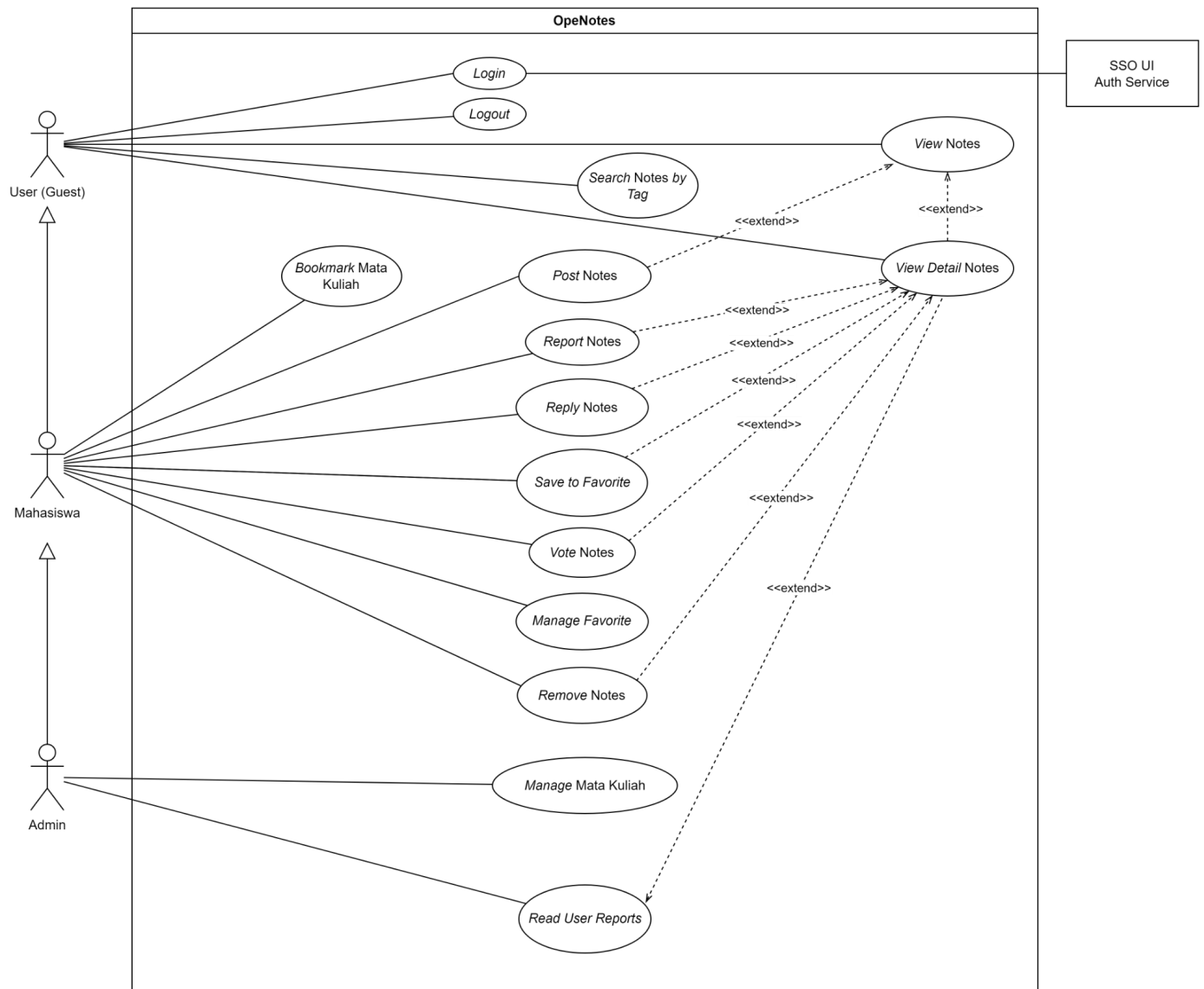
1. Pengguna yang mempunyai *role* sebagai admin membuka dashboard.
2. Selanjutnya dashboard akan menampilkan daftar laporan-laporan beserta tautan kepada Notes yang dilaporkan.
3. Melalui laporan tersebut, admin dapat melakukan pengecekan terhadap Notes tersebut dan menentukan apabila Notes tersebut pantas atau tidak.

No	Requirements TK (tulis satu fitur dalam satu nomor)	Correct	Unambiguous	Complete	Consistent	Verifiable
1	Autentikasi (<i>Login & Logout</i>)	TRUE	TRUE	TRUE	TRUE	TRUE
2	<i>View Notes</i>	TRUE	TRUE	TRUE	TRUE	TRUE
3	<i>View Detail Notes</i>	TRUE	TRUE	TRUE	TRUE	TRUE
4	<i>Post Notes</i>	TRUE	TRUE	TRUE	TRUE	TRUE

5	<i>Bookmark Mata Kuliah</i>	TRUE	TRUE	TRUE	TRUE	TRUE
6	<i>Save to Favorites</i>	TRUE	TRUE	TRUE	TRUE	TRUE
7	<i>Reply Notes</i>	TRUE	TRUE	TRUE	TRUE	TRUE
8	<i>Vote Notes</i>	TRUE	TRUE	TRUE	TRUE	TRUE
9	<i>Search Notes by Tag</i>	TRUE	TRUE	TRUE	TRUE	TRUE
10	<i>Manage Favorite</i>	TRUE	TRUE	TRUE	TRUE	TRUE
11	<i>Report Notes</i>	TRUE	TRUE	TRUE	TRUE	TRUE
12	<i>Remove Notes</i>	TRUE	TRUE	TRUE	TRUE	TRUE
13	<i>Manage Mata Kuliah</i>	TRUE	TRUE	TRUE	TRUE	TRUE
14	<i>Read User Reports</i>	TRUE	TRUE	TRUE	TRUE	TRUE

III. Requirements Modeling

A. Use Case Diagram



B. Use Case Specifications (UCS)

1. UCS 1: *Manage* Mata Kuliah (CRUD)

Author's name	Alvaro Austin
Use Case ID Number	13
Use Case Name	<i>Manage</i> Mata Kuliah
Brief Description	Use case ini memungkinkan admin untuk melakukan operasi CRUD terhadap mata kuliah yang ada pada aplikasi OpeNotes.
Primary Actor	Admin
Secondary Actor	N/A
Pre-condition	Aktor sudah <i>login</i> ke dalam sistem serta sudah terautentikasi sebagai admin.
Post-condition	Perubahan data mata kuliah tersimpan ke dalam sistem dan sistem menampilkan halaman daftar mata kuliah yang telah dikelola sebelumnya.
Parent Use Case	N/A
Included Use Case	N/A
Extension Use Case	N/A

Basic Flow of Events:

Admin	System's Response
1. Jika ingin membuat suatu mata kuliah, maka ke subflow S1. Jika ingin melakukan perubahan terhadap suatu mata kuliah maka ke subflow S2. Jika ingin menghapus suatu mata kuliah, maka ke subflow S3.	
	2. Sistem menampilkan halaman daftar mata kuliah yang telah dikelola sebelumnya.

Subflow S1: Membuat Mata Kuliah

Admin	System's Response
-------	-------------------

1. Memilih menu untuk membuat mata kuliah.	
	2. Menampilkan form berisi <i>fields</i> yang berkaitan dengan mata kuliah.
3. Mengisi informasi mengenai mata kuliah yang akan dibuat.	
4. Memilih menu simpan.	
	5. Melakukan validasi dari isian formulir tersebut. Apabila valid, sistem menyimpan data mata kuliah, lalu menampilkan notifikasi sukses dan kembali ke Basic Flow langkah 2.

Alternative Flow of Events Subflow S1 step 4a (Jika aktor memilih menu batal)

Admin	System's Response
1. Memilih menu batal.	
	2. Kembali ke Basic Flow langkah 2.

Alternative Flow of Events Subflow S1 step 5a (Jika pengisian formulir tidak valid)

Admin	System's Response
	1. Penampilan pesan gagal terhadap validasi input formulir.
	2. Kembali ke Subflow S1 langkah 2.

Alternative Flow of Events Subflow S1 step 5b (Jika sistem gagal menyimpan data mata kuliah)

Admin	System's Response
	1. Penampilan pesan gagal terhadap penyimpanan data mata kuliah.
	2. Kembali ke Subflow S1 langkah 2.

Subflow S2: Mengubah Mata Kuliah

Admin	System's Response
1. Memilih menu mengubah mata kuliah.	
	2. Menampilkan form (<i>editable</i>) berisi data tentang mata kuliah tersebut.

3. Mengubah data pada <i>fields</i> sesuai kebutuhan admin.	
4. Memilih menu simpan.	
	5. Melakukan validasi dari isian formulir tersebut. Apabila valid, sistem menyimpan data mata kuliah, lalu menampilkan notifikasi sukses dan kembali ke Basic Flow langkah 2.

Alternative Flow of Events Subflow S2 step 4a (Jika aktor memilih menu batal)

Admin	System's Response
1. Memilih menu batal.	
	2. Kembali ke Basic Flow langkah 2.

Alternative Flow of Events Subflow S2 step 5a (Jika pengisian formulir tidak valid)

Admin	System's Response
	1. Penampilan pesan gagal terhadap validasi input formulir.
	2. Kembali ke Subflow S2 langkah 2.

Alternative Flow of Events Subflow S2 step 5b (Jika sistem gagal menyimpan data mata kuliah)

Admin	System's Response
	1. Penampilan pesan gagal terhadap penyimpanan data mata kuliah.
	2. Kembali ke Subflow S2 langkah 2.

Subflow S3: Menghapus Mata Kuliah

Admin	System's Response
1. Memilih menu untuk hapus mata kuliah tertentu.	
	2. Menampilkan form konfirmasi apakah yakin akan menghapus mata kuliah tersebut.
3. Jika yakin, maka hapus.	
	4. Menghapus data mata kuliah tersebut dari dalam sistem, menampilkan notifikasi sukses menghapus, dan kembali ke Basic

	Flow langkah 2.
Alternative Flow of Events Subflow S3 step 3a (Jika aktor memilih menu batal)	
Admin	System's Response
1. Memilih menu batal.	
	2. Kembali ke Basic Flow langkah 2.
Alternative Flow of Events Subflow S3 step 4a (Jika sistem gagal menghapus mata kuliah)	
Admin	System's Response
	1. Penampilan pesan gagal terhadap penghapusan mata kuliah.
	2. Kembali ke Subflow S3 langkah 2.

