

BAB IV
HASIL DAN PEMBAHASAN

4.1 Tahap Requirement

Tahapan pertama dalam *waterfall* berfokus pada perencanaan, pengumpulan dan pemahaman kebutuhan aplikasi. Aktivitas yang dilakukan dalam tahap *requirement* adalah merencanakan perancangan pembangunan aplikasi, mengumpulkan data yang dibutuhkan untuk membangun aplikasi, dan menganalisis kebutuhan aplikasi yang akan dibangun.

4.1.1 Perencanaan Pelaksanaan

Metode yang diimplementasikan dalam rancang bangun adalah metode *waterfall*. Pada tahap ini, pelaksanaan rancangan bangun berdasarkan metode pengembangan sistem dijadwalkan dan digambarkan dalam **Tabel 4.1**.

Tabel 4.1 Gantt Chart Pelaksanaan

Kegiatan	Bulan 1				Bulan 2				Bulan 3				Bulan 4			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Tahap Requirement																
Perencanaan																
Pelaksanaan																
Pengumpulan Data																
Analisis Sistem Berjalan																
Analisis Kebutuhan																
Tahap Design																
Perancangan UML																

Pertanyaan	Jawaban
Apa saja masalah yang Anda hadapi dalam proses presensi menggunakan sistem saat ini?	Proses presensi saat ini memakan waktu karena dosen harus memanggil satu per satu mahasiswa untuk mencatat kehadiran di sistem. Hal ini sering menyebabkan keterlambatan dalam memulai kelas dan mengurangi waktu efektif belajar.
Apakah sistem presensi manual saat ini memungkinkan terjadinya kecurangan?	Ya, meskipun menggunakan sistem, kecurangan seperti titip absen masih bisa terjadi karena dosen mencatat kehadiran berdasarkan panggilan nama, bukan bukti fisik kehadiran yang lebih valid.
Bagaimana Anda memantau kehadiran mahasiswa saat ini?	Pemantauan dilakukan dengan login ke sistem presensi, kemudian mencatat kehadiran mahasiswa satu per satu selama kelas berlangsung. Data ini kemudian diolah untuk menghasilkan laporan kehadiran.
Apakah Anda mengalami kesulitan dalam mengelola data presensi mahasiswa?	Ya, pengelolaan data presensi memerlukan waktu dan terkadang terjadi kesalahan input yang harus diperbaiki. Proses rekapitulasi data juga tidak efisien dan memakan waktu, terutama jika kelas berukuran besar.
Fitur apa yang Anda butuhkan	Kami membutuhkan fitur yang

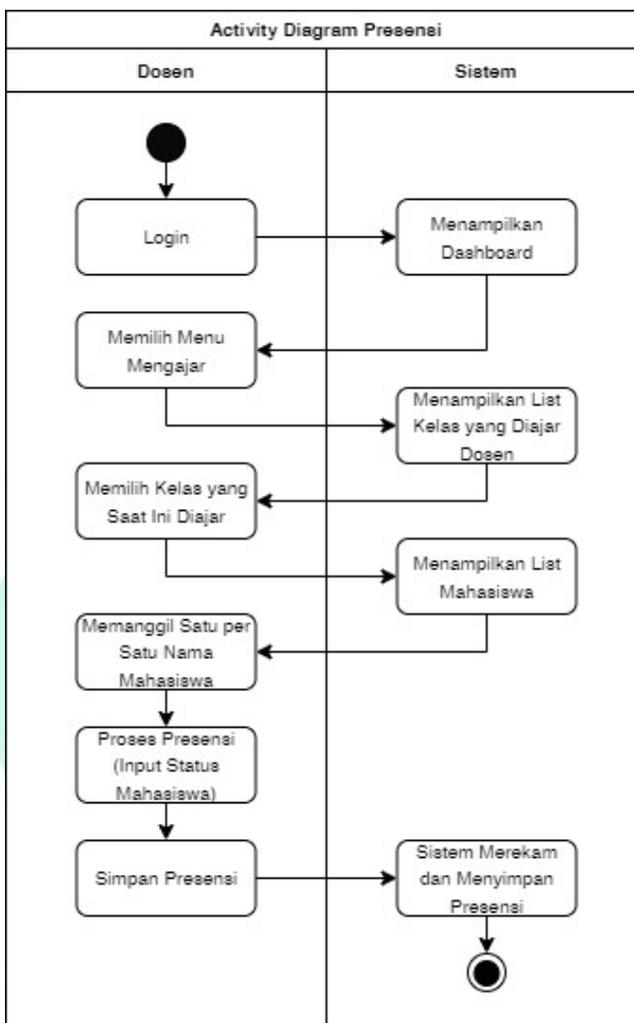
dalam sistem presensi baru untuk meningkatkan efisiensi?	memungkinkan presensi dilakukan secara otomatis, seperti menggunakan QR Code. Sistem juga harus menyediakan laporan kehadiran secara real-time dan terintegrasi dengan sistem informasi akademik lainnya.
Apakah ada kebutuhan khusus terkait keamanan data dalam sistem presensi yang baru?	Ya, keamanan data sangat penting. Kami membutuhkan sistem yang dapat mengenkripsi data presensi dan mengimplementasikan autentikasi dua faktor untuk memastikan data kehadiran hanya diakses oleh pihak yang berwenang.
Bagaimana Anda ingin melihat data presensi mahasiswa di sistem baru?	Kami ingin dapat melihat data presensi secara real-time melalui dashboard yang user-friendly. Fitur pencarian dan filter untuk melihat presensi berdasarkan kelas, tanggal, atau nama mahasiswa juga sangat diperlukan.
Apakah Anda memerlukan notifikasi otomatis terkait kehadiran mahasiswa?	Ya, notifikasi otomatis yang memberi tahu mahasiswa dan dosen tentang status kehadiran dan perubahan dalam jadwal kelas akan sangat membantu dalam meningkatkan keterlibatan dan disiplin.
Apakah ada fitur lain yang menurut Anda penting untuk dimasukkan	Selain fitur-fitur yang disebutkan, kami juga menginginkan fitur analitik untuk

dalam sistem presensi baru?

memantau tren kehadiran mahasiswa dan menghasilkan laporan yang dapat digunakan untuk pengambilan keputusan akademik.

4.1.3 Analisis Sistem Berjalan

Menganalisis sistem presensi yang diterapkan pada Universitas Nasional, mencakup sistem presensi kelas mahasiswa yang saat ini berjalan pada Universitas Nasional. Berdasarkan observasi yang telah dilakukan, alur sistem presensi kelas mahasiswa saat ini dilakukan oleh dosen atau pengajar dengan memanggil satu per satu mahasiswa dalam kelas dan melakukan proses presensi dalam aplikasi apps.unas.ac.id. **Gambar 4.1.** menunjukkan visualisasi dari proses menggunakan *Activity Diagram*.



Gambar 4.1 Visualisasi Sistem Berjalan

Berdasarkan alur tersebut, proses presensi pada apps.unas.ac.id menggunakan cara konvensional dengan dosen atau pengajar melakukan *login* ke apps.unas.ac.id lalu memanggil satu per satu mahasiswa dan mengubah status presensi mahasiswa dalam aplikasi. Hal ini menyebabkan lambatnya proses presensi mahasiswa sehingga efisiensi dapat ditingkatkan dalam proses presensi dengan mengimplementasikan teknologi *QR code*. Dalam rangka meningkatkan efisiensi dan akurasi proses presensi, serta memanfaatkan kemajuan teknologi informasi, maka penting untuk merancang dan membangun sebuah aplikasi presensi mahasiswa menggunakan teknologi *QR code*.

4.1.4 Analisis Kebutuhan

Elisitasi digunakan untuk menganalisis yang dibutuhkan oleh aplikasi yang akan dibangun. Kebutuhan-kebutuhan tersebut didapat berdasarkan hasil analisis data hasil tanya jawab. Kepala Badan Pengembangan Teknologi dan Sistem Informasi pada Universitas Nasional merupakan narasumbernya. Sebagai bagian dari perancangan sistem presensi, elisitasi telah dibuat untuk menganalisis kebutuhan berdasarkan fungsionalitas dan non-fungsionalitasnya. Hasil final dari semua tahapan dalam elisitasi disebut *draft* akhir elisitasi. Hasil ini digunakan sebagai dasar dan pedoman untuk membangun aplikasi.

- o **Elisitasi Tahap 1**

Berisi analisis kebutuhan awal, didapatkan dari proses mengumpulkan data melalui observasi dan wawancara, digunakan sebagai dasar untuk membangun sistem. **Tabel 4.3** menunjukkan elisitasi yang dihasilkan.

Tabel 4.3 Tahap 1 Elisitasi

Functional	
	ANALISIS KEBUTUHAN
	Saya ingin sistem dapat :
1.	Mahasiswa dapat melakukan <i>login</i>
2.	Mahasiswa dapat melihat <i>list</i> mata kuliah yang diambil
3.	Mahasiswa dapat melihat riwayat presensi
4.	Mahasiswa dapat melakukan presensi dengan <i>Scan QR Code</i>
5.	Mahasiswa dapat meng- <i>edit</i> profil
6.	Mahasiswa dapat melihat <i>list</i> mahasiswa pada kelas tertentu
7.	Mahasiswa dapat melakukan <i>logout</i>
8.	Dosen dapat melakukan <i>login</i>

9.	Dosen dapat melihat <i>list</i> mata kuliah yang diajar
10.	Dosen dapat mengelola presensi mahasiswa
11.	Dosen dapat melakukan generate QR Code untuk presensi
12.	Dosen dapat meng- <i>edit</i> profil
13.	Dosen dapat membuka dan menutup presensi
14.	Dosen dapat melakukan <i>logout</i>
Non Functional	
ANALISIS KEBUTUHAN	
Saya ingin sistem dapat :	
1.	Sistem memiliki komponen yang dinamis
2.	Sistem memiliki <i>user interface</i> yang menarik
4.	Sistem bersifat <i>user friendly</i>

o **Elisitasi Tahap 2**

Hasil dari tahap 1 digunakan untuk membuat tahap 2 dari elisitasi, yaitu mengelompokkan hasil elisitasi tahap 1 dengan MDI.

M : *Mandatory* (Wajib)

D : *Desirable* (Diinginkan)

I : *Inessential* (Tidak Mutlak Diinginkan/Tidak Penting)

Tabel 4. 4 Tahap 2 Elisitasi

Functional				
	ANALISIS KEBUTUHAN	M	D	i
Saya ingin sistem dapat :				
1.	Mahasiswa dapat melakukan <i>login</i>	√		

2.	Mahasiswa dapat melihat <i>list</i> mata kuliah yang diambil	√		
3.	Mahasiswa dapat melihat riwayat presensi	√		
4.	Mahasiswa dapat melakukan presensi dengan <i>Scan QR Code</i>	√		
5.	Mahasiswa dapat meng- <i>edit</i> profil		√	
6.	Mahasiswa dapat melihat <i>list</i> mahasiswa pada kelas tertentu		√	
7.	Mahasiswa dapat melakukan <i>logout</i>	√		
8.	Dosen dapat melakukan <i>login</i>	√		
9.	Dosen dapat melihat <i>list</i> mata kuliah yang diajar	√		
10.	Dosen dapat mengelola presensi mahasiswa	√		
11.	Dosen dapat melakukan generate <i>QR Code</i> untuk presensi	√		
12.	Dosen dapat meng- <i>edit</i> profil		√	
13.	Dosen dapat membuka dan menutup presensi	√		
14.	Dosen dapat melakukan <i>logout</i>	√		
Non Functional				
ANALISIS KEBUTUHAN				
Saya ingin sistem dapat :				
1.	Sistem memiliki komponen yang dinamis	√		
2.	Sistem memiliki <i>user interface</i> yang menarik	√		
4.	Sistem bersifat <i>user friendly</i>	√		

o **Elisitasi Tahap 3**

Ini menghasilkan penyederhanaan terhadap hasil elisitasi tahap 2 dengan mengurangi kebutuhan yang memiliki opsi I di metode MDI. Lalu, metode TOE digunakan pada elisitasi tahap 3 untuk menyederhanakan kembali semua kebutuhan yang tersisa.

T (*Technical*) : Tingkat kesulitan pengembangan sistem.

O (*Operational*) : Tingkat kesulitan penggunaan sistem.

E (*Economical*) : Tingkat biaya pengembangan sistem.

H (*High*) : Sulit untuk dikerjakan.

M (*Middle*) : Mampu untuk dikerjakan.

L (*Low*) : Mudah untuk dikerjakan.

Tabel 4. 5 Tahap 3 Elisitasi

Functional										
ANALISIS KEBUTUHAN		T			O			E		
		H	M	L	H	M	L	H	M	L
Saya ingin sistem dapat :										
1.	Mahasiswa dapat melakukan <i>login</i>			√			√			√
2.	Mahasiswa dapat melihat <i>list</i> mata kuliah yang diambil			√			√			√
3.	Mahasiswa dapat melihat riwayat presensi			√			√			√
4.	Mahasiswa dapat melakukan presensi dengan <i>Scan QR Code</i>			√			√			√
5.	Mahasiswa dapat meng- <i>edit</i> profil			√			√			√
6.	Mahasiswa dapat melihat list			√			√			√

	mahasiswa pada kelas tertentu								
7.	Mahasiswa dapat melakukan <i>logout</i>			√			√		√
8.	Dosen dapat melakukan <i>login</i>			√			√		√
9.	Dosen dapat melihat <i>list</i> mata kuliah yang diajar			√			√		√
10.	Dosen dapat mengelola presensi mahasiswa			√			√		√
11.	Dosen dapat melakukan <i>generate</i> QR Code untuk presensi			√			√		√
12.	Dosen dapat meng- <i>edit</i> profil			√			√		√
13.	Dosen dapat membuka dan menutup presensi			√			√		√
14.	Dosen dapat melakukan <i>logout</i>			√			√		√
Non Functional									
ANALISIS KEBUTUHAN									
Saya ingin sistem dapat :									
1.	Sistem memiliki komponen yang dinamis			√			√		√
2.	Sistem memiliki <i>user interface</i> yang menarik			√			√		√
4.	Sistem bersifat <i>user friendly</i>			√			√		√

○ **Elisitasi Final**

Hasil final dari semua tahapan dalam elisitasi disebut *draft* akhir elisitasi. Hasil ini digunakan sebagai dasar dan pedoman untuk membangun sistem dalam penelitian. Dari elisitasi tahap 3, dihasilkan elisitasi final sistem yang nantinya dijadikan dasar dalam rancang bangun aplikasi presensi kelas mahasiswa.

Tabel 4.6. menunjukkan elisitasi final.

Tabel 4. 6 Elisitasi Final

Functional	
	ANALISIS KEBUTUHAN
	Saya ingin sistem dapat :
1.	Mahasiswa dapat melakukan <i>login</i>
2.	Mahasiswa dapat melihat <i>list</i> mata kuliah yang diambil
3.	Mahasiswa dapat melihat riwayat presensi
4.	Mahasiswa dapat melakukan presensi dengan <i>Scan QR Code</i>
5.	Mahasiswa dapat meng- <i>edit</i> profil
6.	Mahasiswa dapat melihat <i>list</i> mahasiswa pada kelas tertentu
7.	Mahasiswa dapat melakukan <i>logout</i>
8.	Dosen dapat melakukan <i>login</i>
9.	Dosen dapat melihat <i>list</i> mata kuliah yang diajar
10.	Dosen dapat mengelola presensi mahasiswa
11.	Dosen dapat melakukan <i>generate QR Code</i> untuk presensi
12.	Dosen dapat meng- <i>edit</i> profil
13.	Dosen dapat membuka dan menutup presensi
14.	Dosen dapat melakukan <i>logout</i>
Non Functional	

	ANALISIS KEBUTUHAN
	Saya ingin sistem dapat :
1.	Sistem memiliki komponen yang dinamis
2.	Sistem memiliki <i>user interface</i> yang menarik
4.	Sistem bersifat <i>user friendly</i>

4.2 Tahap *Design*

Tahap *Design* dilakukan berdasarkan hasil analisis masalah pada sistem berjalan serta analisis terhadap apa yang dibutuhkan oleh aplikasi yang telah dijelaskan. Di dalam tahapan ini, analisis perancangan sistem dilakukan dengan OOAD. Hasil dari analisis di dalam tahapan ini berupa pemodelan UML (*Unified Modelling Language*), yang mencakup spesifikasi *database* serta *use case*, aktivitas, *sequence*, dan diagram *class*.

