BAB IV

ANALISIS DAN PERANCANGAN SISTEM

4.1 Analisis Sistem Yang Sedang Berjalan

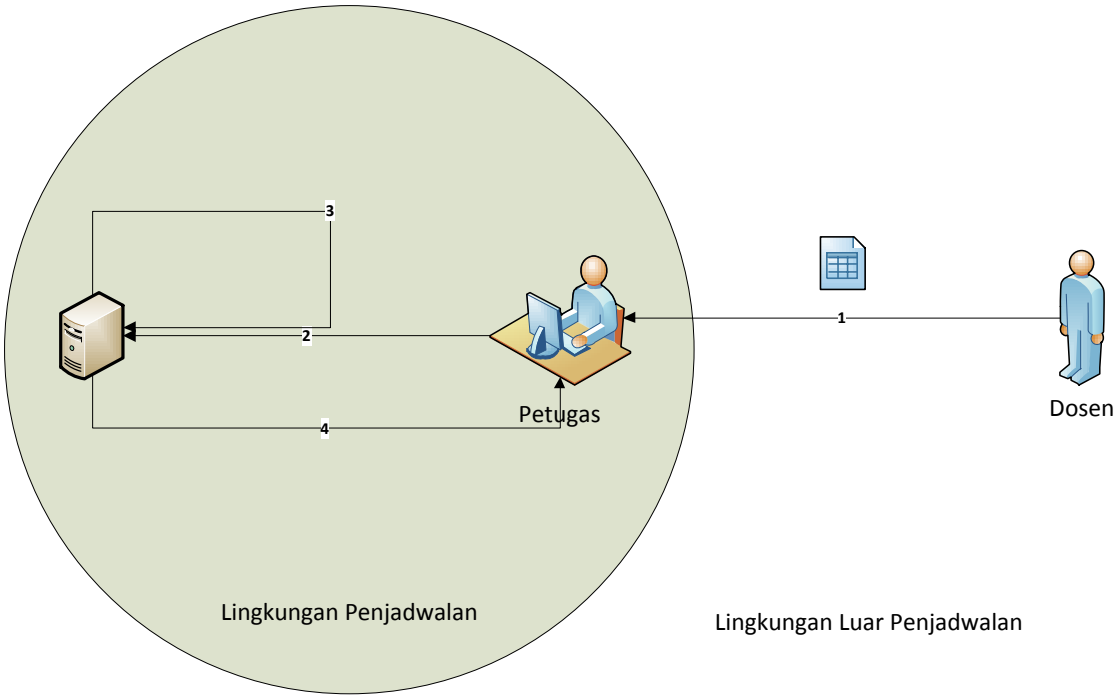
Kegiatan analisis sistem yang berjalan dilakukan dengan analisis yang berorientasi pada objek-objek yang diperlukan oleh sistem yang dirancang, dimaksudkan untuk menitik beratkan kepada fungsi sistem yang berjalan dengan tidak terlalu menitik beratkan kepada alur proses dari sistem. Selanjutnya hasil analisis ini digambarkan dan didokumentasikan dengan metodologi berorientasi objek melalui diagram *use case*, skenario *use case* dan aktifitas diagram. Pertimbangan penggunaan diagram ini karena dianggap diagram-diagram tersebut mewakili secara keseluruhan sistem yang berjalan dan dapat dimengerti oleh *user*.

4.1.1 Analisis Kebutuhan

Sebelum membuat suatu sistem, hendaknya melakukan analisis terlebih dahulu terhadap kebutuhan-kebutuhan apa saja yang diperlukan dengan menggunakan metode-metode yang telah ada.

4.1.1.1 Proses Bisnis

Proses bisnis dari sistem yang sedang berjalan adalah sebagai berikut:



Gambar 4. 1 Proses Bisnis sistem yang sedang berjalan

Keterangan:

1. Dosen Mengisi data penawaran matakuliah dan menyerahkannya pada petugas penjadwalan kuliah.
2. Petugas memasukkan data jadwal kuliah sesuai dengan data penawaran matakuliah dari dosen.
3. Sistem melakukan pemeriksaan apakah data jadwal yang dimasukkan oleh penjadwal tidak ada yang mengalami bentrok. Jika jadwal ada yang bentrok maka sistem akan meminta petugas untuk memasukkan data jadwal kuliah kembali (proses no 2). Jika jadwal tidak ada yang bentrok maka sistem akan mencetak jadwal kuliah (proses no 4).
4. Sistem akan mencetak data penjadwalan kuliah berupa Jadwal Kuliah, BAP (Berita Acara Perkuliahan), dan Tugas Mengajar.