2. UCS 2: View Notes (Read)

Author's name	Alvaro Austin
Use Case ID Number	2
Use Case Name	View Notes
Brief Description	Use case ini merupakan alur proses User (<i>Guest</i>) untuk melihat kumpulan Notes awal yang telah di-post oleh Mahasiswa ataupun Admin terkait suatu mata kuliah tertentu.
Primary Actor	User (<i>Guest</i>)
Secondary Actor	N/A
Pre-condition	N/A
Post-condition	Sistem menampilkan kumpulan Notes awal yang telah tersimpan terkait suatu mata kuliah.
Parent Use Case	N/A
Included Use Case	N/A
Extension Use Case	<ul style="list-style-type: none"> • <i>View Detail Notes</i> • <i>Post Notes</i>

Basic Flow of Events:

User (<i>Guest</i>)	System's Response
----------------------------	--------------------------

1. Memilih menu “ Memilih Mata Kuliah ”.	
	2. Sistem menampilkan daftar mata kuliah yang sudah tersedia.
3. Memilih suatu mata kuliah yang telah ditampilkan pada langkah ke-2.	
	4. Sistem menampilkan kumpulan Notes awal yang terkait dengan mata kuliah yang telah dipilih pada langkah ke-2
5. Jika ingin melihat detail dari suatu Notes, maka lanjut ke Use Case View Detail Notes . Jika ingin meng-post suatu Notes, maka lanjut ke Use Case Post Notes .	

Asumsi:

- Pada Basic Flow dari **Use Case View Detail Notes**, post condition dari use case ini tetap terpenuhi pada tahap ke-4. Tahap ke-5 hanya merupakan tahap kondisional untuk opsi berpindah pada **Use Case Extension**.
- Primary Actor pada Use Case ini merupakan User (*Guest*) yang berarti karena Mahasiswa dan Admin merupakan subclass dari User (*Guest*) maka Use Case ini juga berlaku untuk Mahasiswa dan Admin pula secara tidak langsung

3. UCS 3: *View Detail Notes*

Author's name	Christopher Nathanael Wijaya
Use Case ID Number	3
Use Case Name	<i>View Detail Notes</i>
Brief Description	<i>Use case</i> ini merupakan alur proses User (<i>Guest</i>) untuk melihat detail dari suatu Notes. Dalam hal ini, aktor dapat melihat kumpulan <i>reply</i> dari suatu Notes yang telah dipilih sebelumnya.
Primary Actor	User (<i>Guest</i>)
Secondary Actor	N/A
Pre-condition	N/A
Post-condition	Sistem menampilkan detail dari suatu Notes yang dipilih oleh aktor.
Parent Use Case	N/A

Included Use Case	N/A
Extension Use Case	<ul style="list-style-type: none"> • <i>Report Notes</i> • <i>Reply Notes</i> • <i>Save to Favorite</i> • <i>Vote Notes</i> • <i>Remove Notes</i>

Basic Flow of Events:

User (Guest)	System's Response
1. Jika User (<i>Guest</i>) tidak memiliki tautan terhadap suatu <i>detail</i> Notes Melaksanakan Use Case View Notes .	
2. Memilih suatu Notes dari kumpulan Notes yang telah ditampilkan pada langkah ke-1.	
	3. Menampilkan <i>Detail</i> Notes beserta kumpulan <i>reply</i> Notes yang berkaitan.
<p>4. Jika ingin me-<i>report</i> suatu Notes, maka lanjut ke Use Case Report Notes.</p> <p>Jika ingin me-<i>reply</i> suatu Notes, maka lanjut ke Use Case Reply Notes.</p> <p>Jika ingin meng-<i>save</i> suatu Notes pada folder <i>favorite</i>, maka lanjut ke Use Case Save to Favorite.</p> <p>Jika ingin meng-<i>vote</i> suatu Notes, maka lanjut ke Use Case Vote Notes.</p> <p>Jika ingin me-<i>remove</i> Notes, maka lanjut ke Use Case Remove Notes.</p>	
Alternative Flow: untuk step 1a (Jika aktor memiliki tautan Notes, maka aktor dapat langsung mengakses detail Notes tersebut)	
User (Guest)	System's Response
1. Mengakses tautan Notes yang telah dimiliki.	
	2. Menampilkan Notes yang sesuai dengan tautan yang diakses oleh aktor.

Asumsi:

- Pada Basic Flow dari **Use Case View Detail Notes**, post condition dari use case ini tetap terpenuhi pada tahap ke-3. Tahap ke-4 hanya merupakan tahap kondisional untuk opsi berpindah pada **Use Case Extension**.
- Primary Actor pada Use Case ini merupakan User (*Guest*) yang berarti karena Mahasiswa dan Admin merupakan subclass dari User (*Guest*) maka Use Case ini juga berlaku untuk Mahasiswa dan Admin pula secara tidak langsung

4. UCS 4: *Post Notes*

Author's name	Christopher Nathanael Wijaya
Use Case ID Number	4
Use Case Name	<i>Post Notes</i>
Brief Description	Use case ini merupakan alur proses Mahasiswa untuk membuat suatu Notes baru pada suatu mata kuliah tertentu.
Primary Actor	Mahasiswa
Secondary Actor	N/A
Pre-condition	Aktor sudah login ke dalam sistem, dan berada di halaman daftar Notes awal pada suatu mata kuliah yang dipilih sebelumnya.
Post-condition	Notes yang dibuat oleh aktor tersimpan pada sistem, dan sistem menampilkan halaman daftar Notes pada suatu mata kuliah yang mengandung Notes yang baru dibuat oleh aktor.
Parent Use Case	N/A
Included Use Case	N/A
Extension Use Case	N/A

Basic Flow of Events:

Mahasiswa	System's Response
1. Memilih menu " Post Notes ".	
	2. Menampilkan <i>form</i> untuk pembuatan Notes.
3. Mengisi informasi Notes pada <i>form</i> .	
4. Memilih menu " Unggah ".	
	5. Melakukan validasi dari isian <i>form</i> . Jika valid, sistem menyimpan Notes dan

	menampilkan halaman daftar Notes yang baru.
Alternative Flow untuk step 5a (Jika <i>form</i> yang diisi tidak valid sehingga gagal menyimpan, contoh kasus tidak valid: file yang dikirim terlalu besar, dll)	
	1. Menampilkan notifikasi bahwa <i>form</i> tidak valid dan kembali ke langkah 2 pada <i>Basic Flow</i>

5. UCS 5: *Manage Favorite*

Author's name	Febrian Dwi Kimhan
Use Case ID Number	10
Use Case Name	<i>Manage Favorite</i>
Brief Description	Use case ini menjelaskan alur aktor dalam mengelola folder-folder favoritnya
Primary Actor	Mahasiswa
Secondary Actor	N/A
Pre-condition	Aktor sudah <i>login</i> ke dalam sistem dan telah berada di halaman utama OpeNotes.
Post-condition	Pengubahan data folder favorit tersimpan di dalam sistem. Sistem menampilkan halaman daftar folder favorit yang pernah dibuat oleh aktor.
Parent Use Case	N/A
Included Use Case	N/A
Extension Use Case	N/A

Basic Flow of Events:

Mahasiswa	System's Response
1. Memilih menu " <i>Manage Favorite</i> ".	
	2. Menampilkan daftar folder favorit yang pernah dibuat oleh aktor.
3. Jika ingin membuat folder favorit baru, maka ke subflow S1.	

<p>Jika ingin mengubah folder favorit, maka ke subflow S2.</p> <p>Jika ingin menghapus folder favorit, maka ke subflow S3.</p>	
--	--

Subflow S1: Membuat Folder Favorit Baru

Mahasiswa	System's Response
1. Memilih menu untuk membuat folder favorit baru.	
	2. Menampilkan form judul folder yang akan dibuat yang harus diisi.
3. Mengisi judul folder yang dibuat.	
	4. Menyimpan data folder favorit baru, menampilkan notifikasi sukses membuat folder favorit baru, dan kembali ke Basic Flow langkah 2.
Alternative Flow of Events Subflow S1 untuk step 3a (jika aktor memilih menu batal):	
Mahasiswa	System's Response
	1. Folder favorit tidak dibuat dan kembali ke Basic Flow langkah 2

Subflow S2: Mengubah Folder Favorit

Mahasiswa	System's Response
1. Memilih menu untuk mengubah folder favorit pada folder yang ingin diubah.	
	2. Menampilkan form judul folder yang bisa diedit dan tidak bisa kosong.
3. Mengubah judul folder.	
	4. Menyimpan data folder favorit yang sudah diubah, menampilkan notifikasi sukses mengubah folder favorit, dan kembali ke Basic Flow langkah 2.
Alternative Flow of Events Subflow S2 untuk step 3a (jika aktor memilih menu batal):	

Mahasiswa	System's Response
	1. Tidak menyimpan data folder favorit yang diubah dan kembali ke Basic Flow langkah 2.

Subflow S3: Menghapus Folder Baru

Mahasiswa	System's Response
1. Memilih menu untuk menghapus folder favorit pada folder yang ingin dihapus.	
	2. Menampilkan pesan konfirmasi apakah aktor yakin untuk menghapus folder favorit.
3. Memilih menu "Ya".	
	4. Menghapus folder favorit, menampilkan notifikasi sukses menghapus folder favorit, dan kembali ke Basic Flow langkah 2.
Alternative Flow of Events Subflow S3 untuk step 3a (jika aktor memilih menu batal):	
Mahasiswa	System's Response
	1. Tidak menghapus folder favorit dan kembali ke Basic Flow langkah 2

6. UCS 6: Save to Favorite

Author's name	Febrian Dwi Kimhan
Use Case ID Number	6
Use Case Name	<i>Save to Favorite</i>
Brief Description	Use case ini menjelaskan alur aktor dalam dalam menambahkan Notes ke satu atau lebih folder favorit.
Primary Actor	Mahasiswa
Secondary Actor	N/A
Pre-condition	Aktor sudah login ke dalam sistem dan telah menjalankan use case " View Notes ".
Post-condition	Notes berhasil tersimpan ke dalam satu atau lebih folder favorit yang dipilih.

Parent Use Case	N/A
Included Use Case	N/A
Extension Use Case	N/A

Basic Flow of Events:

Mahasiswa	System's Response
1. Memilih salah satu Notes dan memilih menu untuk menyimpan ke folder favorit.	
	2. Menampilkan daftar folder favorit yang sudah dibuat.
3. Memilih satu atau lebih folder sebagai tempat menyimpan Notes dan memilih menu simpan.	
	4. Menyimpan Notes ke folder-folder yang dipilih, menampilkan notifikasi sukses menyimpan Notes, dan kembali ke halaman " View Notes ".
Alternative Flow untuk step 3a (Jika aktor memilih menu batal)	
Mahasiswa	System's Response
	1. Tidak menyimpan Notes ke folder-folder yang dipilih dan kembali ke halaman " View Notes ".