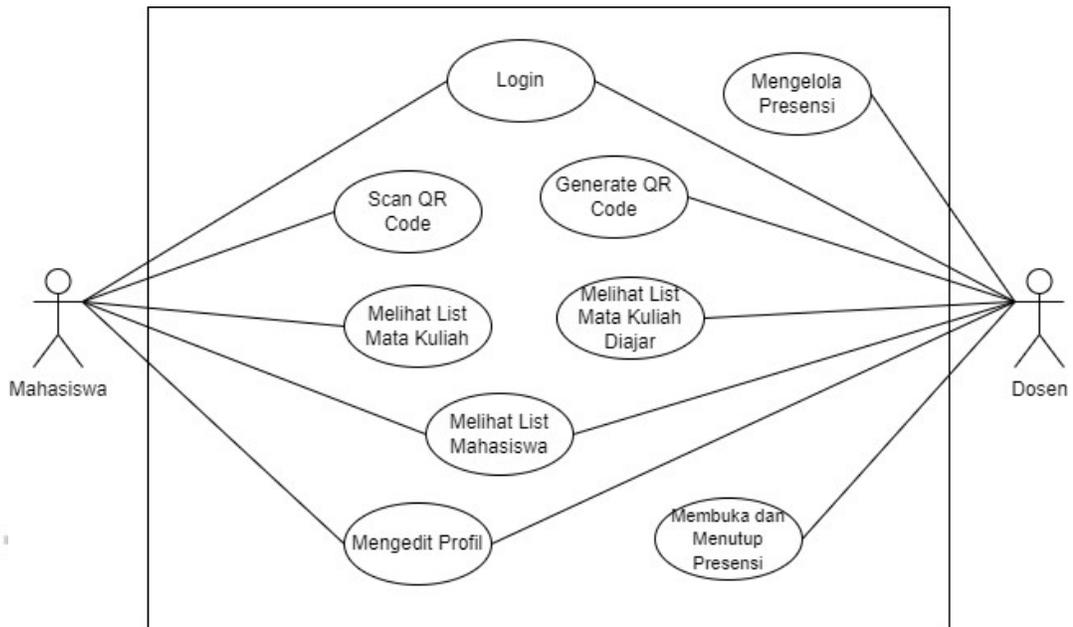
4.2.1 *Use Case Diagram*

Pemodelan ini dibentuk sesuai hasil analisis kebutuhan sistem (Elisitasi). Berdasarkan elisitasi tersebut, dapat disimpulkan proses yang dijalankan oleh setiap aktor di dalam sistem sebagai berikut:

- Aktor 1 (Mahasiswa)
 - a. *Login*
 - b. *Scan QR Code*
 - c. Melihat List Mata Kuliah
 - d. Melihat List Mahasiswa
 - e. *Meng-edit* Profil
- Aktor 2 (Dosen)
 - a. *Login*
 - b. *Generate QR Code*
 - c. Melihat List Mata Kuliah Diajar
 - d. Melihat List Mahasiswa
 - e. *Meng-edit* Profil

- f. Mengelola Presensi
- g. Membuka dan Menutup Presensi

Proses-proses yang dijalankan oleh setiap aktor tersebut divisualisasikan **Gambar 4.2**.



Gambar 4.2 Use Case Diagram

4.2.2 Use Case Description

Berdefinisi deskripsi dari proses yang telah tergambar di dalam *Use Case Diagram*. Deskripsi-deskripsi tersebut digambarkan dalam tabel-tabel berikut:

Tabel 4.7 Use Case Login

Use Case Name	Login
Actor	Mahasiswa dan Dosen
Description	Pada aktivitas ini menggambarkan proses <i>login user</i>
Trigger	<i>User</i> membuka aplikasi
Pre-Condition	Memiliki <i>username</i> dan <i>password</i>

Normal Course	User Action	System
		Sistem menampilkan <i>form login</i>
	User mengisi <i>form login</i>	
	User mengklik tombol "Sign In"	
		Sistem menampilkan <i>dashboard</i> utama
Post-Condition	User dapat mengakses <i>dashboard</i> utama	
Alternative Flows	Menampilkan "Username atau Password Salah."	

Tabel 4.8 Use Case Melihat List Mata Kuliah

Use Case Name	Melihat List Mata Kuliah	
Actor	Mahasiswa	
Description	Pada aktivitas ini menggambarkan proses melihat list mata kuliah mahasiswa	
Trigger	User mengakses menu "Mata Kuliah"	
Pre-Condition	Sudah <i>login</i>	
Normal Course	User Action	System
	User berhasil <i>login</i>	
		Sistem menampilkan halaman <i>dashboard</i>
	User memilih menu "Mata Kuliah"	
		Sistem menampilkan list mata kuliah

Post-Condition	Sistem menampilkan list mata kuliah.
Alternative Flows	Menampilkan “Maaf, terdapat kesalahan dalam server”

Tabel 4.9 Use Case Scan QR Code

Use Case Name	Scan QR Code	
Actor	Mahasiswa	
Description	Pada aktivitas ini mahasiswa melakukan <i>scan</i> QR untuk melakukan presensi	
Trigger	Mahasiswa menekan <i>button</i> “Scan QR Code”	
Pre-Condition	Membuka halaman <i>dashboard</i> mahasiswa	
Normal Course	User Action	System
	User menekan button “Scan QR Code”	
		Sistem membuka <i>camera scanner</i>
	User melakukan <i>scan</i> QR	
		Sistem merekam dan menyimpan presensi
Post-Condition	Sistem menampilkan pemberitahuan bahwa presensi berhasil disimpan.	
Alternative Flows	Menampilkan "Maaf presensi gagal disimpan, silahkan coba lagi."	

Tabel 4. 10 Use Case Melihat List Mahasiswa

Use Case Name	Melihat List Mahasiswa	
Actor	Mahasiswa dan Dosen	
Description	Pada aktivitas ini menggambarkan <i>user</i> melihat list mahasiswa pada kelas tertentu	
Trigger	<i>User</i> menekan <i>button</i> “Lihat Mahasiswa”	
Pre-Condition	Membuka halaman <i>dashboard user</i>	
Normal Course	User Action	System
	Mahasiswa memilih menu “Mata Kuliah” atau Dosen memilih menu “Mata Kuliah Diajar”	
		Menampilkan halaman Mata Kuliah atau Mata Kuliah Diajar
	<i>User</i> menekan <i>button</i> “Lihat Mahasiswa” pada mata kuliah tertentu	
		Menampilkan list mahasiswa pada kelas tersebut
Post-Condition	Sistem menampilkan list mahasiswa pada kelas tersebut	
Alternative Flows	Menampilkan “Maaf, terdapat kesalahan dalam server”	

Tabel 4. 11 Use Case Mengedit Profil

Use Case Name	Mengedit Profil	
Actor	Mahasiswa dan Dosen	
Description	Pada aktivitas ini menggambarkan <i>user</i> mengedit profil	
Trigger	User menekan profil	
Pre-Condition	Membuka halaman <i>dashboard</i>	
Normal Course	User Action	System
	User menekan profil	
		Menampilkan profil <i>user</i>
	User mengubah data profil	
		Merekam dan menyimpan perubahan
Post-Condition	Sistem menyimpan perubahan pada profil <i>user</i>	
Alternative Flows	Menampilkan “Maaf, terdapat kesalahan dalam server”	

Tabel 4. 12 Use Case Mengelola Presensi

Use Case Name	Mengelola Presensi
Actor	Dosen
Description	Pada aktivitas ini dosen mengelola presensi mahasiswa

Trigger	Dosen menekan <i>button</i> “Kehadiran”	
Pre-Condition	Membuka halaman <i>dashboard</i> Dosen	
Normal Course	Action	System
	Dosen menekan <i>button</i> “Kehadiran” pada mata kuliah tertentu	
		Menampilkan halaman untuk mengelola presensi mahasiswa
	Dosen mengelola presensi mahasiswa	
		Merekam dan menyimpan perubahan
Post-Condition	Presensi Mahasiswa berhasil disimpan	
Alternative Flows	Menampilkan "Presensi gagal disimpan, silahkan coba lagi"	

Tabel 4. 13 Use Case Generate QR Code

Use Case Name	Generate QR Code	
Actor	Dosen	
Description	Pada aktivitas ini dosen me-generate QR code untuk presensi mahasiswa	
Trigger	Dosen menekan <i>button</i> “Generate QR Code”	
Pre-Condition	Membuka halaman <i>dashboard</i> Dosen	
Normal Course	User Action	System

	User menekan <i>button</i> “Generate QR Code”	
		Menampilkan QR <i>code</i> untuk presensi mahasiswa
Post-Condition	Sistem menampilkan QR <i>code</i>	
Alternative Flows	Menampilkan "QR Code gagal dibuat, silahkan coba lagi"	

Tabel 4. 14 Use Case Melihat Mata Kuliah Diajar

Use Case Name	Melihat Mata Kuliah Diajar	
Actor	Dosen	
Description	Pada aktivitas ini menggambarkan proses melihat mata kuliah yang diajar Dosen	
Trigger	Dosen memilih menu “Mata Kuliah Diajar”	
Pre-Condition	Sudah <i>login</i>	
Normal Course	User Action	System
	User memilih menu “Mata Kuliah Diajar”	Sistem menampilkan halaman yang berisi mata kuliah yang diajar Dosen
Post-Condition	Sistem menampilkan halaman Mata Kuliah Diajar	
Alternative Flows	Menampilkan “Maaf, terdapat kesalahan dalam server”	

Tabel 4. 15 Use Case Membuka Presensi

Use Case Name	Membuka Presensi	
Actor	Dosen	
Description	Pada aktivitas ini dosen membuka presensi agar mahasiswa dapat melakukan <i>scan</i> terhadap QR code yang telah di-generate	
Trigger	Dosen menekan <i>button</i> “Buka Presensi”	
Pre-Condition	Mengakses halaman <i>dashboard</i>	
Normal Course	Action	System
	Dosen menekan <i>button</i> “Buka Presensi”	Menyimpan status dan mengubah <i>button</i> menjadi “Tutup Presensi”
Post-Condition	Sistem menyimpan status presensi terbuka	
Alternative Flows	Menampilkan “Maaf, terdapat kesalahan dalam server”	

Tabel 4. 16 Use Case Menutup Presensi

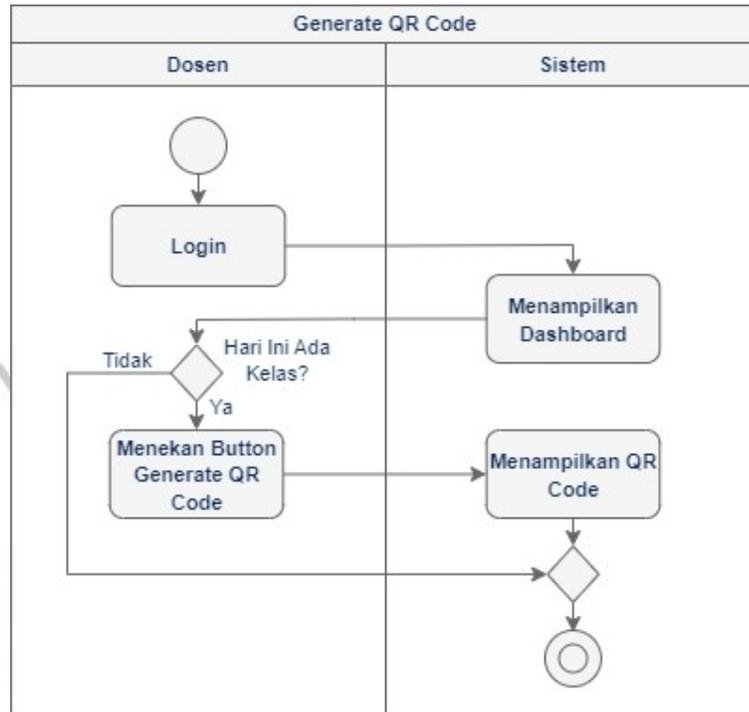
Use Case Name	Menutup Presensi
Actor	Dosen
Description	Pada aktivitas ini dosen menutup presensi sehingga mahasiswa tidak lagi dapat melakukan presensi
Trigger	Dosen menekan <i>button</i> “Tutup Presensi”
Pre-Condition	Dosen telah membuka presensi

	<i>Action</i>	<i>System</i>
<i>Normal Course</i>	Dosen menekan <i>button</i> “Tutup Presensi”	
		Menyimpan status dan mengubah <i>button</i> menjadi “Buka Presensi”
<i>Post-Condition</i>	Sistem menyimpan status presensi tertutup	
<i>Alternative Flows</i>	Menampilkan “Maaf, terdapat kesalahan dalam server”	

4.2.3 *Activity Diagram*

Berisi *flow* dari kegiatan pada setiap proses dijelaskan dalam *use case*. Aliran aktivitas tersebut digambarkan dalam *Activity Diagram*.

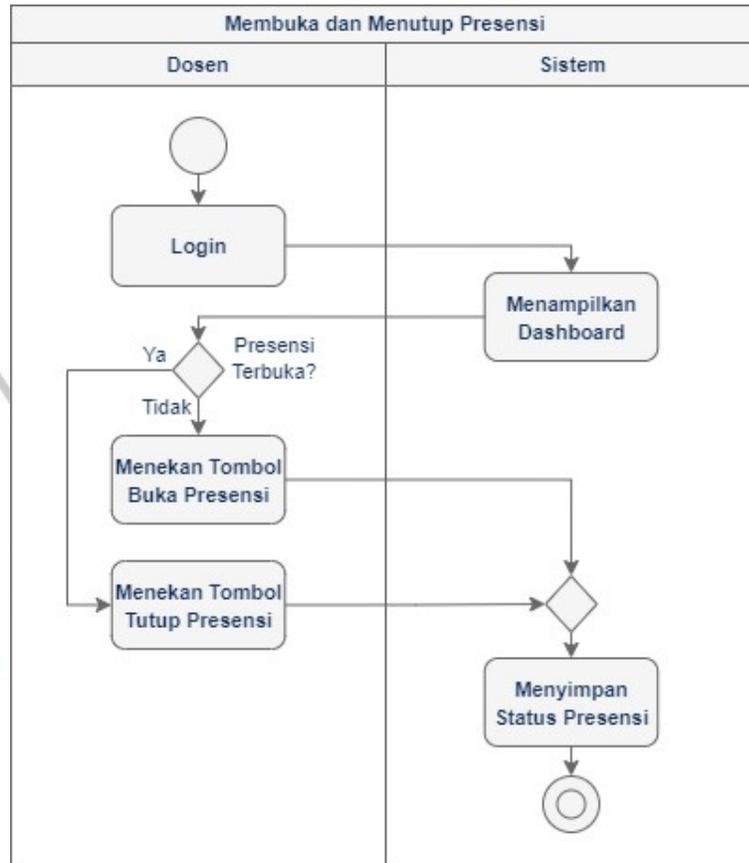
1. Generate QR Code



Gambar 4.3 Generate QR Code Activity Diagram

Gambar 4.3 diagram aktivitas *Generate QR Code*. Aktivitas dalam *Activity Generate QR Code* adalah dosen mengisi *form login* dan aplikasi langsung menampilkan halaman *dashboard* dosen. Lalu, jika hari ini ada kelas, kelas akan muncul di dalam *dashboard* dan dosen dapat menekan tombol “Generate QR Code” untuk menampilkan QR Code untuk presensi mahasiswa.