7. UCS 7: Report Notes

Author's name	Muflih Naufal Maxi
Use Case ID Number	11
Use Case Name	<i>Report Notes</i>
Brief Description	<i>Use case ini merupakan alur proses mahasiswa melaporkan (me-report) Notes mahasiswa lain yang dirasa tidak sesuai karena alasan tertentu.</i>
Primary Actor	Mahasiswa
Secondary Actor	N/A

Pre-condition	Aktor sudah melakukan <i>login</i> dan telah berada di halaman Detail Notes tertentu.
Post-condition	<i>Report</i> yang dikirim Mahasiswa disimpan dalam sistem, dan Mahasiswa menerima notifikasi bahwa <i>report</i> berhasil dikirim.
Parent Use Case	N/A
Included Use Case	N/A
Extension Use Case	N/A

Basic Flow of Events:

Mahasiswa	System's Response
1. Memilih menu <i>report</i> pada Notes yang sedang dibuka.	
	2. Menampilkan halaman untuk mengisi detail <i>report</i> .
3. Mengisi detail <i>report</i> yang berisi kategori <i>report</i> dan alasan <i>report</i> .	
	4. Menampilkan form konfirmasi apakah mahasiswa yakin akan mengirim <i>report</i> .
5. Jika yakin, maka pilih "Kirim".	
	6. Melakukan validasi dari isian formulir. Jika valid, sistem menyimpan <i>report</i> yang dikirim mahasiswa dan menampilkan notifikasi kepada mahasiswa.

Alternative Flow untuk step 5a (Jika aktor memilih menu batal):

Mahasiswa	System's Response
1. Memilih menu batal.	
	2. Menampilkan kembali halaman Notes sebelumnya.

Alternative Flow untuk step 6a (Jika detail report yang diisi user tidak valid):

Mahasiswa	System's Response
	1. Menampilkan notifikasi bahwa kategori/alasan <i>report</i> yang diberikan mahasiswa tidak valid.

8. UCS 8: *Read User Reports*

Author's name	Muflih Naufal Maxi
Use Case ID Number	14
Use Case Name	<i>Read User Reports</i>
Brief Description	<i>Use case ini merupakan alur proses admin membaca dan me-review report yang diberikan mahasiswa.</i>
Primary Actor	Admin
Secondary Actor	N/A
Pre-condition	Aktor sudah melakukan <i>login</i> serta sudah terautentikasi sebagai admin.
Post-condition	<i>Report mahasiswa diterima dan Notes yang di-report dihapus dari sistem.</i>
Parent Use Case	N/A
Included Use Case	N/A
Extension Use Case	N/A

Basic Flow of Events:

Admin	System's Response
1. Memilih menu " daftar user reports yang belum di-review ".	
	2. Menampilkan daftar <i>user reports</i> yang belum di-review.
3. Memilih <i>report</i> yang ingin dilihat detailnya.	
	4. Menampilkan detail <i>report</i> yang berisi <i>username</i> mahasiswa, tautan Notes yang di-report, kategori <i>report</i> , dan alasan <i>report</i> .
5. Mengakses tautan Notes untuk di-review.	
6. Menandai <i>report</i> sebagai <i>report</i> yang valid.	

	7. Memberikan status “Reviewed” pada <i>report</i> , menghapus Notes yang di- <i>report</i> , dan menampilkan notifikasi kepada admin.
Alternative Flow untuk step 6a (Jika <i>report</i> yang diterima bersifat <i>invalid</i>)	
Admin	System’s Response
1. Menandai <i>report</i> sebagai <i>report</i> yang tidak valid.	
	2. Memberikan status “Reviewed” pada <i>report</i> dan menampilkan notifikasi kepada admin.

9. UCS 9: Remove Notes

Author’s name	Muflih Naufal Maxi
Use Case ID Number	12
Use Case Name	<i>Remove Notes</i>
Brief Description	<i>Use case ini memungkinkan mahasiswa yang sudah terautentikasi untuk menghapus Notes yang dapat dihapus mahasiswa tersebut.</i>
Primary Actor	Mahasiswa
Secondary Actor	N/A
Pre-condition	Aktor sudah <i>login</i> ke dalam sistem dan telah berada di halaman Detail Notes tertentu.
Post-condition	Sistem menghapus Notes yang ingin dihapus aktor, dan aktor mendapat <i>feedback</i> yang menyatakan Notes nya sudah berhasil dihapus.
Parent Use Case	N/A
Included Use Case	N/A
Extension Use Case	N/A

Basic Flow of Events:

Mahasiswa	System’s Response
1. Memilih menu <i>delete</i> pada Notes yang sedang dibuka.	

	2. Menampilkan form konfirmasi apakah yakin akan menghapus Notes.
3. Jika yakin, maka pilih “Hapus”.	
	4. Menghapus Notes yang ingin dihapus mahasiswa, memberikan <i>feedback</i> bahwa Notes sudah berhasil dihapus, dan menampilkan kembali halaman daftar Notes pada mata kuliah tersebut.
Alternative Flow untuk step 3a (Jika aktor memilih menu batal):	
Mahasiswa	System's Response
1. Memilih menu batal.	
	2. Menampilkan kembali halaman Notes sebelumnya.

10. UCS 10: Vote Notes

Author's name	Fernando Nathaniel Sutanto
Use Case ID Number	8
Use Case Name	Vote Notes
Brief Description	<i>Use case</i> ini memungkinkan aktor untuk memberikan <i>upvote</i> atau <i>downvote</i> pada Notes yang ada baik yang dimiliki diri dia sendiri atau pengguna lain.
Primary Actor	Mahasiswa
Secondary Actor	N/A
Pre-condition	Aktor sudah <i>login</i> ke dalam sistem dan telah berada di halaman <i>View Notes</i> mata kuliah tertentu.
Post-condition	Sistem menyimpan perubahan <i>vote</i> dan aktor dapat melihat <i>feedback</i> yang menyatakan pilihannya sudah tercatat.
Parent Use Case	N/A
Included Use Case	N/A
Extension Use Case	N/A

Basic Flow of Events:

Mahasiswa	System's Response
1. Jika ingin membuat <i>vote</i> untuk pertama kali, maka ke subflow S1. Jika ingin mengubah pilihan <i>vote</i> , maka ke subflow S2. Jika ingin menghapus <i>vote</i> yang sudah ada, maka ke subflow S3.	
	2. Memberikan <i>feedback</i> bahwa aksinya sudah tercatat.

Subflow S1: Membuat Vote

Mahasiswa	System's Response
1. Memilih menu <i>upvote</i> atau <i>downvote</i> pada Notes yang diinginkan.	
	2. Menyimpan data <i>vote</i> .

Alternative Flow of Events Subflow S1 step 2a (Jika sistem gagal menyimpan data *vote*)

Admin	System's Response
	1. Penampilan pesan gagal terhadap penyimpanan data <i>vote</i> .
	2. Kembali ke Subflow S1 langkah 1.

Subflow S2: Mengubah Vote

Mahasiswa	System's Response
1. Memilih menu yang menjadi lawan dari yang sudah di- <i>vote</i> sebelumnya pada Notes yang diinginkan.	
	2. Menyimpan data <i>vote</i> .

Alternative Flow of Events Subflow S2 step 2a (Jika sistem gagal menyimpan data *vote*)

Admin	System's Response
	1. Penampilan pesan gagal terhadap penyimpanan data <i>vote</i> .

	2. Kembali ke Subflow S2 langkah 1.
--	-------------------------------------

Subflow S3: Menghapus Vote

Mahasiswa	System's Response
1. Memilih menu yang sama dengan yang sudah di-vote sebelumnya pada Notes yang diinginkan	
	2. Menyimpan data vote.
Alternative Flow of Events Subflow S3 step 2a (Jika sistem gagal menyimpan data vote)	
Admin	System's Response
	1. Penampilan pesan gagal terhadap penyimpanan data vote.
	2. Kembali ke Subflow S3 langkah 1.

11. UCS 11: Reply Notes

Author's name	Fernando Nathaniel Sutanto
Use Case ID Number	7
Use Case Name	Reply Notes
Brief Description	Use case ini memungkinkan mahasiswa untuk menanggapi Notes yang sudah ada dengan memberikan Notes yang dibuat oleh mahasiswa tersebut atau mahasiswa lain.
Primary Actor	Mahasiswa
Secondary Actor	N/A
Pre-condition	Aktor sudah <i>login</i> ke dalam sistem dan telah berada di halaman Detail Notes tertentu.
Post-condition	Sistem menyimpan Notes yang dibuat aktor dan aktor dapat melihat <i>feedback</i> yang menyatakan bahwa Notes miliknya sudah tersimpan.
Parent Use Case	N/A
Included Use Case	N/A

Extension Use Case	N/A
--------------------	-----

Basic Flow of Events:

Mahasiswa	System's Response
1. Memilih menu <i>reply</i> pada Notes yang ingin ditanggapi.	
	2. Menampilkan <i>forms</i> yang berisi <i>fields</i> yang terkait dengan <i>reply</i> Notes.
3. Mengisi informasi terkait <i>reply Notes</i> dan memilih menu simpan.	
	4. Memberikan <i>feedback</i> bahwa Notes sudah berhasil disimpan.

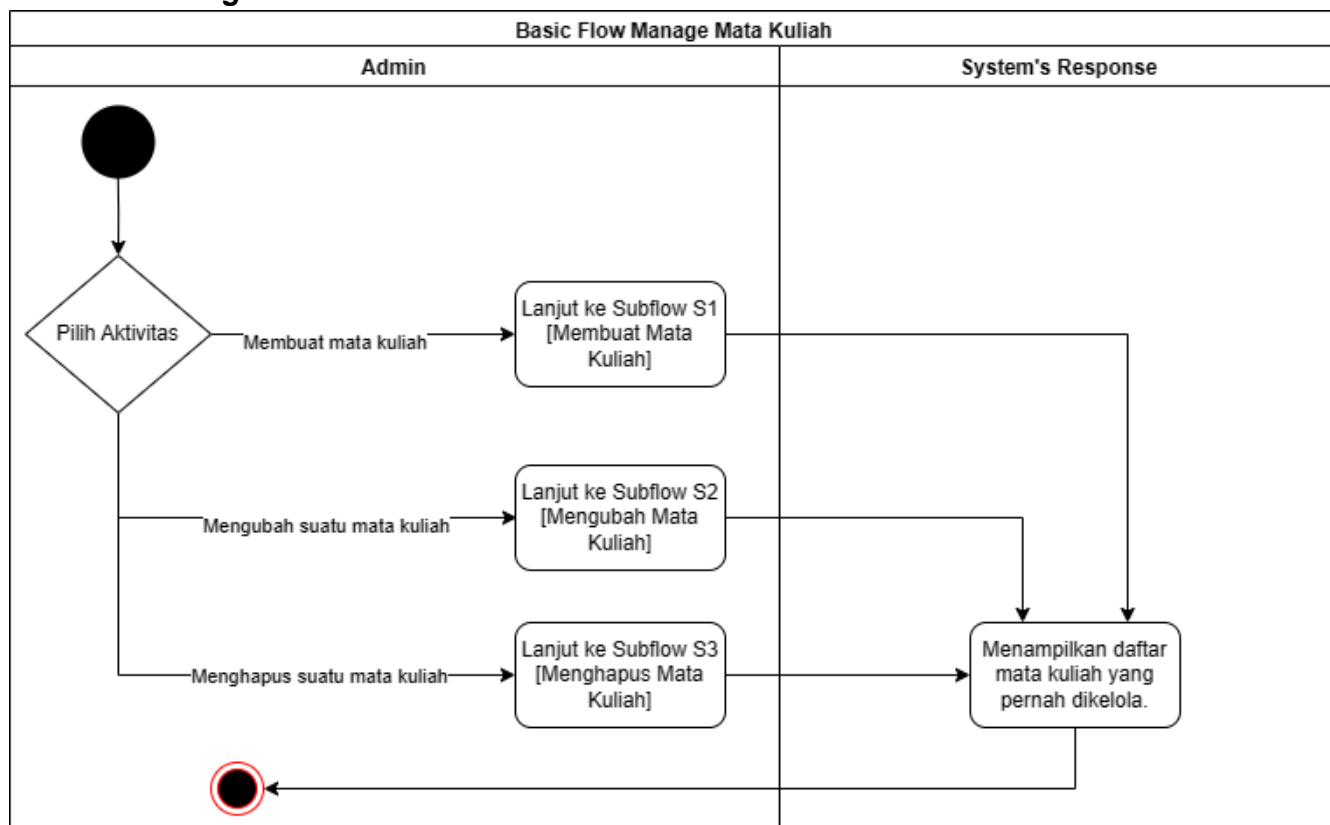
Alternative Flow of Events Basic Flow step 4a (Jika sistem gagal menyimpan data *reply Notes*)

Admin	System's Response
	1. Penampilan pesan gagal terhadap penyimpanan data <i>reply Notes</i> .
	2. Kembali ke Basic Flow langkah 2.

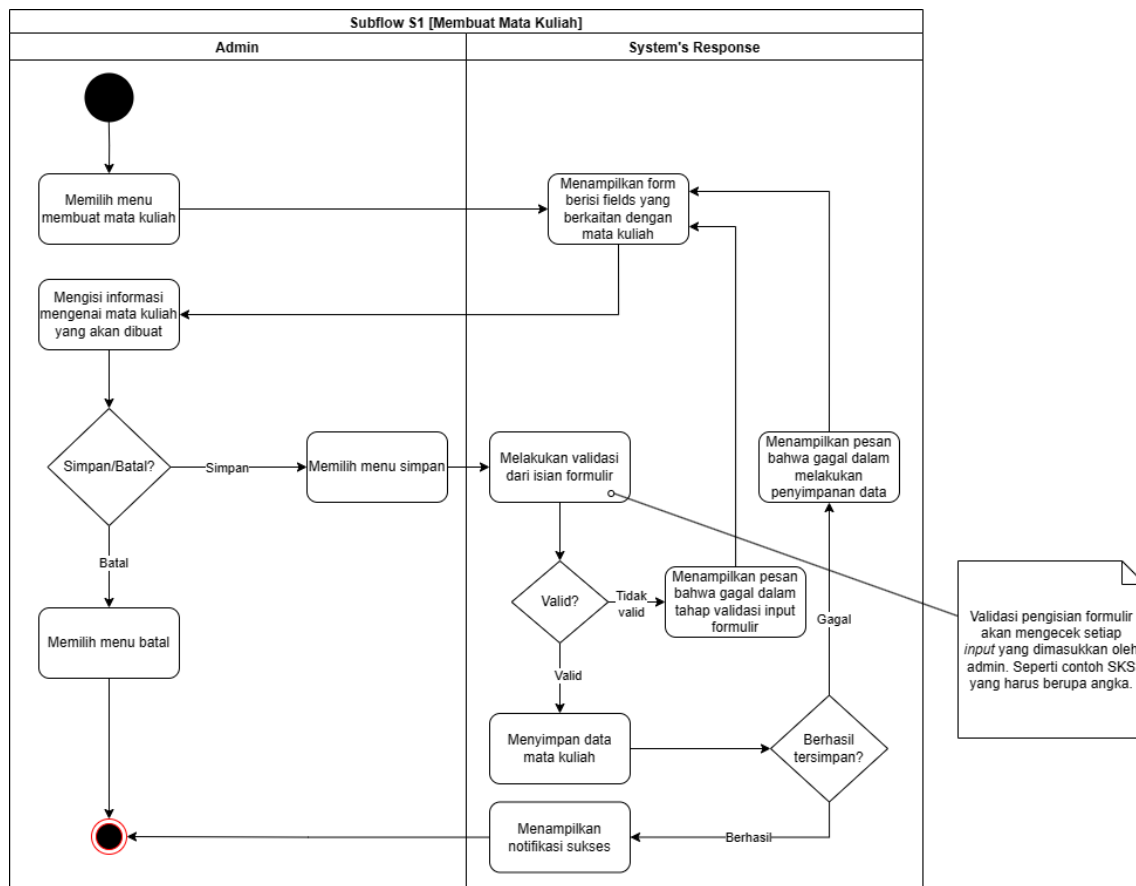
C. Use Case Activity Diagrams

1. Activity Diagram for UCS 1: *Manage Mata Kuliah* (Alvaro)

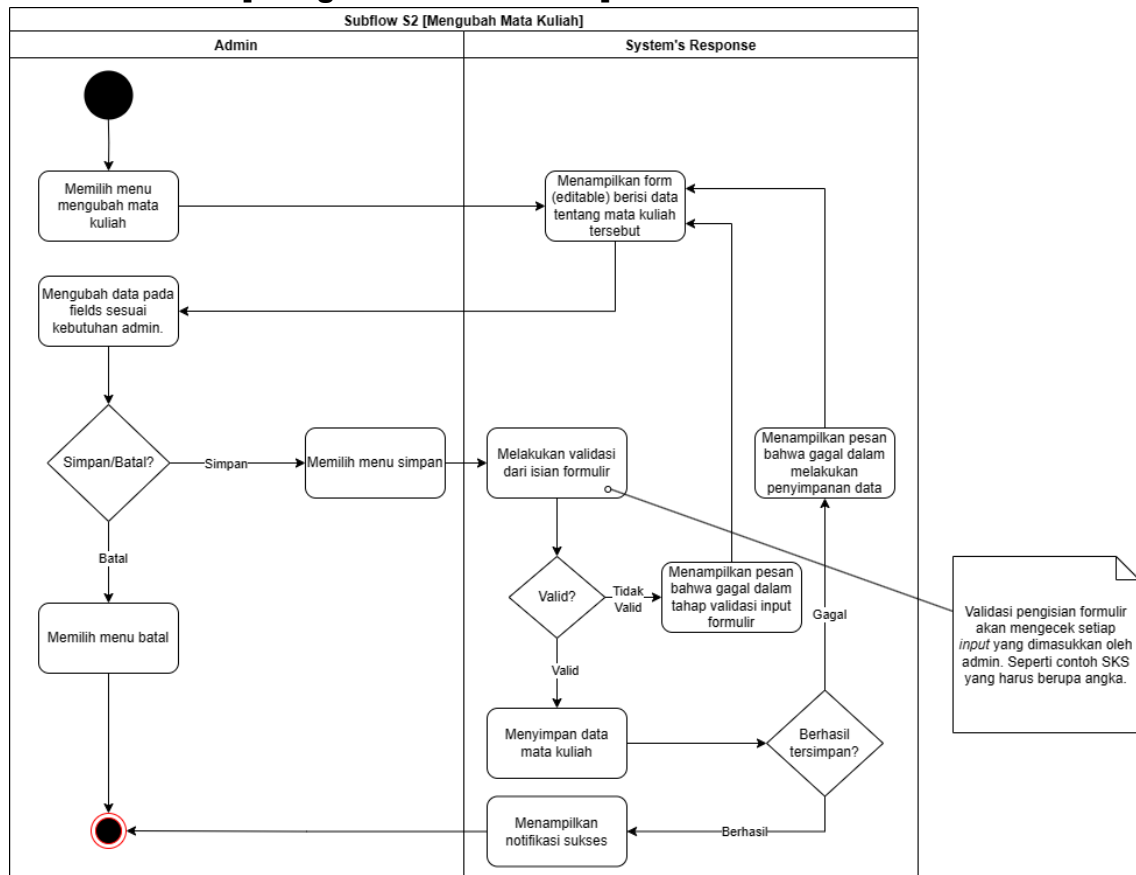
AD Basic Flow *Manage Mata Kuliah*



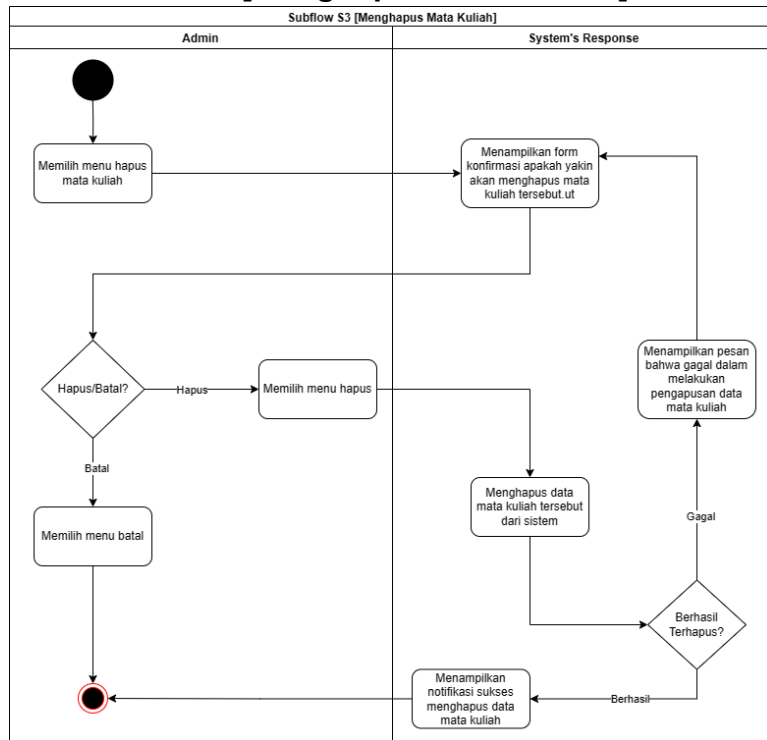
I. AD Subflow S1 [Membuat Mata Kuliah]