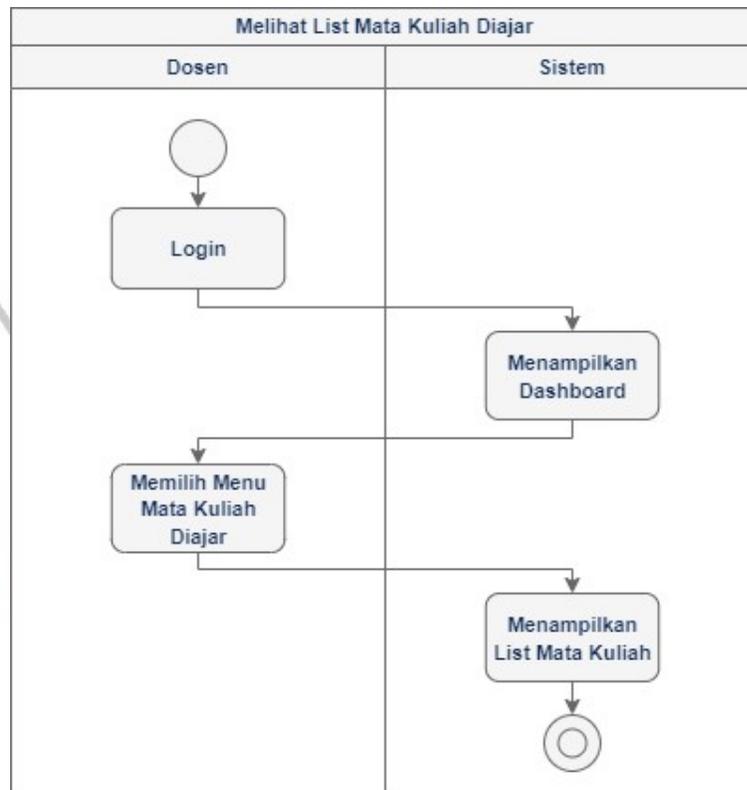
2. Membuka dan Menutup Presensi



Gambar 4.4 Activity Diagram Membuka dan Menutup Presensi

Gambar 4.4 Merupakan gambar Activity Diagram Membuka dan Menutup Presensi. Aktivitas di dalamnya dilakukan oleh dosen dengan melakukan *login* dan sistem akan menampilkan *dashboard* dosen. Jika status presensi belum terbuka, dosen dapat menekan tombol “Buka Presensi” dan sistem akan mengubah status presensi menjadi terbuka. Tulisan di tombol berubah menjadi “Tutup Presensi” dan jika dosen menekan tombol tersebut, status presensi akan tertutup. Status presensi disimpan oleh sistem.

3. Melihat List Mata Kuliah Diajar



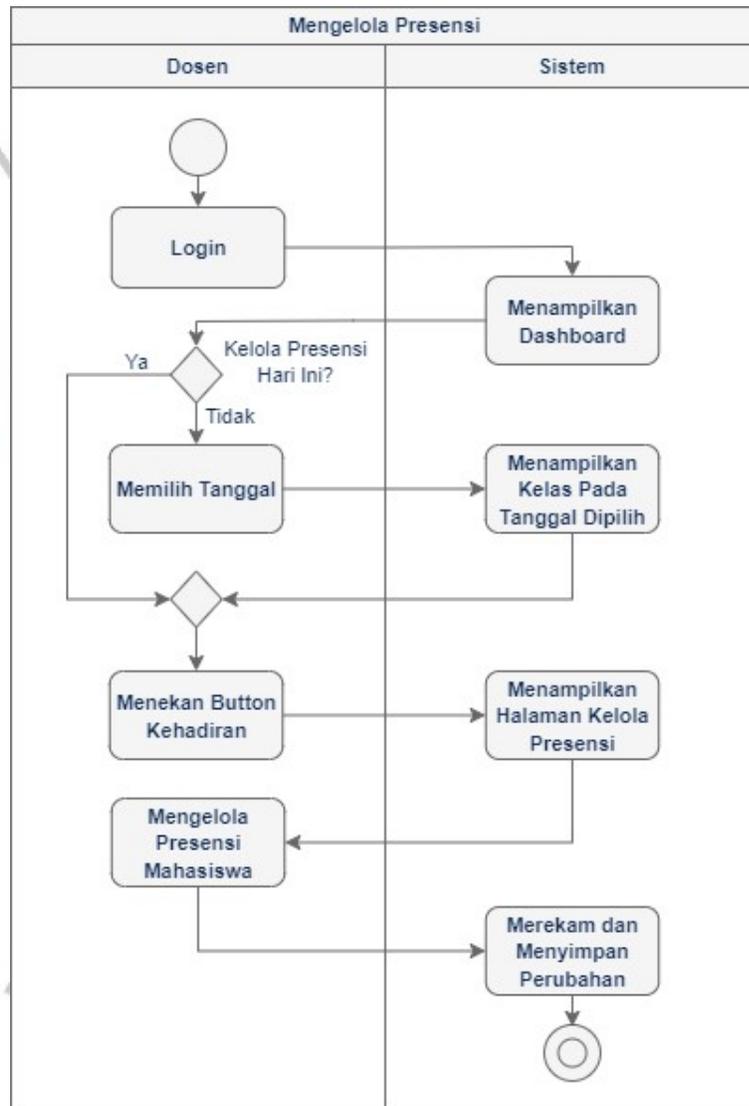
Gambar 4.5 Activity Diagram Melihat List Mata Kuliah Diajar

Gambar 4.5 gambar Activity Melihat List Mata Kuliah Diajar. Aktivitas dalam Activity Diagram tersebut dilakukan oleh dosen dengan melakukan *login* dan sistem akan menampilkan *dashboard* dosen. Setelah itu, dosen mengakses menu “Mata Kuliah Diajar” dan di dalam aplikasi akan tampil list mata kuliah yang diajar oleh dosen.

4. Mengelola Presensi

Aktivitas di dalam Activity Diagram Mengelola Presensi dilakukan oleh dosen dengan melakukan *login* dan system akan menampilkan *dashboard* dosen. Lalu, jika dosen ingin mengelola presensi hari ini, dosen dapat menekan tombol “Kehadiran” sehingga sistem akan menampilkan halaman

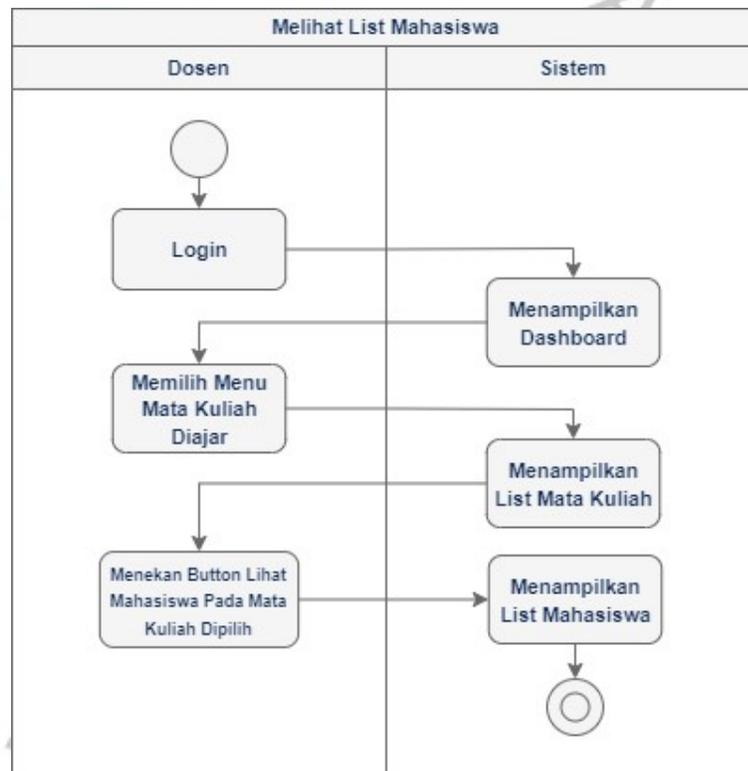
kelola presensi. Dosen dapat mengelola status presensi mahasiswa apabila ada kendala dalam penggunaan QR Code. Dosen juga dapat mengelola presensi mahasiswa pada kelas di tanggal tertentu. *Activity* Mengelola Presensi divisualisasikan **Gambar 4.6.**



Gambar 4. 6 Activity Diagram Mengelola Presensi

5. Melihat List Mahasiswa (Dosen)

Aktivitas dalam *Activity Diagram* Melihat List Mahasiswa dilakukan oleh dosen dengan melakukan *login* dan sistem akan menampilkan *dashboard* dosen. Lalu, dosen memilih menu “Mata Kuliah Diajar” dan setelah list mata kuliah tampil, dosen menekan tombol “Lihat Mahasiswa” pada mata kuliah yang diinginkan. *Activity* Melihat List Mahasiswa divisualisasikan **Gambar 4.7**.

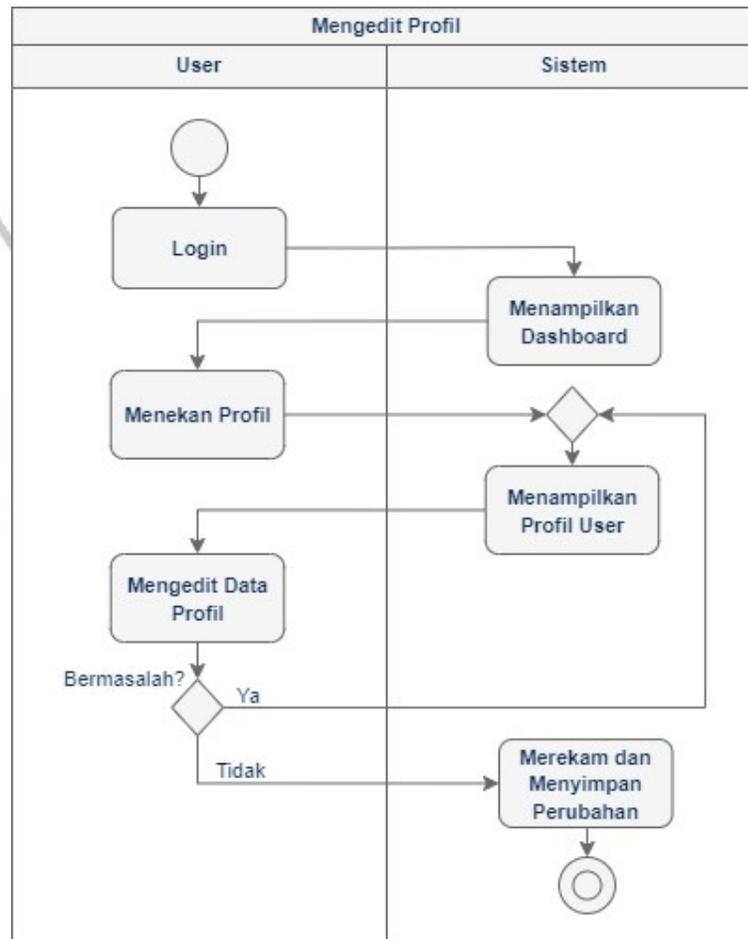


Gambar 4. 7 *Activity Diagram* Melihat List Mahasiswa (Dosen)

6. Mengedit Profil

Aktivitas dalam *Activity Diagram* Meng-edit Profil dilakukan oleh *user* (mahasiswa dan dosen). Setelah *user* mengisi *form login*, aplikasi menampilkan *dashboard* utama. Setelahnya, *user* menekan profil pada kanan atas halaman *dashboard* dan

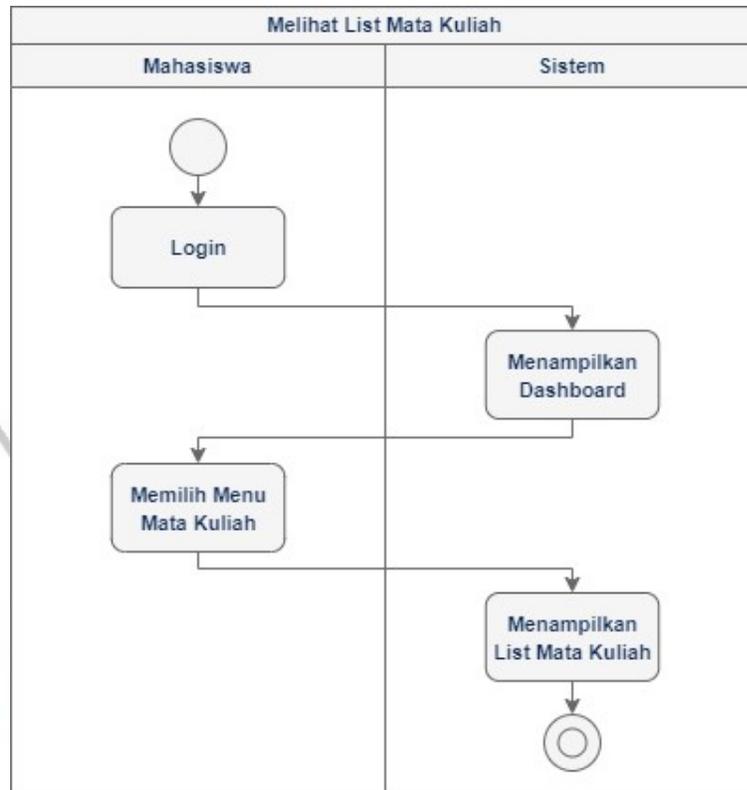
sistem akan menampilkan profil *user*. *User* dapat mengubah profil dan sistem akan menyimpan perubahan profil jika tidak ada masalah. *Activity* Meng-*edit* Profil divisualisasikan **Gambar 4.8**.



Gambar 4.8 Activity Diagram Mengedit Profil

7. Melihat List Mata Kuliah

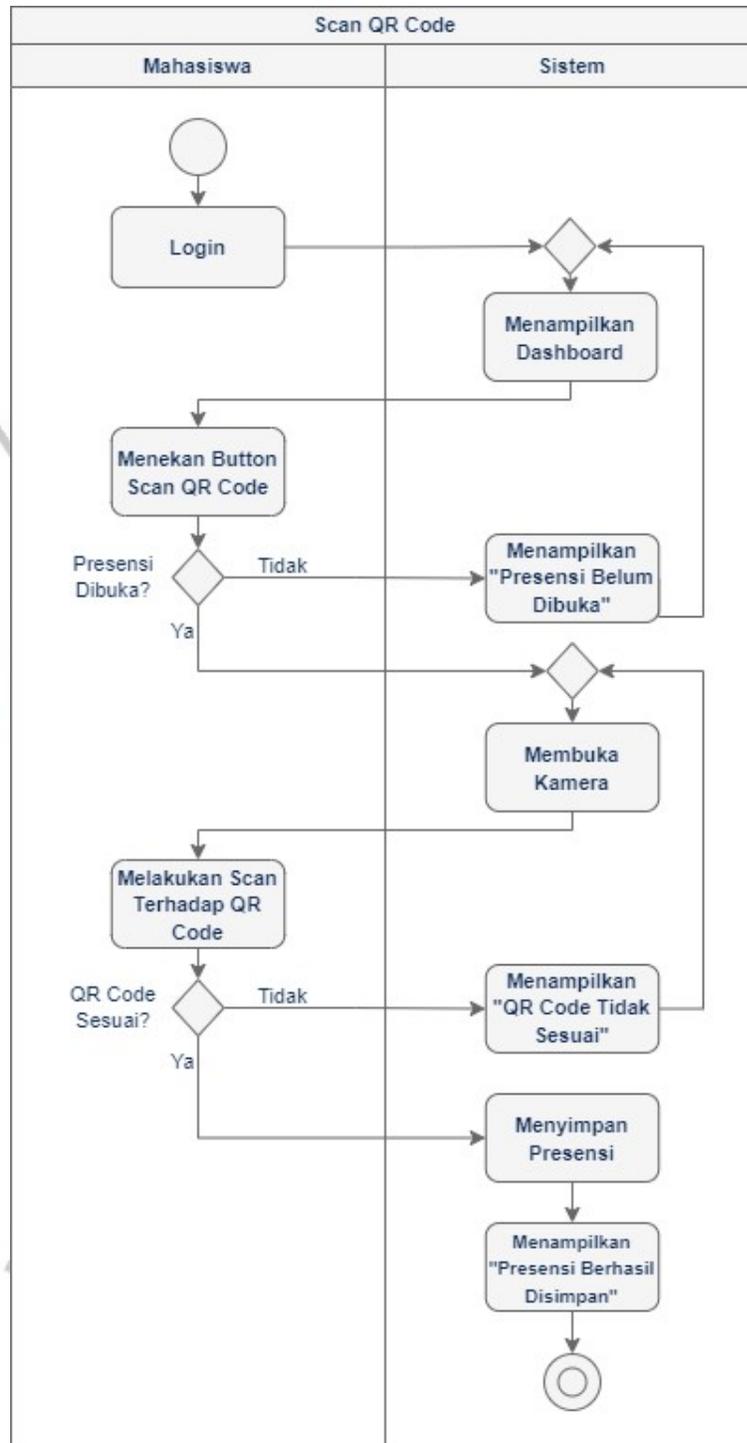
Aktivitas dalam *Activity Diagram* Melihat List Mata Kuliah dilakukan oleh mahasiswa dengan melakukan *login* dan aplikasi akan membuka *dashboard* mahasiswa. Setelahnya, mahasiswa dapat mengakses menu “Mata Kuliah” dan aplikasi akan memberikan tampilan halaman dengan berisi list mata kuliah mahasiswa. **Gambar 4.9** berisi visualisasi dari *Activity* Melihat List Mata Kuliah.



Gambar 4.9 Activity Diagram Melihat List Mata Kuliah

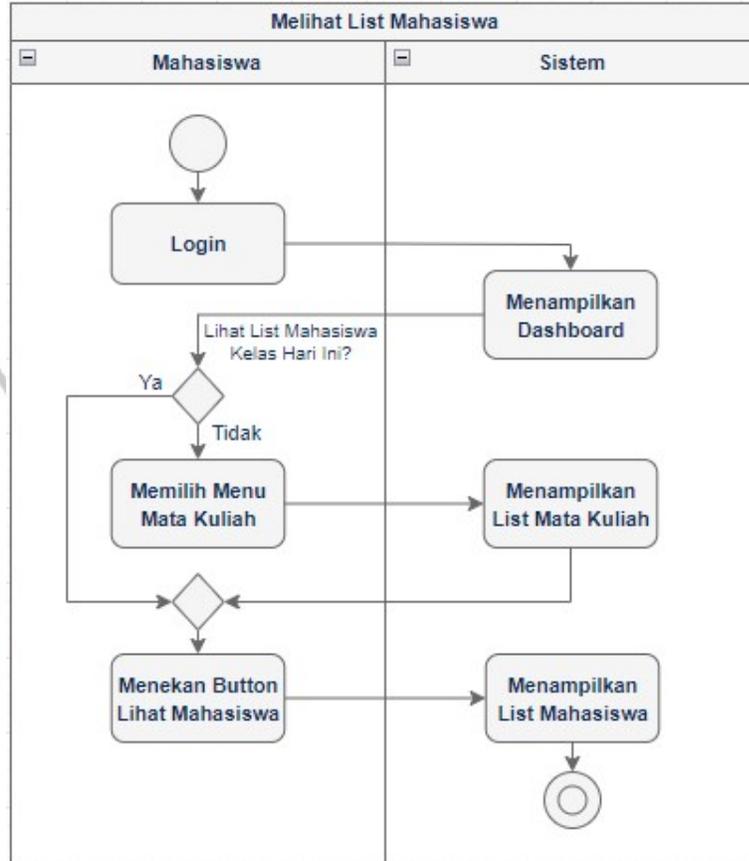
8. Scan QR Code

Aktivitas dalam *Activity Scan QR* dilakukan oleh mahasiswa dengan melakukan *login* dan sistem akan menampilkan *dashboard* mahasiswa. Jika hari ini mahasiswa ada kelas, kelas tersebut akan muncul di dalam *dashboard* dan mahasiswa dapat menekan tombol “Scan QR Code” untuk membuka kamera, jika dosen telah membuka presensi kelas. Setelah itu, mahasiswa dapat melakukan *scan QR code* yang diberikan oleh dosen untuk melakukan presensi. Sistem akan menyimpan presensi mahasiswa dan otomatis mengubah status presensi mahasiswa menjadi “Hadir” jika mahasiswa berhasil melakukan *scan QR code* pada kelas yang sedang berlangsung. *Activity Diagram Scan QR Code* divisualisasikan **Gambar 4.10**.



Gambar 4. 10 Activity Diagram Scan QR Code

9. Melihat List Mahasiswa

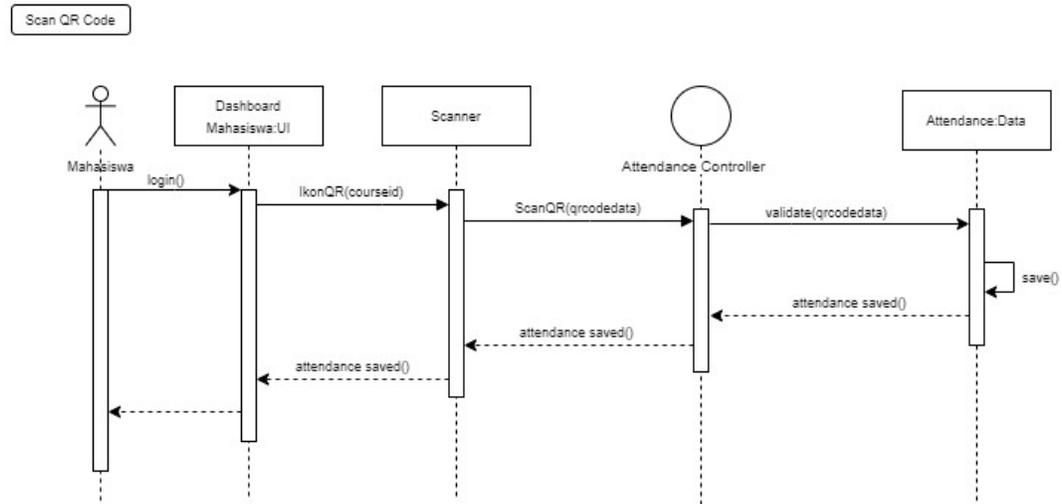


Gambar 4. 11 Activity Diagram Melihat List Mahasiswa (Mahasiswa)

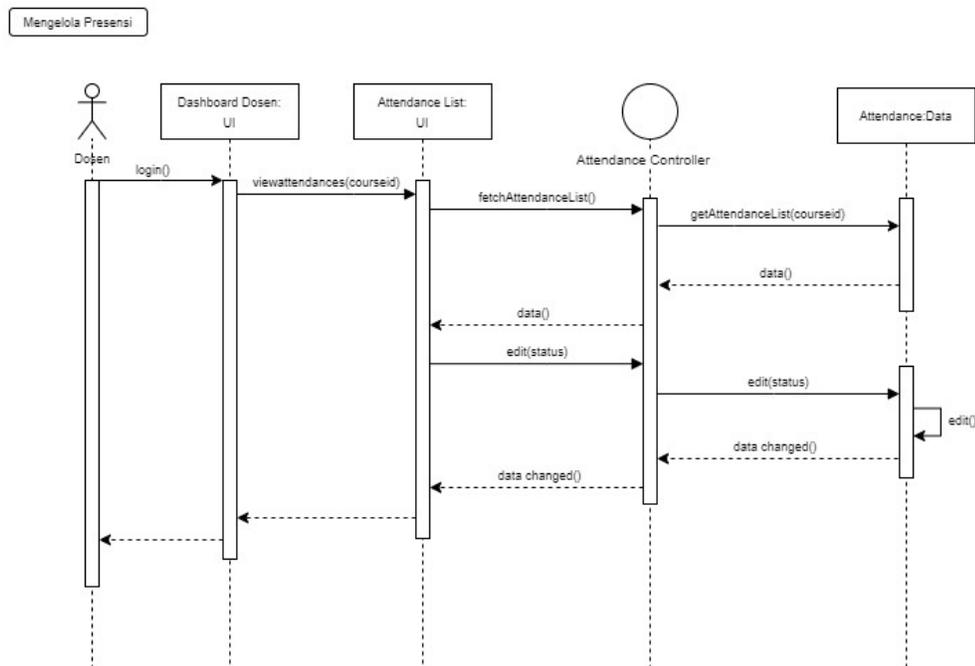
Gambar 4.11 berisi visualisasi dari *Activity* Melihat List Mahasiswa yang dilakukan oleh mahasiswa setelah mengisi *form login* dan aplikasi akan memberikan tampilan *dashboard* mahasiswa. Setelah itu, jika mahasiswa ingin melihat list mahasiswa pada kelas hari ini, mahasiswa dapat menekan tombol "Lihat Mahasiswa" sehingga sistem akan menampilkan list mahasiswa di dalam kelas yang dipilih. Jika mahasiswa ingin melihat list mahasiswa pada kelas lain, mahasiswa dapat memilih menu "Mata Kuliah" untuk melihat list mata kuliah dan menekan tombol "Lihat Mahasiswa" pada mata kuliah yang diinginkan.

4.2.4 Sequence Diagram

Berfungsi memberikan gambaran keterkaitan objek-objek pada setiap aktivitas pada aplikasi presensi dalam urutan waktu tertentu.

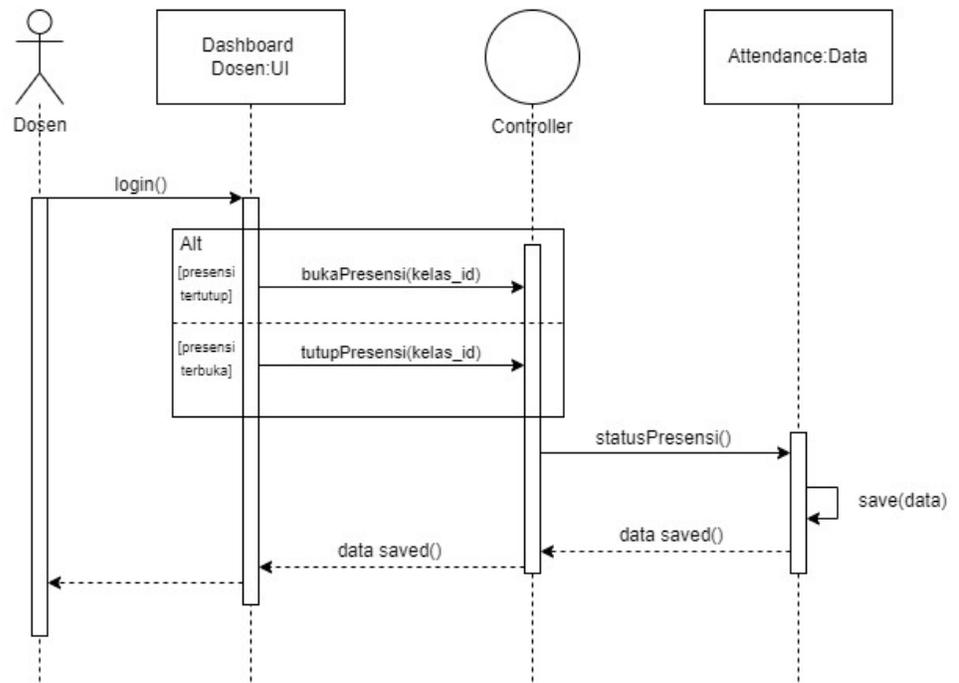


Gambar 4. 12 Sequence Diagram Scan QR Code



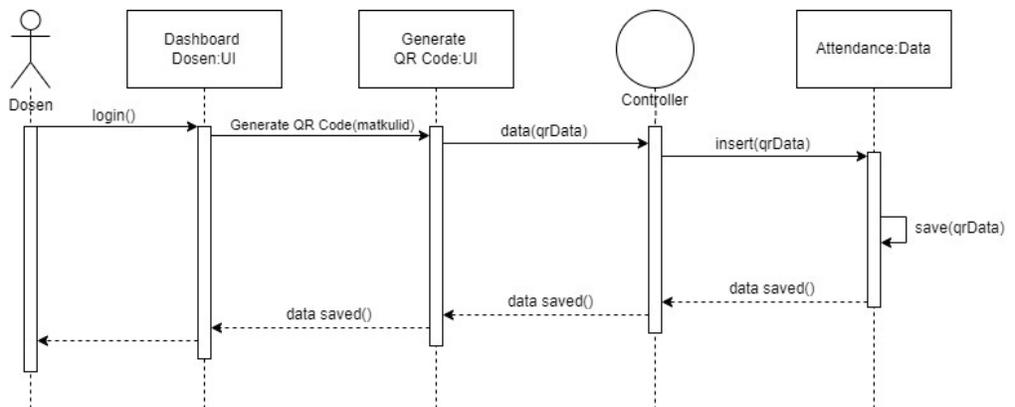
Gambar 4. 13 Sequence Diagram Mengelola Presensi