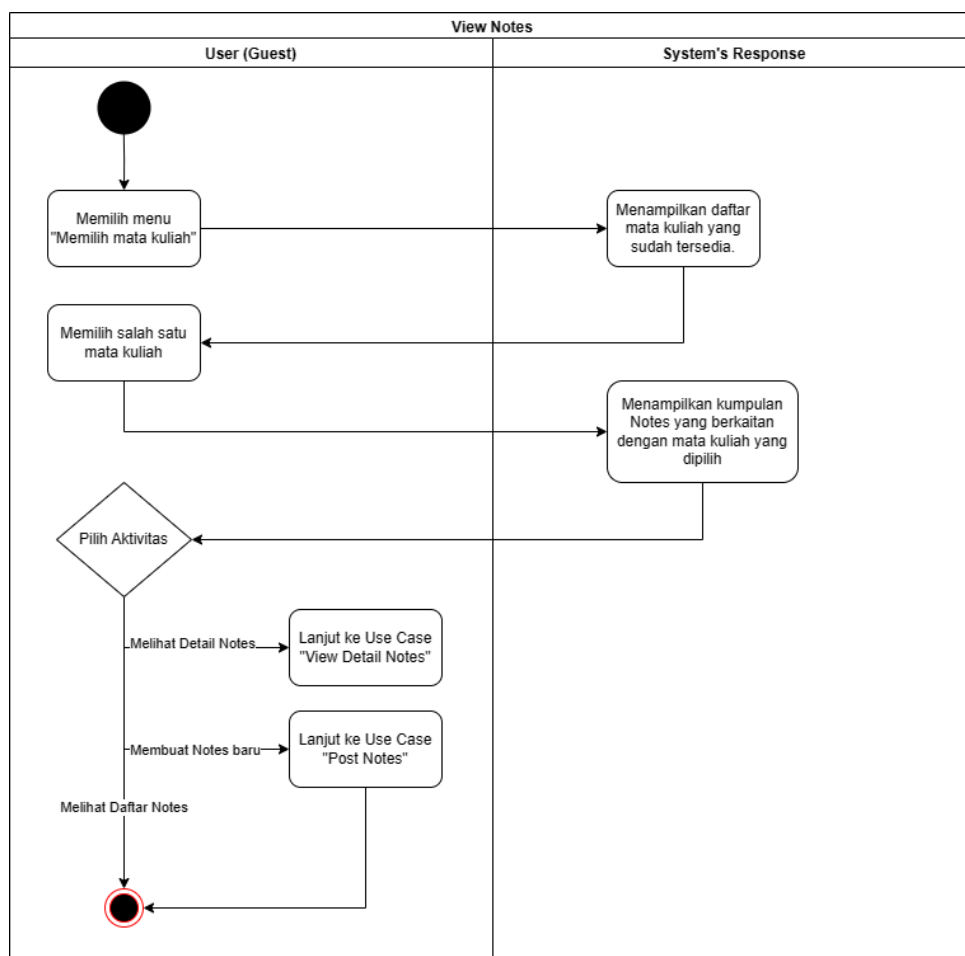
II. AD Subflow S2 [Mengubah Mata Kuliah]



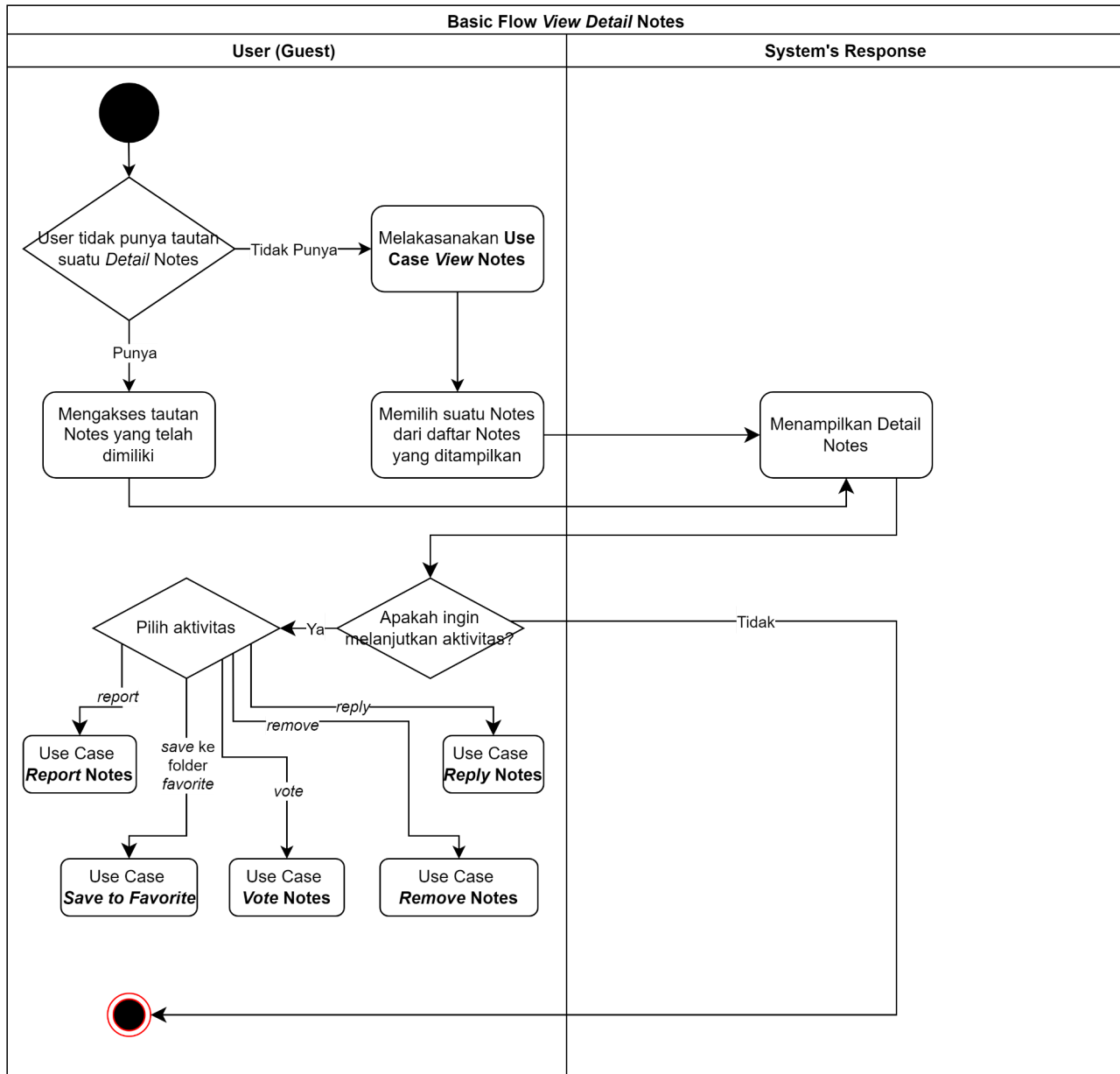
III. AD Subflow S3 [Menghapus Mata Kuliah]



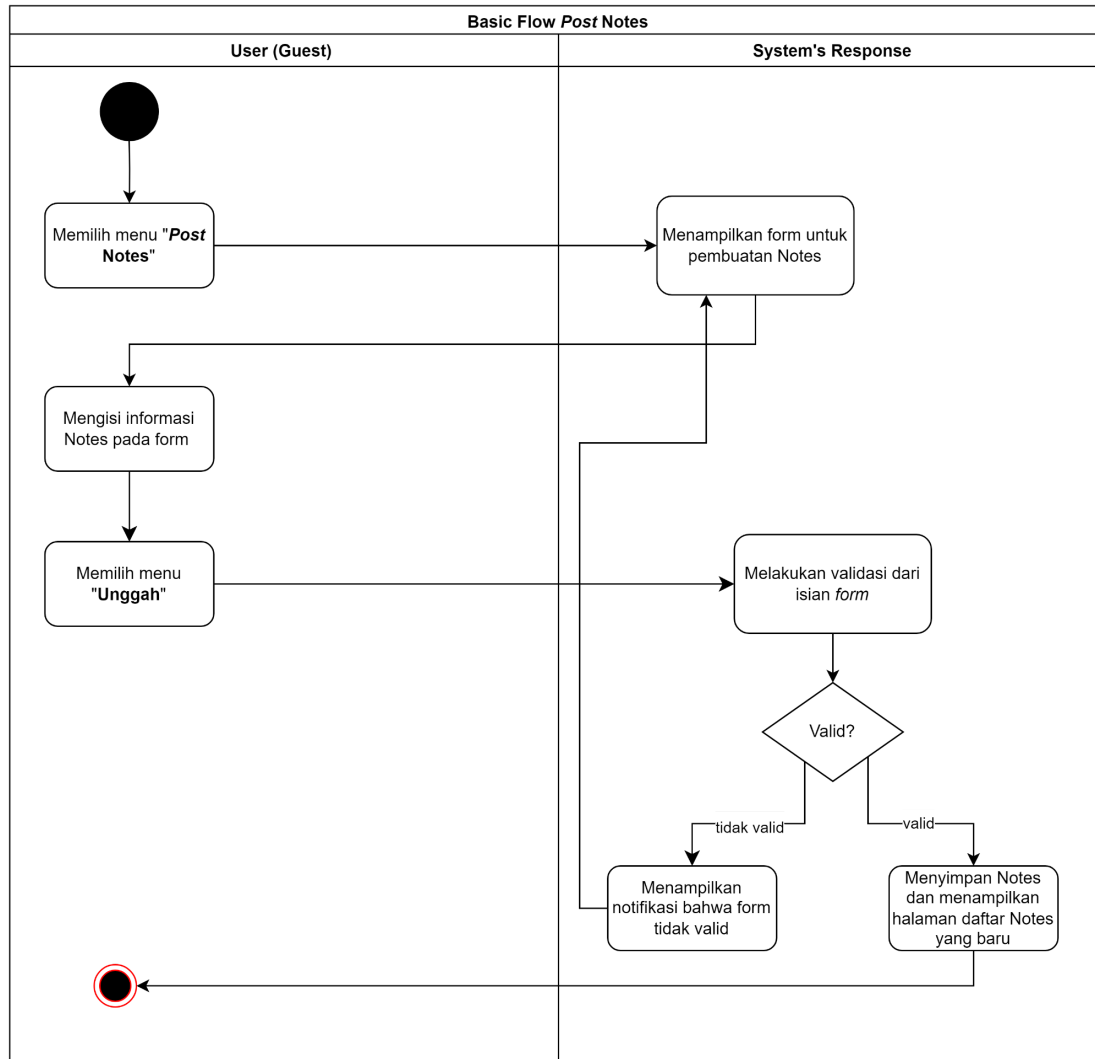
2. Activity Diagram for UCS 2: View Notes (Alvaro)