Membuka dan Menutup Presensi

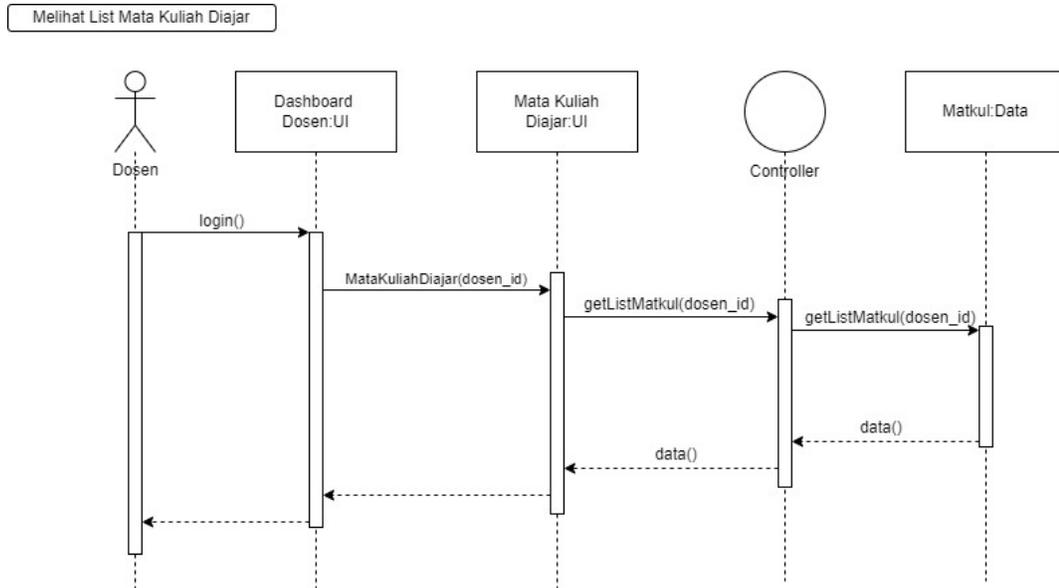


Gambar 4. 14 Sequence Diagram Membuka dan Menutup Presensi

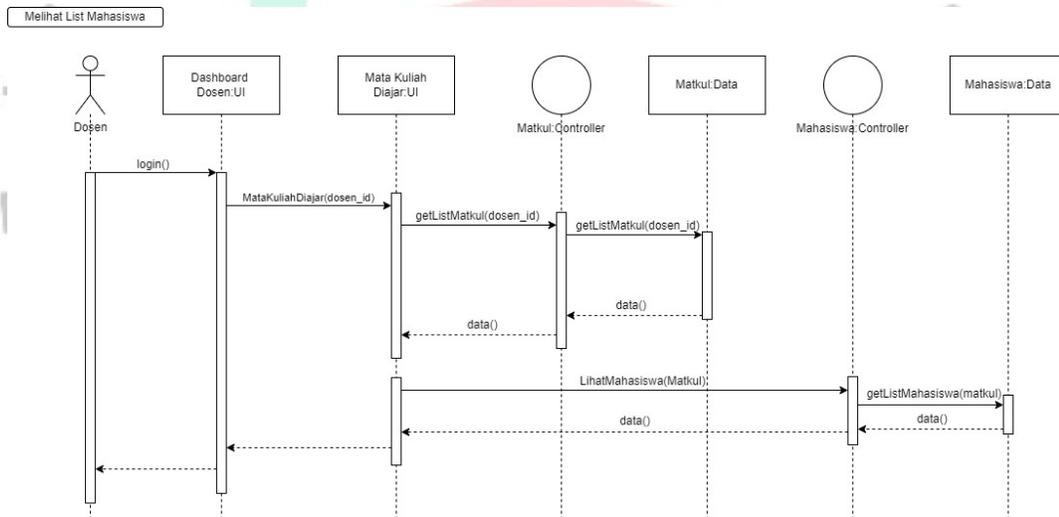
Generate QR Code



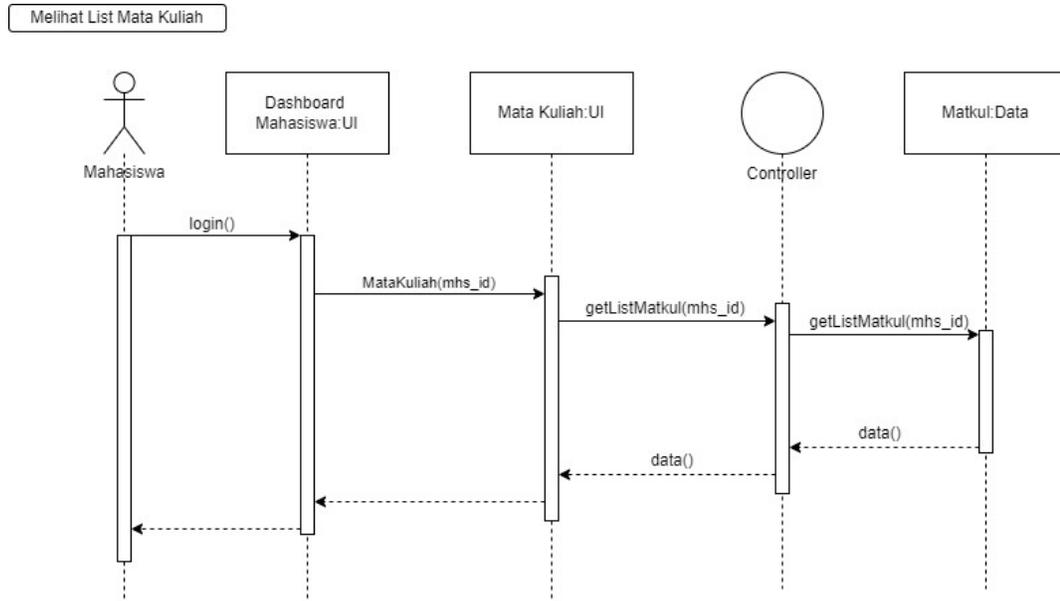
Gambar 4. 15 Sequence Diagram Generate QR Code



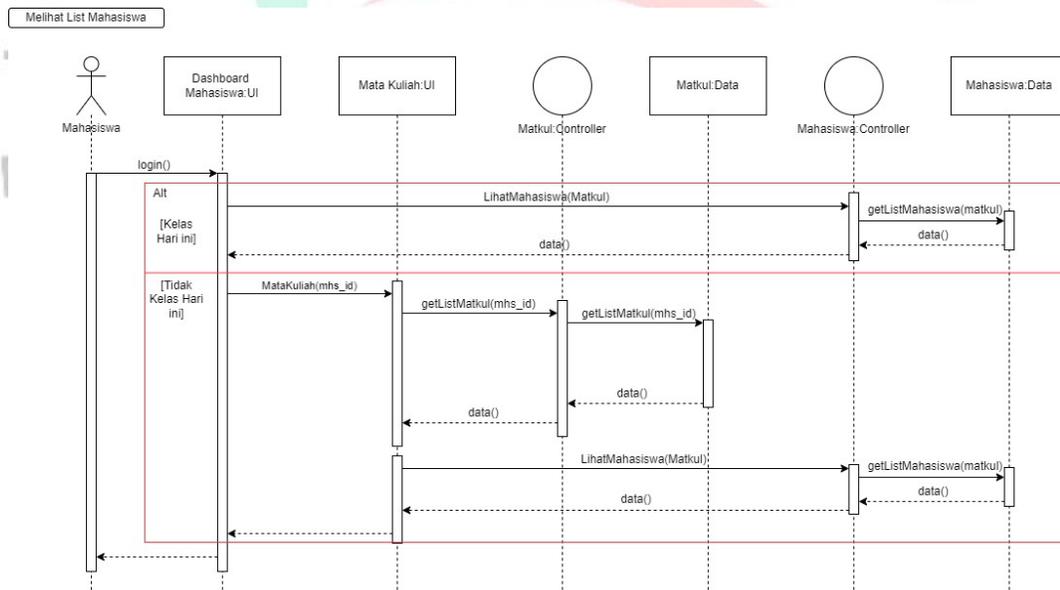
Gambar 4. 16 Sequence Diagram Melihat List Mata Kuliah Diajar



Gambar 4. 17 Sequence Diagram Melihat List Mahasiswa (Dosen)



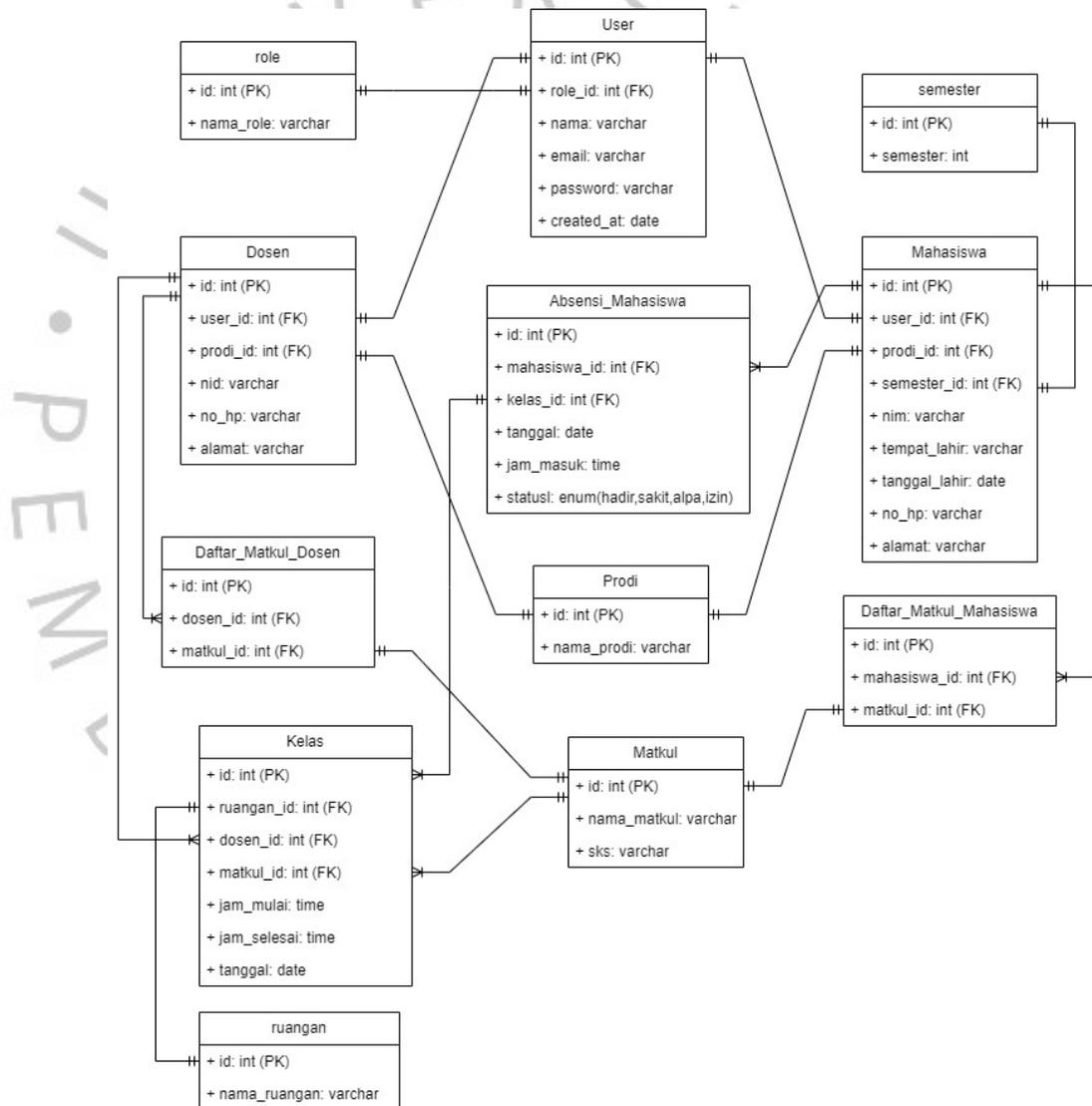
Gambar 4. 18 *Sequence Diagram* Melihat List Mata Kuliah



Gambar 4. 19 *Sequence Diagram* Melihat List Mahasiswa (Mahasiswa)

4.2.5 Class Diagram

Berfungsi memberikan penggambaran dari struktur data di dalam aplikasi presensi yang saat ini dikembangkan. Diagram ini memodelkan *class*, *attribute*, metode, dan relasi setiap kelas pada aplikasi presensi. *Class Diagram* dibuat untuk menggambarkan keperluan *database* secara terperinci dan jelas. **Gambar 4.21.** menunjukkan visualisasi dari diagram kelas yang dibangun.



Gambar 4. 20 Class Diagram

4.2.6 Database Specification

Database Specification merupakan spesifikasi rinci atau detail dari setiap tabel pada *database* yang digunakan dalam sistem presensi. *Database Specification* digunakan untuk mempermudah pengembang aplikasi dalam pembuatan *database*. *Database Specification* divisualisasikan dalam tabel-tabel berikut:

Tabel 4. 17 Tabel *User*

No.	Nama Field	Type	Panjang	Keterangan
1.	id	bigint	20	primary key
2.	role_id	bigint	20	foreign key dari tabel roles
3.	name	varchar	255	nama user
4.	email	varchar	255	email user untuk login
5.	password	varchar	255	password user untuk login

Tabel 4. 18 Tabel Mahasiswa

No.	Nama Field	Type	Panjang	Keterangan
1.	id	bigint	20	primary key
2.	user_id	bigint	20	foreign key dari tabel users
3.	prodi_id	bigint	20	foreign key dari tabel prodi
4.	semester_id	bigint	20	foreign key dari tabel semester
5.	nim	varchar	255	nomor induk mahasiswa
6.	tempat_lahir	varchar	255	tempat lahir mahasiswa
7.	tanggal_lahir	date		tanggal lahir

				mahasiswa
8.	no_hp	varchar	255	nomor ponsel mahasiswa
9.	alamat	varchar	255	alamat mahasiswa

Tabel 4. 19 Tabel Dosen

No.	Nama Field	Type	Panjang	Keterangan
1.	id	bigint	20	primary key
2.	user_id	bigint	20	foreign key dari tabel roles
3.	prodi_id	bigint	20	foreign key dari tabel prodi
4.	nid	varchar	255	nomor induk dosen
5.	no_hp	varchar	255	nomor ponsel dosen
6.	alamat	varchar	255	alamat dosen

Tabel 4. 20 Tabel Presensi Mahasiswa

No.	Nama Field	Type	Panjang	Keterangan
1.	id	bigint	20	primary key
2.	mahasiswa_id	bigint	20	foreign key dari tabel mahasiswa
3.	kelas_id	bigint	20	foreign key dari tabel kelas
4.	tanggal	date		tanggal presensi dilakukan
5.	jam_masuk	time		waktu presensi

				dilakukan
6.	status	enum('Hadir', 'Sakit', 'Izin', 'Alfa')		status presensi mahasiswa

Tabel 4. 21 Tabel Kelas

No.	Nama Field	Type	Panjang	Keterangan
1.	id	bigint	20	primary key
2.	ruangan_id	bigint	20	foreign key dari tabel ruangan
3.	dosen_id	bigint	20	foreign key dari tabel dosen
4.	matkul_id	bigint	20	foreign key dari tabel matkul
5.	jam_mulai	time		jam dimulainya kelas
6.	jam_selesai	time		jam selesainya kelas
7.	tanggal	date		tanggal kelas diadakan

Tabel 4. 22 Tabel Matkul

No.	Nama Field	Type	Panjang	Keterangan
1.	id	bigint	20	primary key
2.	nama_matkul	varchar	255	nama mata kuliah
3.	sks	varchar	10	jumlah sks mata kuliah

Tabel 4. 23 Tabel Daftar Matkul Mahasiswa

No.	Nama Field	Type	Panjang	Keterangan
1.	id	bigint	20	primary key
2.	mahasiswa_id	bigint	20	foreign key dari tabel mahasiswa
3.	matkul_id	bigint	20	foreign key dari tabel matkul

Tabel 4. 24 Tabel Daftar Matkul Dosen

No.	Nama Field	Type	Panjang	Keterangan
1.	id	bigint	20	primary key
2.	dosen_id	bigint	20	foreign key dari tabel roles
3.	matkul_id	bigint	20	foreign key dari tabel matkul

Tabel 4. 25 Tabel Prodi

No.	Nama Field	Type	Panjang	Keterangan
1.	id	bigint	20	primary key
2.	nama_prodi	vachar	255	nama program studi

Tabel 4. 26 Tabel Ruangan

No.	Nama Field	Type	Panjang	Keterangan
1.	id	bigint	20	primary key
2.	ruangan	varchar	50	ruangan diadakannya kelas