3. Activity Diagram for UCS 3: *View Detail Notes* (Christo)

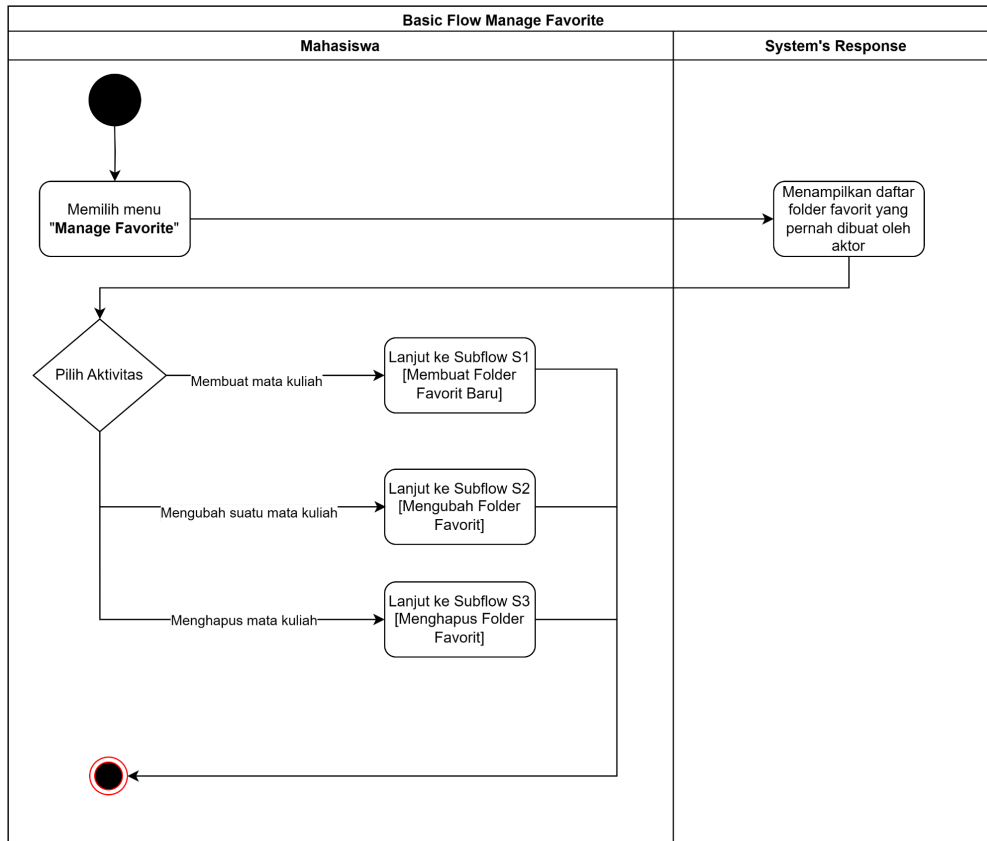


4. Activity Diagram for UCS 4: *Post Notes* (Christo)

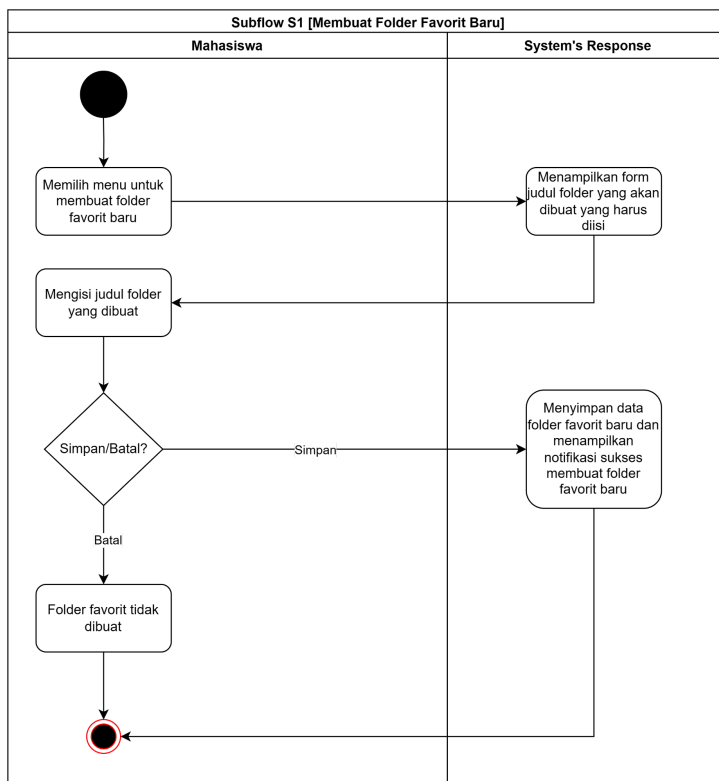


5. Activity Diagram for UCS 5: *Manage Favorite* (Febrian)

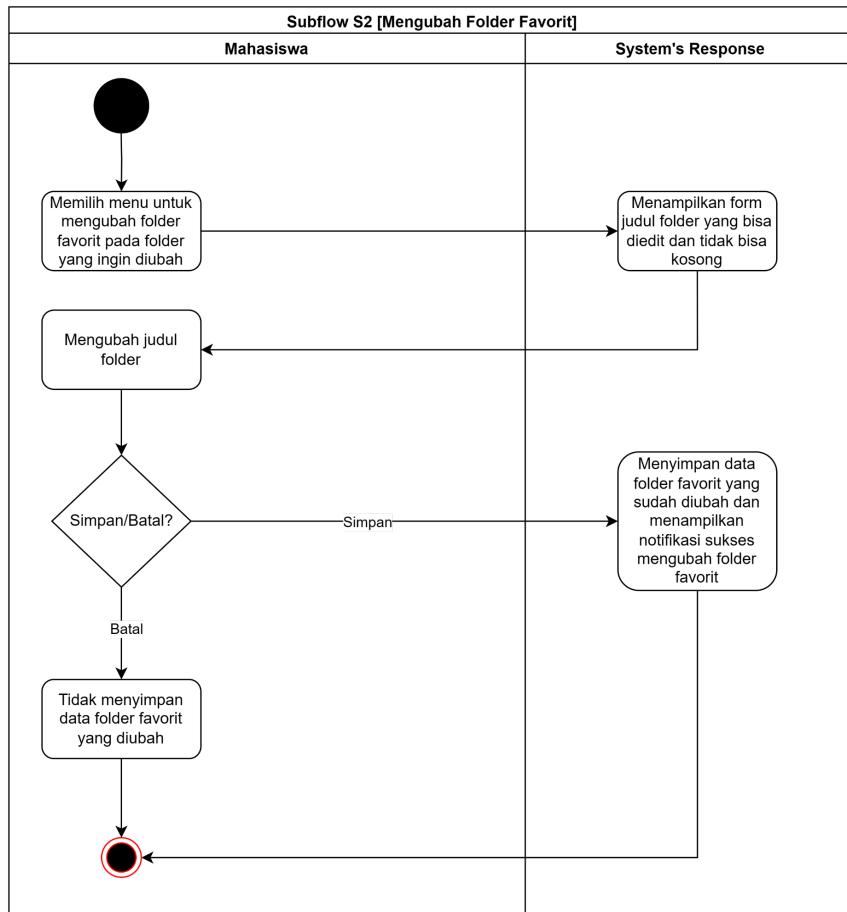
Basic Flow Manage Favorite



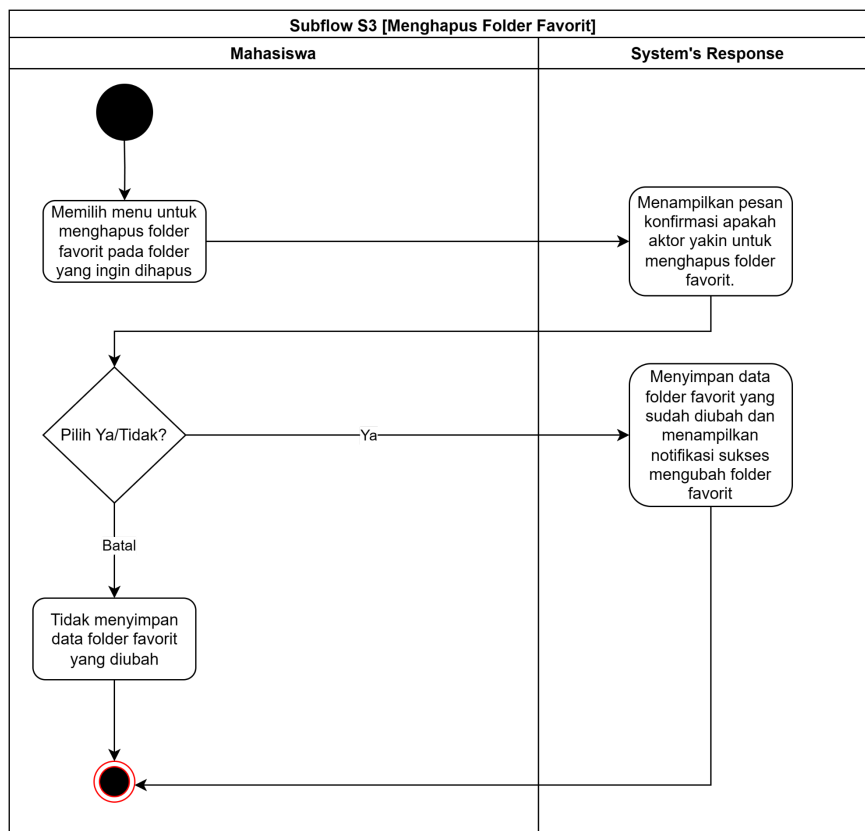
I. AD Subflow S1 [Membuat Folder Favorit Baru]



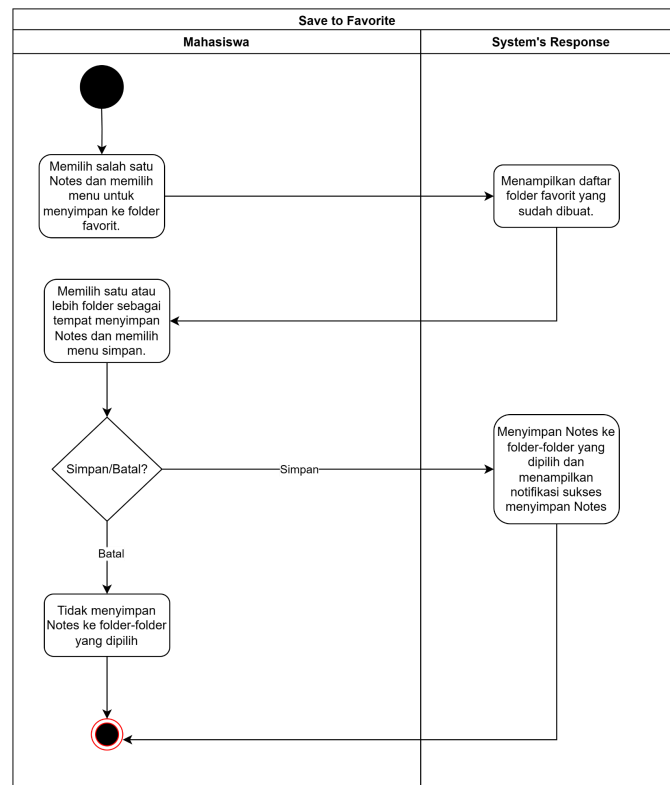
II. AD Subflow S2 [Mengubah Folder Favorit]