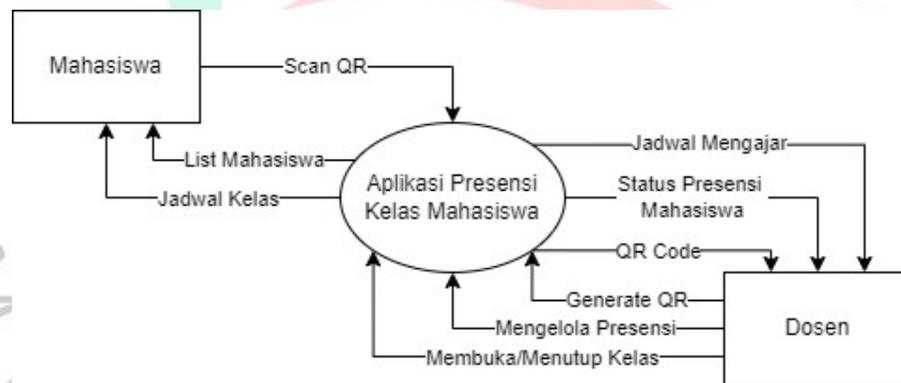
Tabel 4. 27 Tabel *Role*

No.	Nama Field	Type	Panjang	Keterangan
1.	id	bigint	20	primary key
2.	nama_role	varchar	255	nama role user

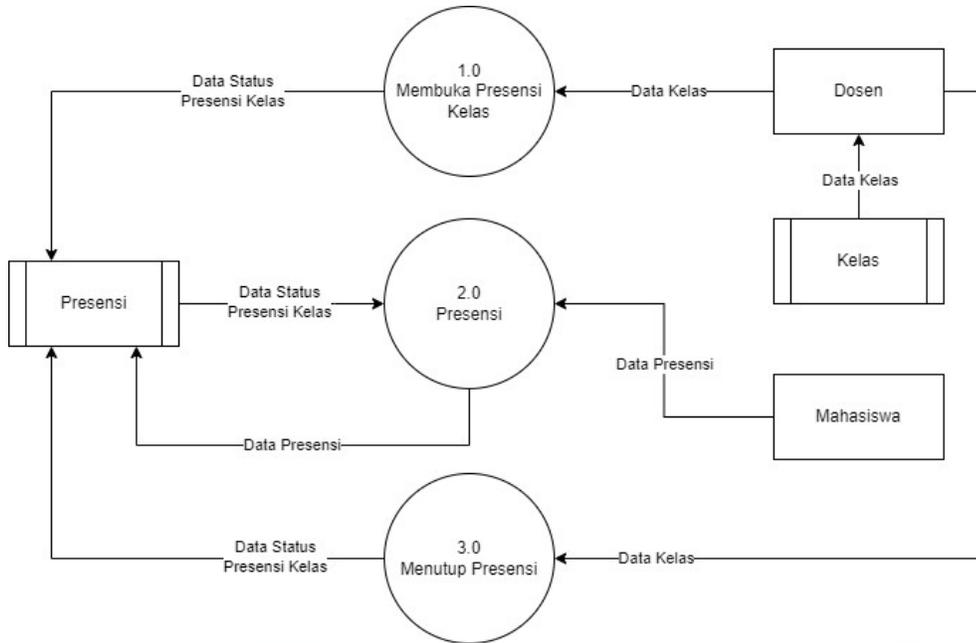
Tabel 4. 28 Tabel Semester

No.	Nama Field	Type	Panjang	Keterangan
1.	id	bigint	20	primary key
2.	semester	varchar	10	semester mahasiswa

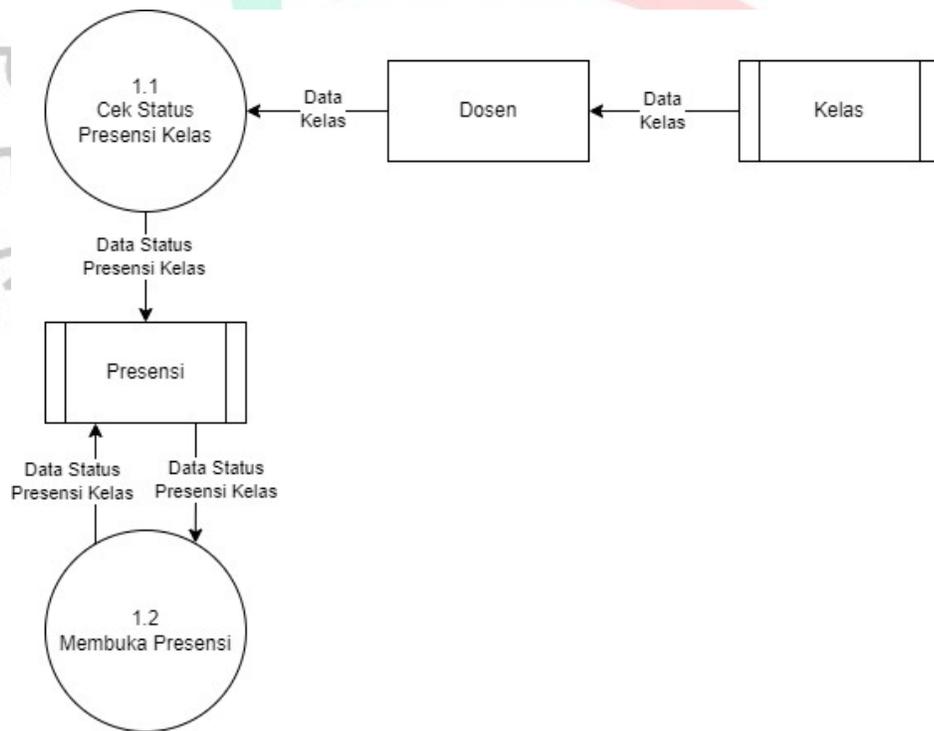
4.2.7 Data Flow Diagram (DFD)



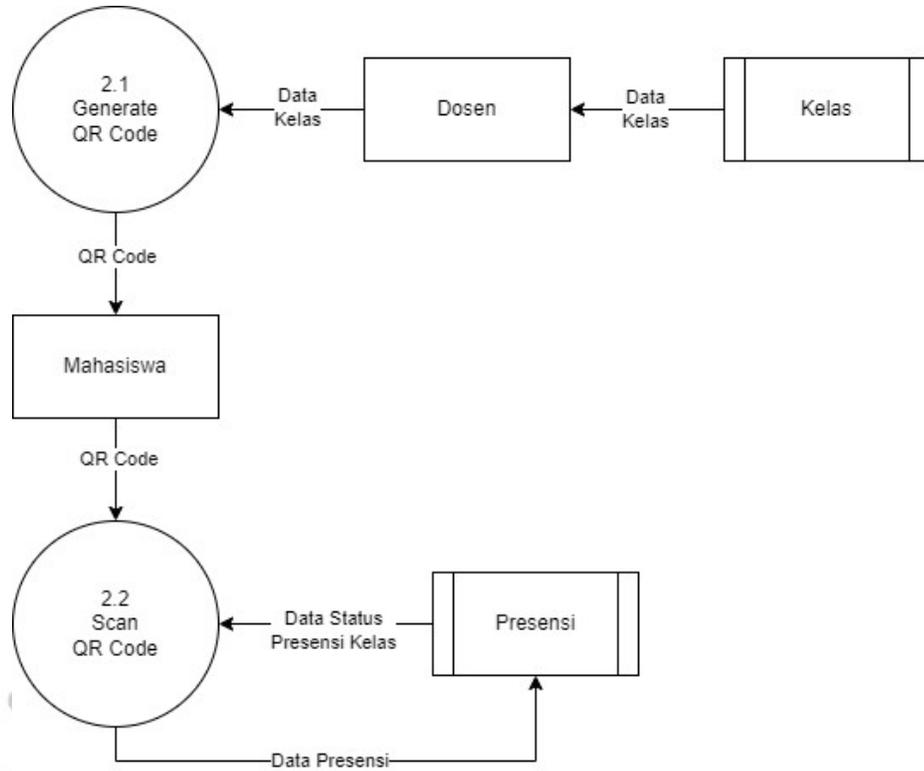
Gambar 4. 21 Diagram Konteks



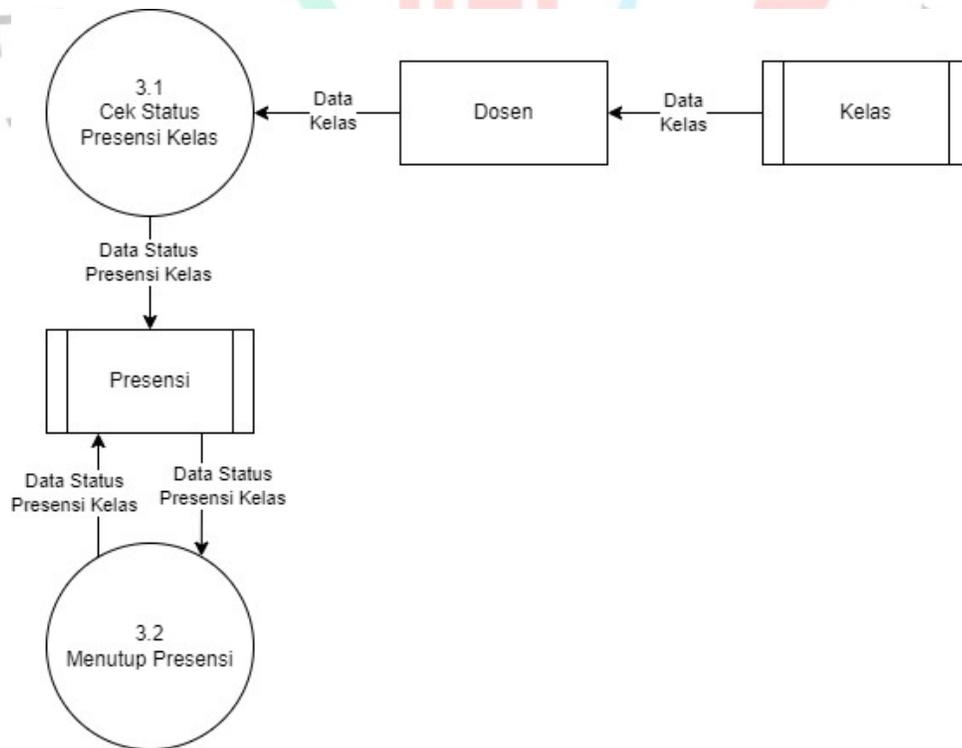
Gambar 4. 22 Data Flow Diagram Level 0



Gambar 4. 23 Data Flow Diagram Level 1 Membuka Presensi

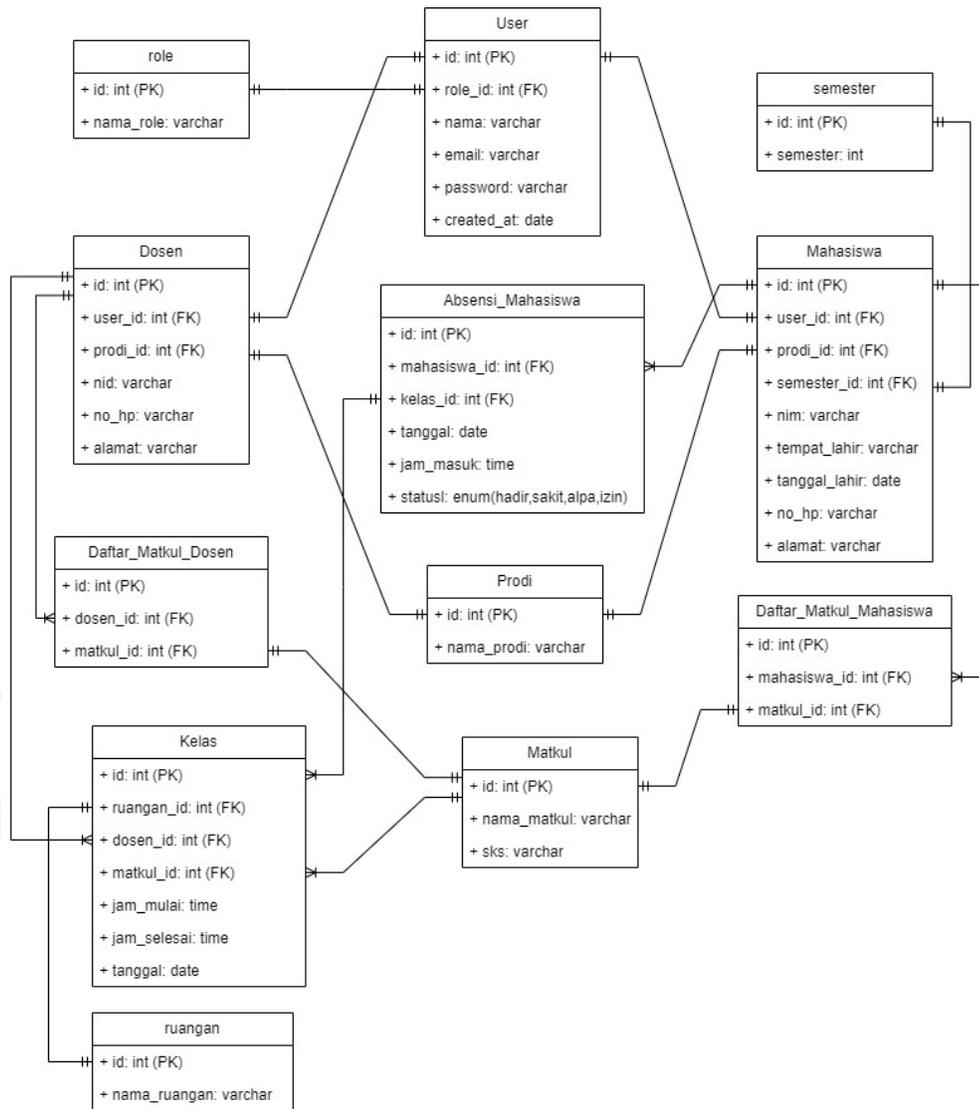


Gambar 4. 24 Data Flow Diagram Level 1 Presensi



Gambar 4. 25 Data Flow Diagram Level 1 Menutup Presensi

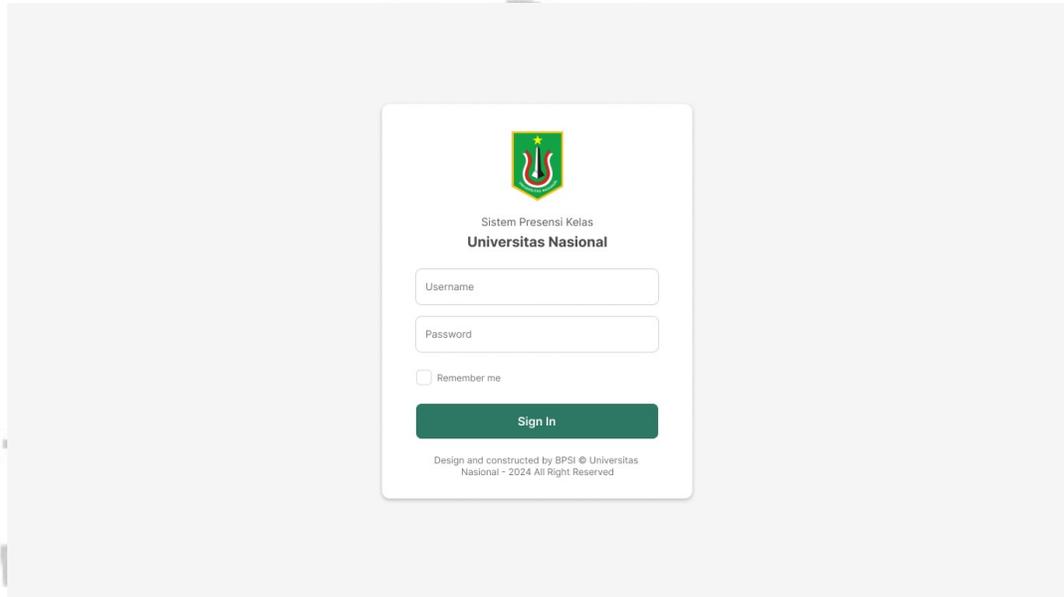
4.2.8 Entity Relationship Diagram (ERD)