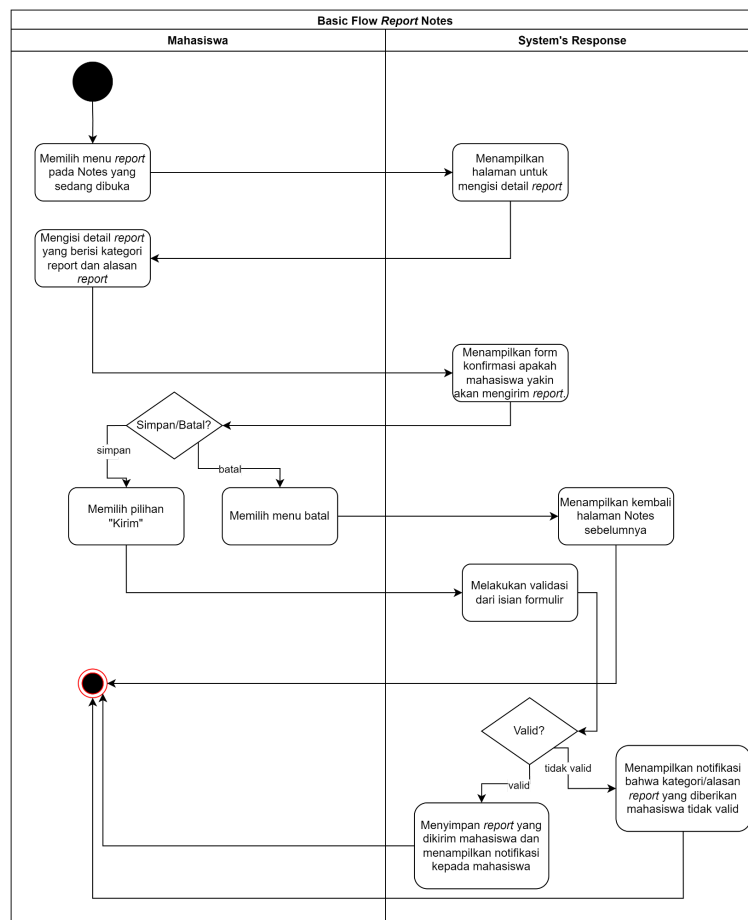
III. AD Subflow S3 [Menghapus Folder Favorit]



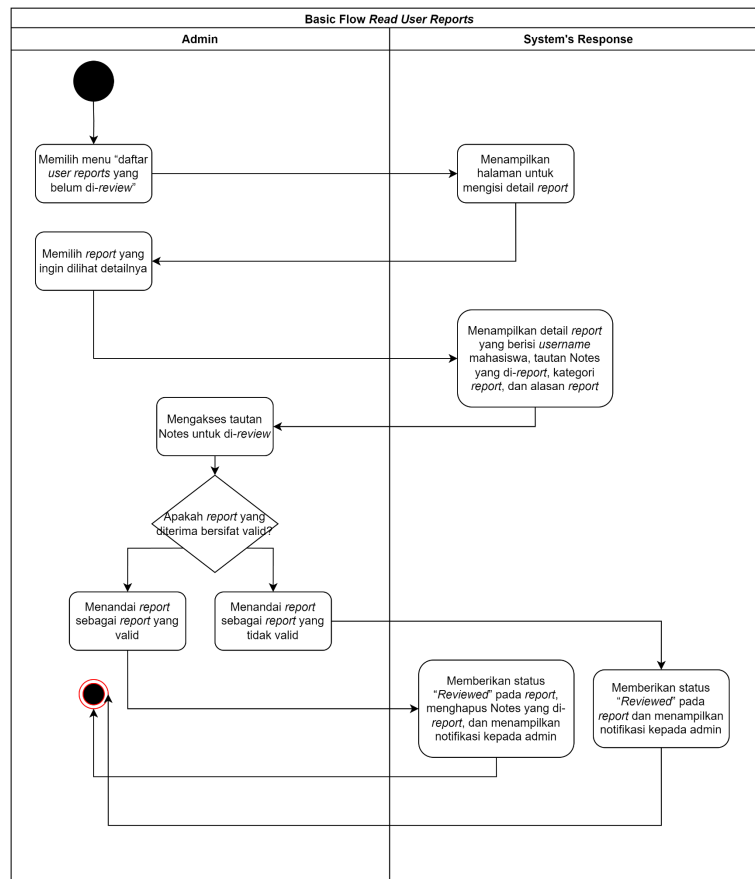
6. Activity Diagram for UCS 6: *Save to Favorite* (Febrian)



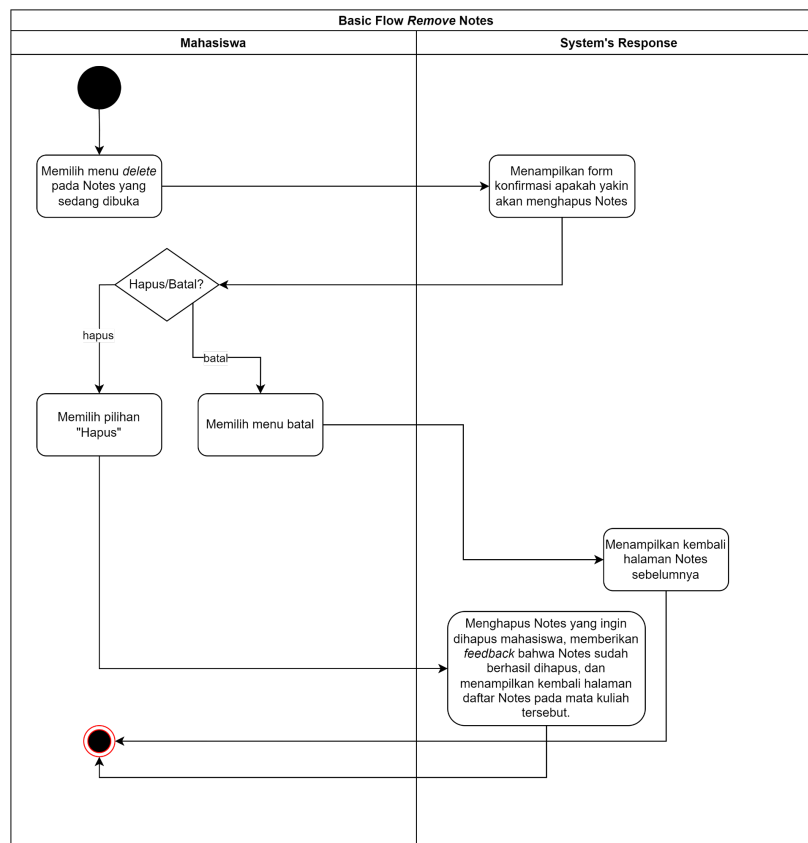
7. Activity Diagram for UCS 7: *Report Notes* (Muflih)



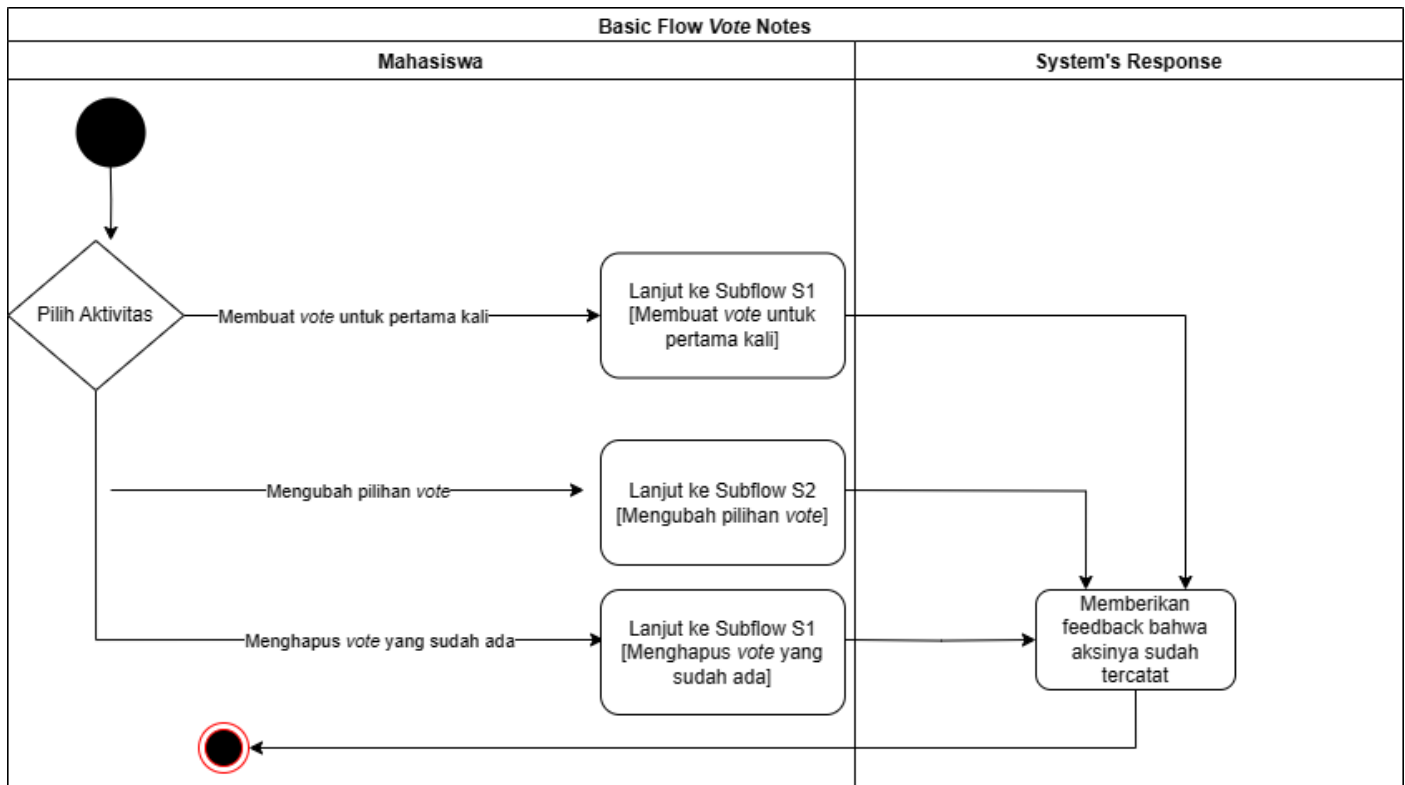
8. Activity Diagram for UCS 8: *Read User Reports (Muflih)*



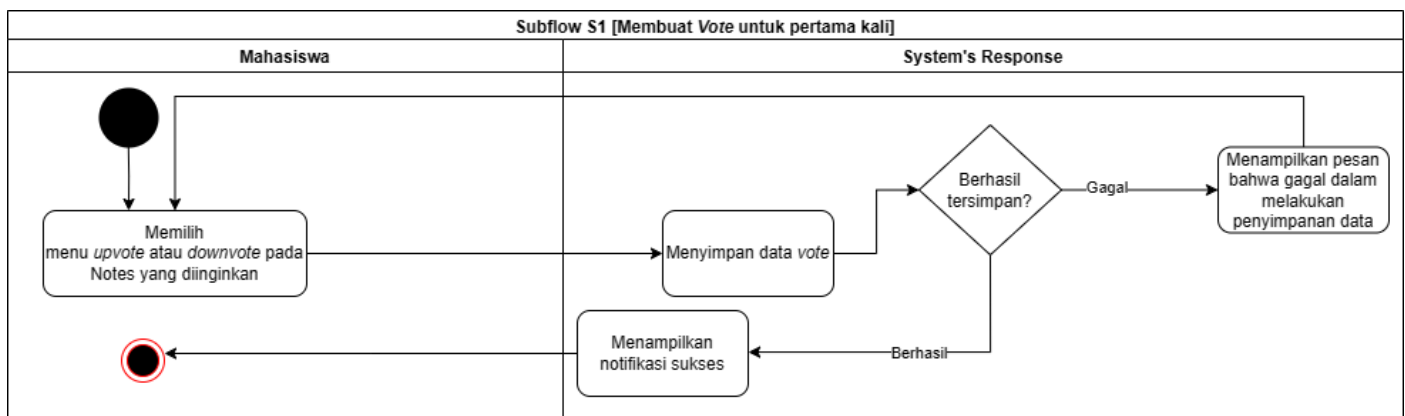
9. Activity Diagram for UCS 9: *Remove Notes (Muflih)*



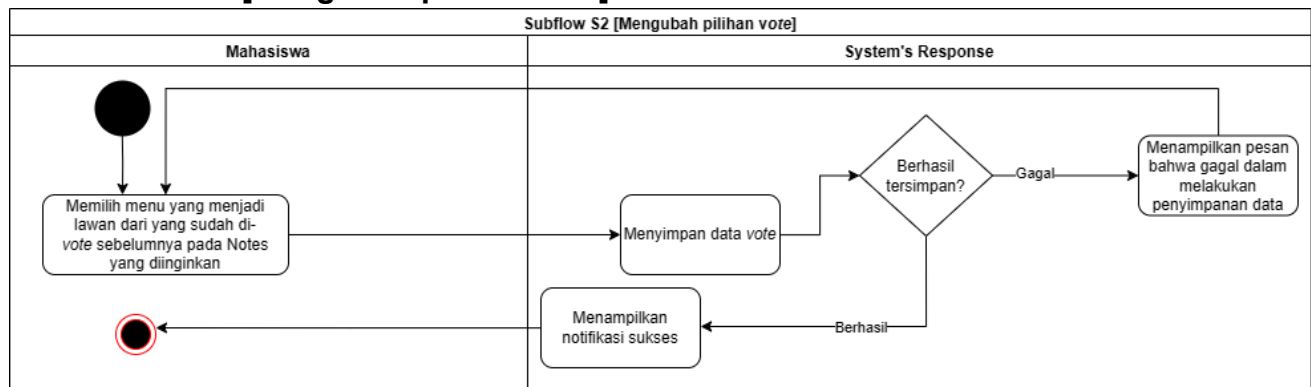
10. Activity Diagram for UCS 10: Vote Notes (Nando)



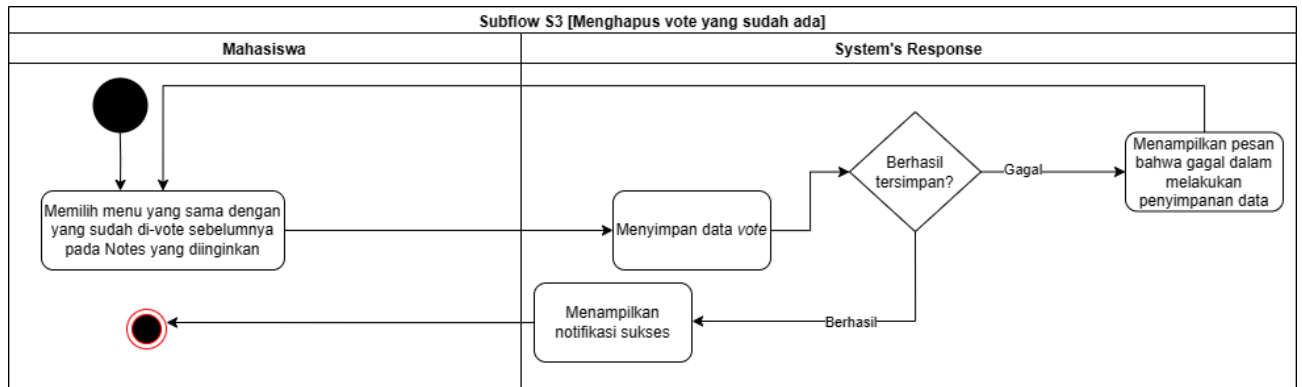
I. AD Subflow S1 [Membuat Vote untuk pertama kali]



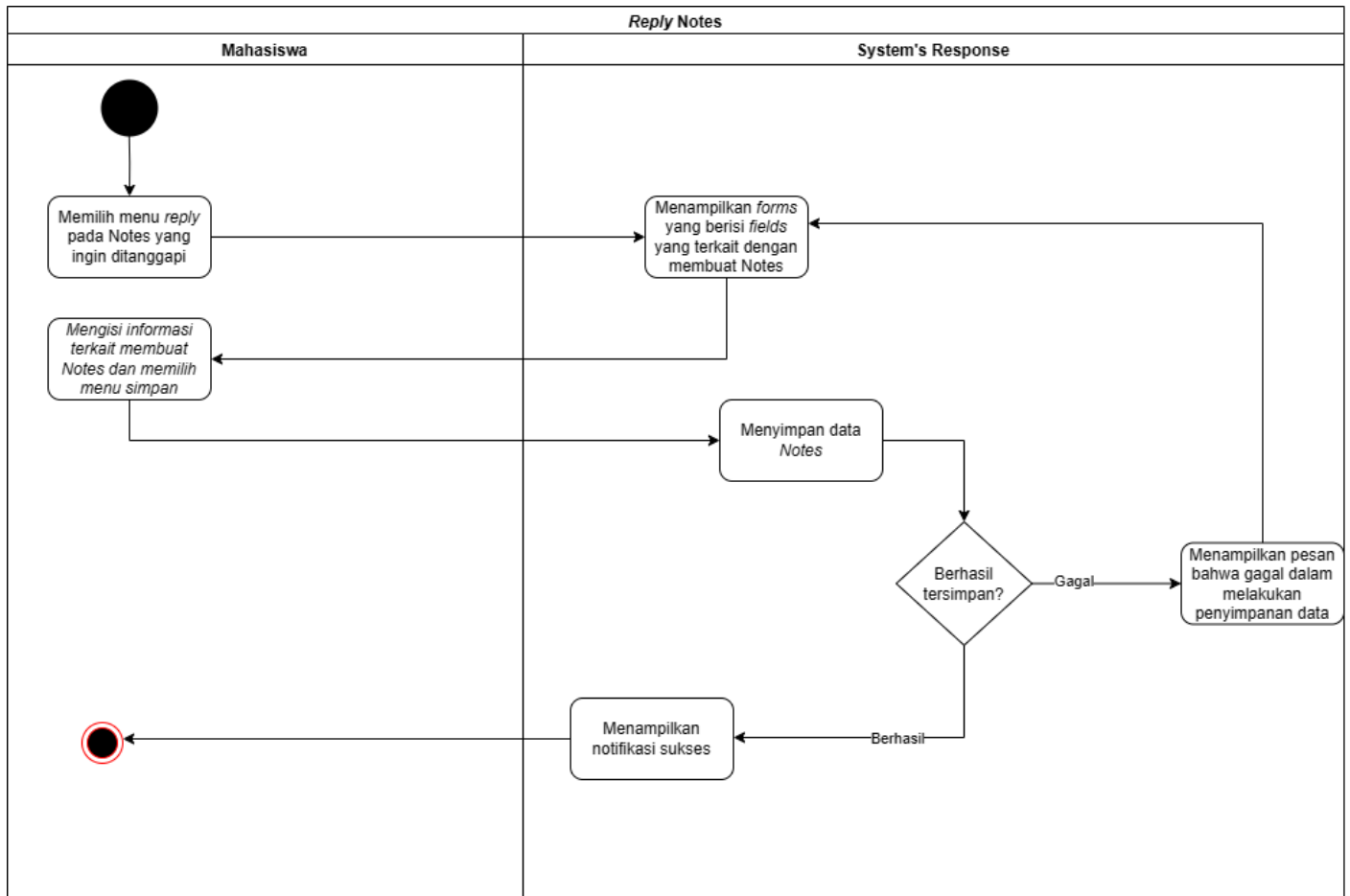
II. AD Subflow S2 [Mengubah pilihan vote]



III. AD Subflow S3 [Menghapus vote yang sudah ada]



11. Activity Diagram for UCS 11: *Reply Notes (Nando)*



D. [TK 3: 25 /100] Analysis Class Diagram

Letakkan Analysis Class Diagram disini

IV. System Design

A. [TK 4: 10/100] Design Class Diagram

Letakkan Design Class Diagram (Tanpa Method) disini

B. [TK 4: 15/100] Tabel Class Diagram

No.	Nama Method	Deskripsi	Berada di kelas mana?	Alasan (<i>Cohesion, Coupling, Connascence, or other Design Concept</i>)	Use Case Terkait
1.					
2.					
	...[add rows as needed]				

C. [TK 4: 75/100] Sequence Diagrams

1. Sequence Diagram [Nama Use Case 1]

Masukan Sequence Diagram dari UCS disini. Lanjutkan sesuai jumlah use case kelompok anda. Setiap anggota kelompok membuat Activity Diagram sesuai dengan UCS Masing Masing

2. Sequence Diagram [Nama Use Case 2]