Gambar 4. 26 Entity Relationship Diagram (ERD)

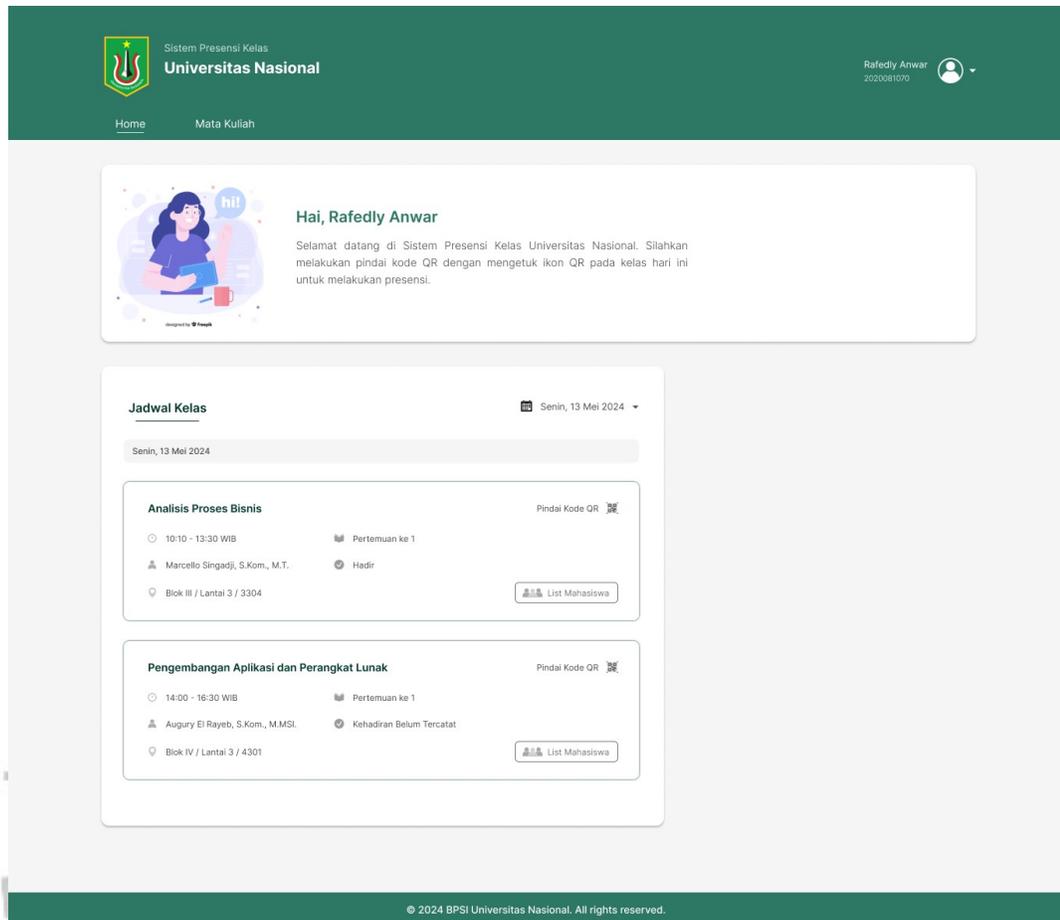
4.3 Tahap *Implementation*

Tahap ini menerjemahkan atau mengimplementasikan dokumen-dokumen arsitektur yang telah dibuat menjadi kode yang fungsional. Kode-kode berdasarkan spesifikasi dalam tahap *design* dituliskan untuk membangun aplikasi. Berikut adalah tampilan-tampilan dari hasil implementasi:



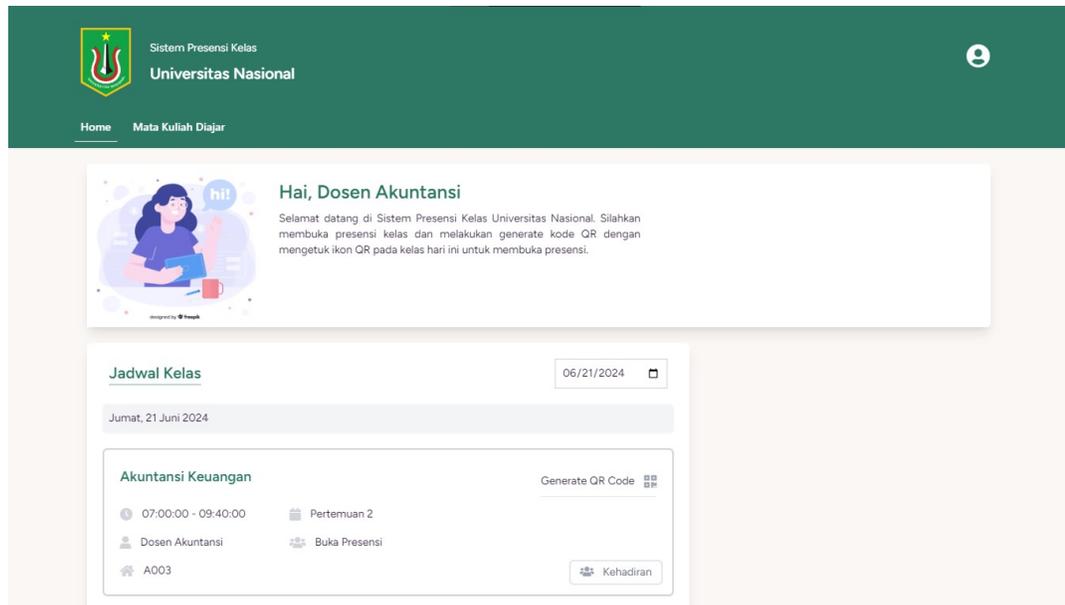
Gambar 4. 27 *Login Form*

Gambar 4.27 gambar dari *Form Login*. Pertama, *user* membuka aplikasi, tampilan *login form* adalah tampilan yang muncul pertama kali (*Landing Page*). Tampilan *login form* berisi *form* untuk *user* masuk ke dalam aplikasi dengan menginput *email* serta *password* lalu menekan tombol "Sign In".



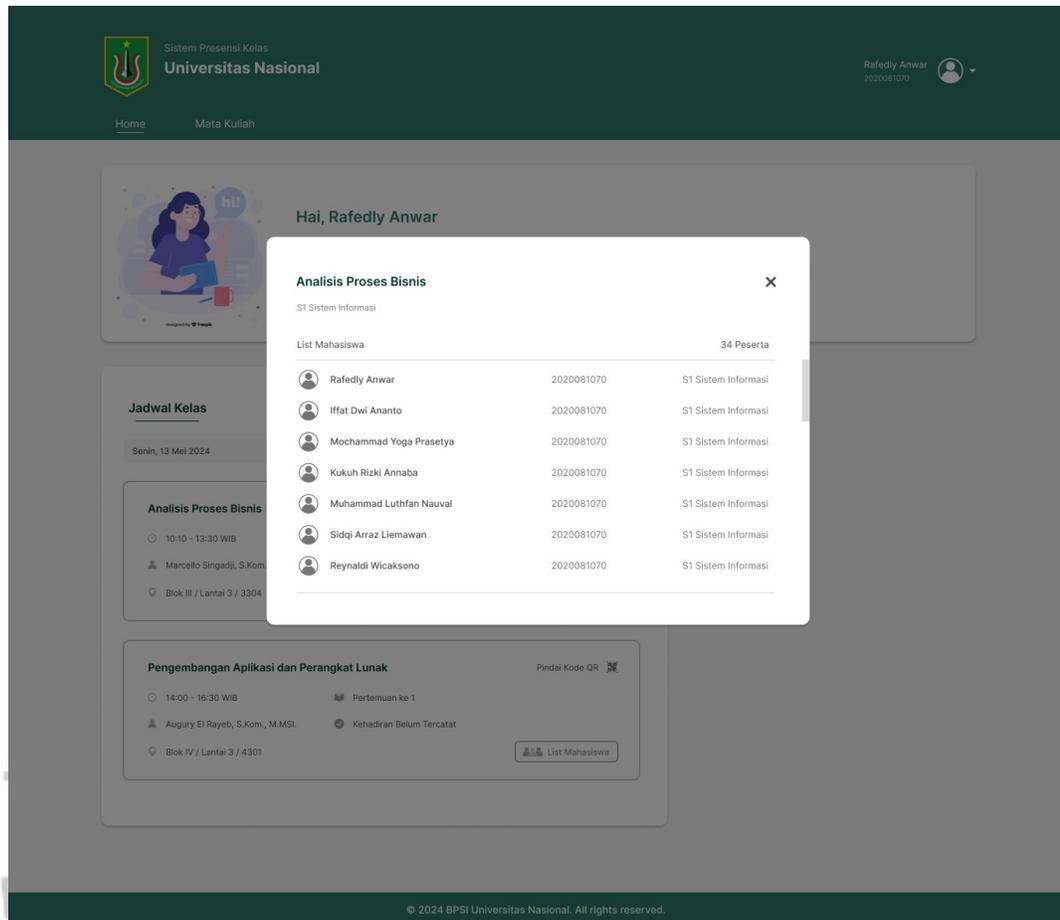
Gambar 4. 28 Tampilan *Home* Mahasiswa

Gambar 4.28 gambar dari halaman *Home* Mahasiswa. Setelah melakukan *login*, *user* yang merupakan mahasiswa akan masuk ke *home*. Halaman *home* berisi jadwal kelas yang memiliki jadwal hari ini atau saat ini. Jika hari ini tidak ada kelas, maka sistem akan menampilkan “Tidak ada jadwal kelas hari ini” di dalam jadwal kelas. Mahasiswa dapat mengganti tanggal kelas untuk melihat jadwal kelas besok atau pada tanggal tertentu. Pada setiap kelas di dalam jadwal kelas, terdapat tombol “List Mahasiswa” untuk melihat list mahasiswa pada kelas dan terdapat tombol “Pindai Kode QR” yang akan membuka kamera jika kelas sudah dibuka oleh dosen untuk melakukan presensi.



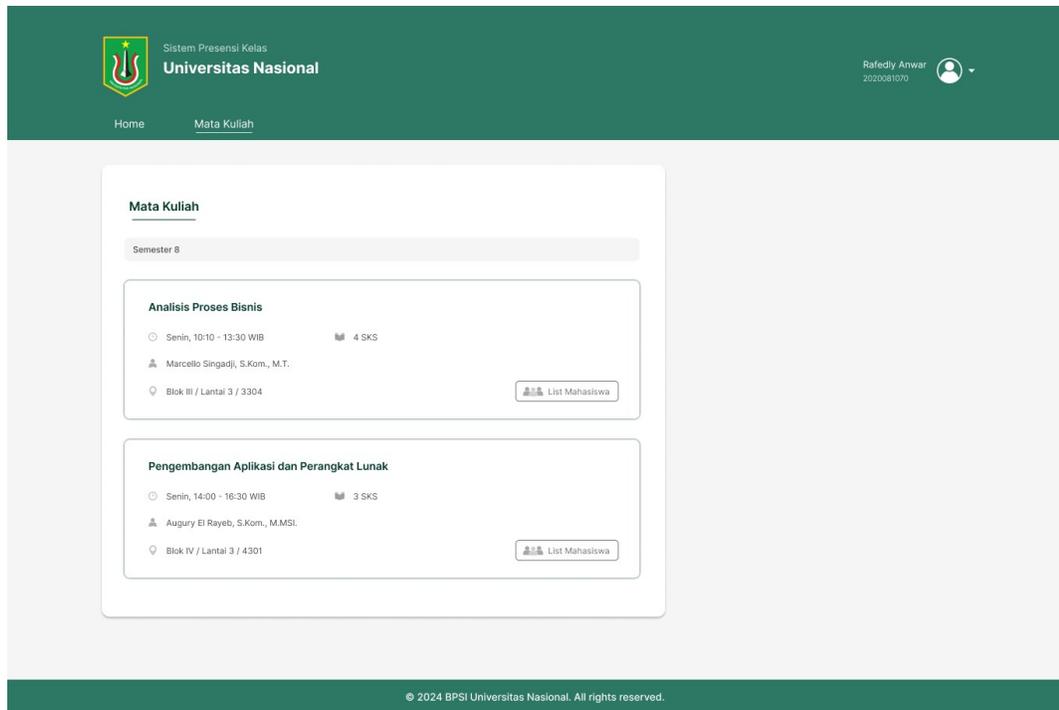
Gambar 4. 29 Tampilan *Home* Dosen

Gambar 4.29 gambar dari halaman *Home* Dosen. Setelah melakukan *login*, *user* yang merupakan mahasiswa akan masuk ke *home*. Halaman *home* berisi jadwal kelas yang diajar dosen yang memiliki jadwal hari ini atau saat ini. Jika hari ini tidak ada kelas yang diajar, maka sistem akan menampilkan “Tidak ada jadwal kelas hari ini” di dalam jadwal kelas. Dosen dapat mengganti tanggal kelas untuk melihat jadwal kelas besok atau pada tanggal tertentu. Pada setiap kelas di dalam jadwal kelas, terdapat tombol “Kehadiran” untuk membuka halaman Kelola Presensi untuk dosen mengelola presensi mahasiswa secara manual, dan tombol “Generate QR Code” untuk melakukan *generate QR code*. Lalu, juga terdapat tombol “Buka Kelas” untuk membuka kelas dan jika kelas terbuka maka terdapat tombol “Tutup Kelas” untuk menutup kelas sehingga presensi tertutup dan mahasiswa tidak dapat melakukan *scan QR code*.



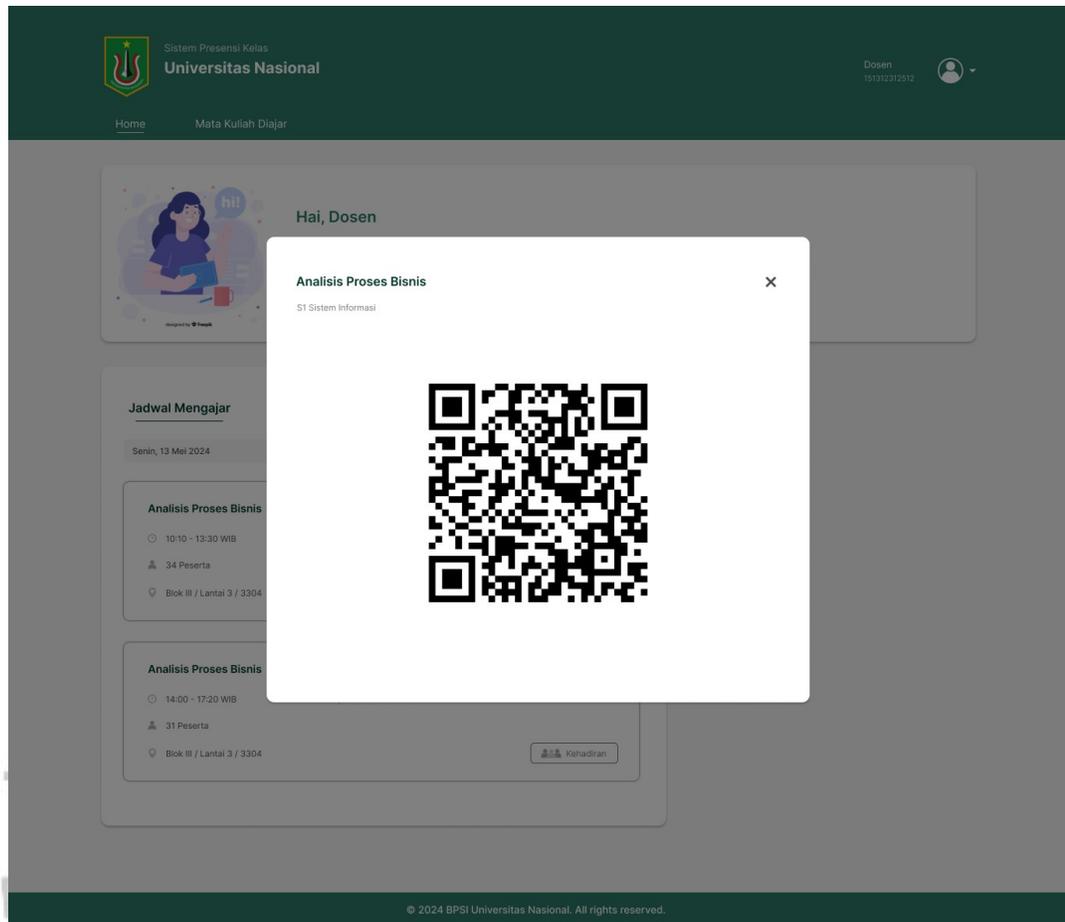
Gambar 4. 30 List Mahasiswa (Mahasiswa)

Gambar 4.30 merupakan tampilan List Mahasiswa untuk mahasiswa. Halaman ini akan tampil jika mahasiswa menekan tombol "List Mahasiswa" pada *Home*, atau mahasiswa juga dapat membuka menu "Mata Kuliah" untuk membuka halaman list mata kuliah yang berisi semua mata kuliah diambil mahasiswa ambil semester ini lalu menekan tombol "List Mahasiswa" pada mata kuliah yang diinginkan untuk menampilkan list mahasiswa di dalam mata kuliah tersebut.



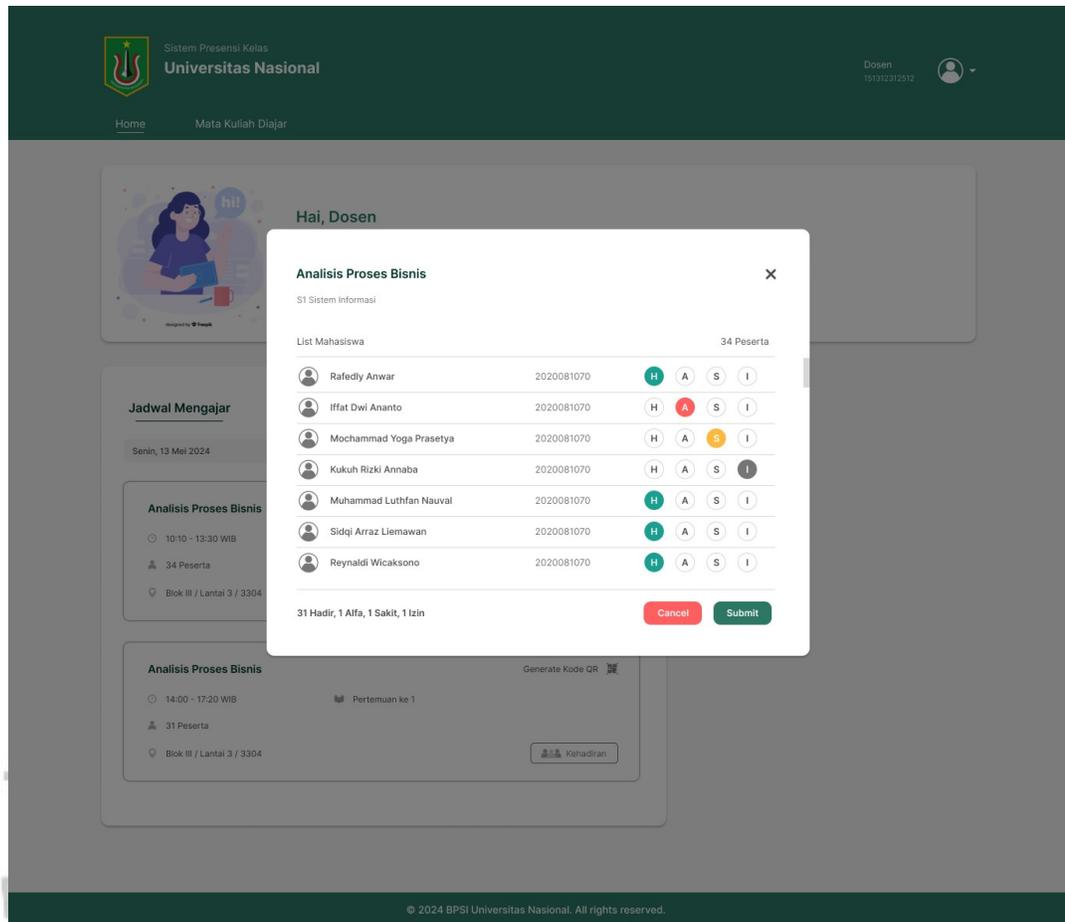
Gambar 4. 31 List Mata Kuliah

Gambar 4.31 merupakan visualisasi dari tampilan List Mata Kuliah untuk mahasiswa. Setelah melakukan *login*, jika mahasiswa memilih menu “Mata Kuliah”, maka halaman List Mata Kuliah akan tampil. Halaman ini berisi list dan detail dari seluruh matkul diambil mahasiswa di dalam semester saat ini. Pada halaman juga terlampir *button* “List Mahasiswa” dengan fungsi mengakses halaman List Mahasiswa.



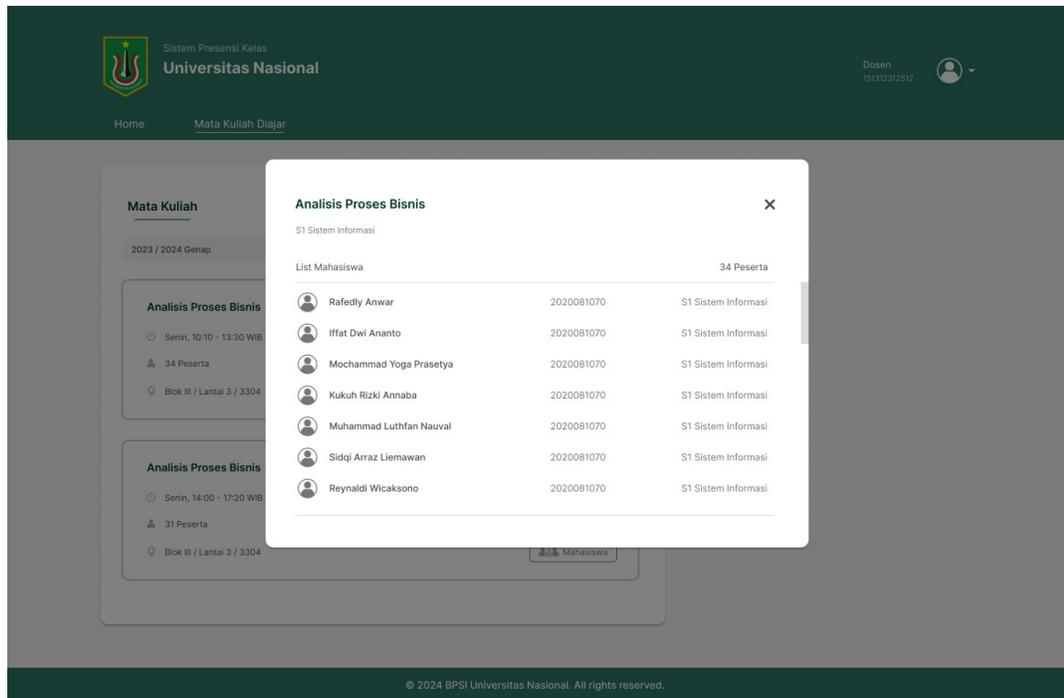
Gambar 4.32 *Generate QR Code*

Gambar 4.32 merupakan visualisasi dari halaman *Generate*. Tampilan ini akan muncul setelah dosen menekan tombol “Generate QR Code” pada mata kuliah yang berjalan hari ini dan terdapat di *home* dosen. QR Code yang tampil tersebut memiliki fungsi bagi mahasiswa melakukan *scan* dan melakukan presensi.



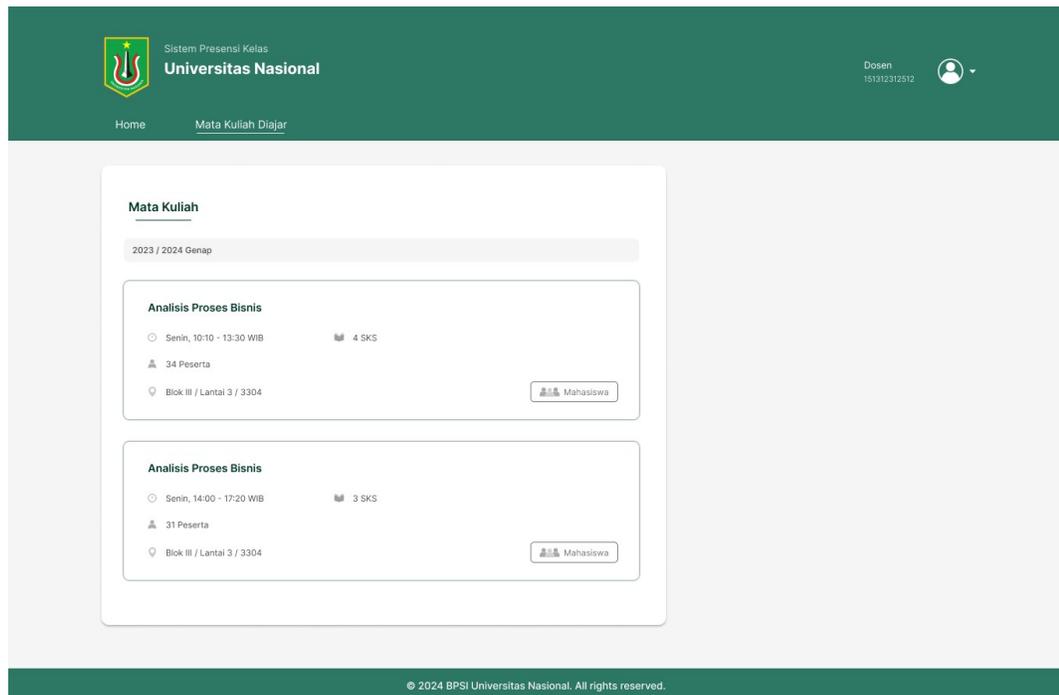
Gambar 4. 33 Tampilan Kelola Presensi

Gambar 4.33 gambar dari halaman Kelola Presensi. Ini muncul setelah dosen menekan tombol “Kehadiran” pada mata kuliah yang berjalan hari ini dan terdapat di *home* dosen. Halaman ini berisi list mahasiswa di dalam kelas serta *radio option* untuk mengubah status mahasiswa dan mengelola presensi. Halaman ini berfungsi untuk mengelola presensi mahasiswa secara manual apabila terdapat kendala mahasiswa dalam melakukan presensi dengan *QR code*.



Gambar 4. 34 List Mahasiswa (Dosen)

Gambar 4.34 merupakan visualisasi dari halaman List Mahasiswa untuk dosen. Halaman ini akan tampil jika dosen membuka menu "Mata Kuliah Diajar" untuk membuka halaman list mata kuliah yang berisi semua matkul diajar oleh dosen di dalam semester saat ini lalu menekan tombol "List Mahasiswa" pada mata kuliah yang diinginkan untuk menampilkan list mahasiswa di dalam mata kuliah tersebut.



Gambar 4. 35 Tampilan List Mata Kuliah Diajar

Gambar 4.35 merupakan gambar dari tampilan List Mata Kuliah Diajar oleh dosen. Setelah melakukan *login*, jika dosen memilih menu “Mata Kuliah Diajar”, maka halaman List Mata Kuliah Diajar akan tampil. Halaman ini berisi list dan detail dari seluruh mata kuliah yang diajar oleh dosen pada semester berjalan. Pada halaman ini terlampir *button* “List Mahasiswa” dengan fungsi memberikan tampilan halaman List Mahasiswa di dalam kelas mata kuliah tertentu.

4.4 Tahap *Verification*

Tahap akhir di dalam rancang bangun aplikasi presensi adalah *verification*. Tahap *verification* merupakan tahap di mana aplikasi telah selesai dibuat dan akan diuji atau dilakukan *testing*. Berdasarkan hasil pengujian, setiap skenario berhasil sesuai dengan harapan di dalam rancangan skenario. Rancangan skenario pengujian digambarkan dalam **Tabel 4.30**.

Tabel 4. 29 Skenario Pengujian

No.	Test ID	Test Condition	Scenario	Expected Result
1.	LOGIN_POS	Positif	Mengisi form login dengan username dan password yang benar	Login berhasil
2.	LOGIN_NEG_USE RNAME_FLS	Negatif	Mengisi form login dengan username salah	Login gagal
3.	LOGIN_NEG_PAS SWORD_FLS	Negatif	Mengisi form login dengan password salah	Login gagal
4.	LOGIN_NEG_USE RNAME_PASSWO RD_FLS	Negatif	Mengisi form login dengan username dan password salah	Login gagal
5.	DATE_EDIT_POS	Positif	Mengubah tanggal pada jadwal kelas ke tanggal yang user memiliki jadwal kelas	Menampilkan kelas sesuai tanggal yang dipilih
6.	DATE_EDIT_NEG	Negatif	Mengubah tanggal pada jadwal kelas ke tanggal yang user tidak memiliki jadwal kelas	Menampilkan “Tidak ada jadwal kelas hari ini” pada tanggal yang dipilih
7.	LIST_MHS	Positif	Menekan button “List Mahasiswa”	Menampilkan list mahasiswa pada kelas yang dipilih
8.	SCAN_QR_POS_1	Positif	Menekan button “Scan QR Code”	Membuka kamera

9.	SCAN_QR_POS_2	Positif	Melakukan scan terhadap QR Code	Sistem menyimpan presensi
10.	SCAN_QR_NEG_1	Negatif	Melakukan scan terhadap QR Code yang tidak sesuai	Menampilkan alert “QR Code tidak sesuai”
11.	SCAN_QR_NEG_2	Negatif	Menekan button “Scan QR Code” saat kelas belum dibuka	Menampilkan alert “Presensi belum dibuka”
12.	SCAN_QR_NEG_3	Negatif	Melakukan scan terhadap QR Code lebih dari 1x	Menampilkan alert “Anda telah melakukan presensi.”
13.	GENERATE_QR	Positif	Menekan button “Generate Qr Code”	Menampilkan halaman berisi QR Code
14.	BUKA_PRESENSI	Positif	Menekan buton “Buka Presensi”	Sistem menyimpan status presensi terbuka dan button menjadi “Tutup Presensi”
15.	TUTUP_PRESENSI	Positif	Menekan button “Tutup Presensi”	Sistem menyimpan status presensi tertutup dan

				button menjadi “Buka Presensi”
16.	KELOLA_KEHADIRAN	Positif	Menekan tombol “Kehadiran”	Menampilkan halaman berisi list mahasiswa dan opsi untuk mengelola kehadiran mahasiswa
17.	KELOLA_KEHADIRAN_UPDATE	Positif	Mengubah status kehadiran mahasiswa dengan mengisi opsi pada radio option dan menekan tombol “Submit”	Sistem menyimpan status kehadiran mahasiswa
18.	KELOLA_KEHADIRAN_HADIR_SEMUA	Positif	Menekan tombol “Hadir Semua” dalam halaman kelola kehadiran	Status presensi semua mahasiswa di dalam kelas menjadi “Hadir”