4.1.1.2 Service Time

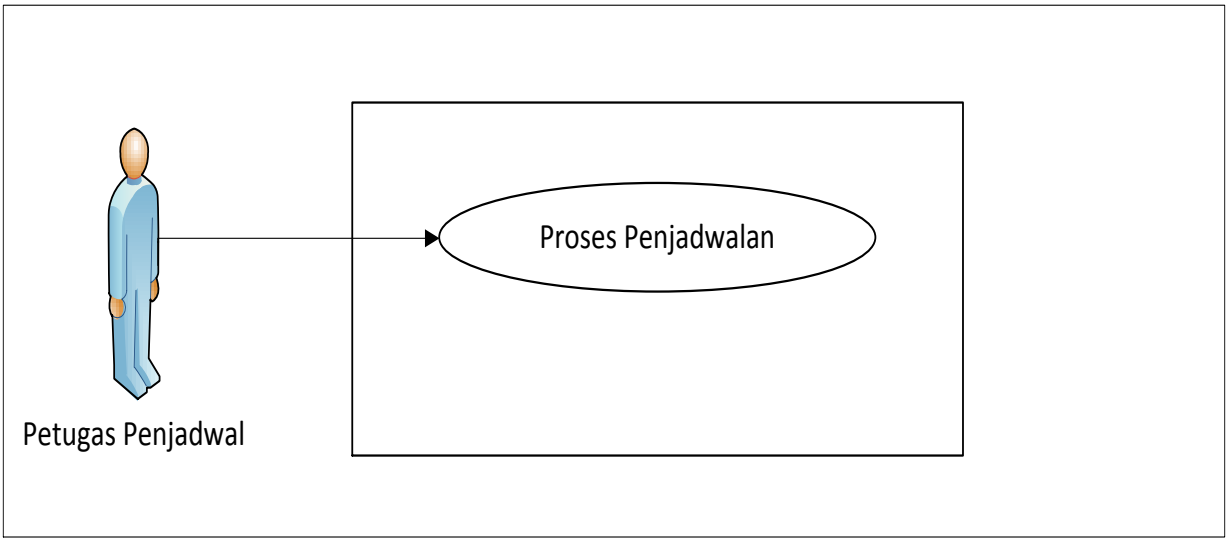
Sistem penjadwalan yang ada pada saat ini di program studi sistem informasi UNIKOM masih merupakan sistem penjadwalan yang semi manual. Maksudnya semi manual adalah proses pemasangan jadwal kuliah untuk ruang kuliah, dosen, kelas, dan mata kuliah masih dilakukan secara manual, tetapi ada applikasi yang digunakan untuk melakukan pengecekan akan mata kuliah yang berbentrokan, tetapi untuk perbaikannya masih juga dilakukan secara manual.

Proses yang dijalankan masihlah belum efisien dari segi waktu. Petugas harus memeriksa dan mencoba untuk membuat jadwal untuk setiap dosen, ruang, dan matakuliah agar tidak terjadi bentrok.

4.1.1.3 Use Case Diagram

Use case diagram (diagram *use case*) adalah diagram yang menyajikan interaksi antara *use case* dan *actor*. Dimana *actor* dapat berupa orang, peralatan atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun. *Use case* menggambarkan fungsionalitas sistem atau persyaratan – persyaratan yang harus dipenuhi sistem dari pandangan pemakai.

Berilkut ini adalah gambar model *Use Case Diagram* pembuatan jadwal kuliah yang ada di Sistem Informasi Unikom yang sedang berjalan :



Gambar 4. 2 *Use Case* Sistem Penjadwalan yang sedang berjalan

4.1.1.4 Skenario *Use Case*

Skenario *Use Case* digunakan untuk memudahkan dalam menganalisa skenario yang akan kita gunakan pada fase-fase selanjutnya dengan melakukan penilaian terhadap skenario tersebut. Adapun tahapan dari skenario *use case* pada sistem penjadwalan yang sedang berjalan adalah sebagai berikut:

Nama *use case*: Proses Penjadwalan

Tabel 4. 1 Skenario *use case* sistem yang sedang berjalan

Judul	Proses Penjadwalan
No. *Use case*	UC-JADWAL-01
Deskripsi *Use Case*	Aktor Mengisi data penjadwalan
Aktor	Penjadwal
Trigger	Aktor memasukkan data jadwal kuliah.
Kondisi Awal	Aplikasi menampilkan *form* untuk mengisi data penjadwalan dosen dan kelas.

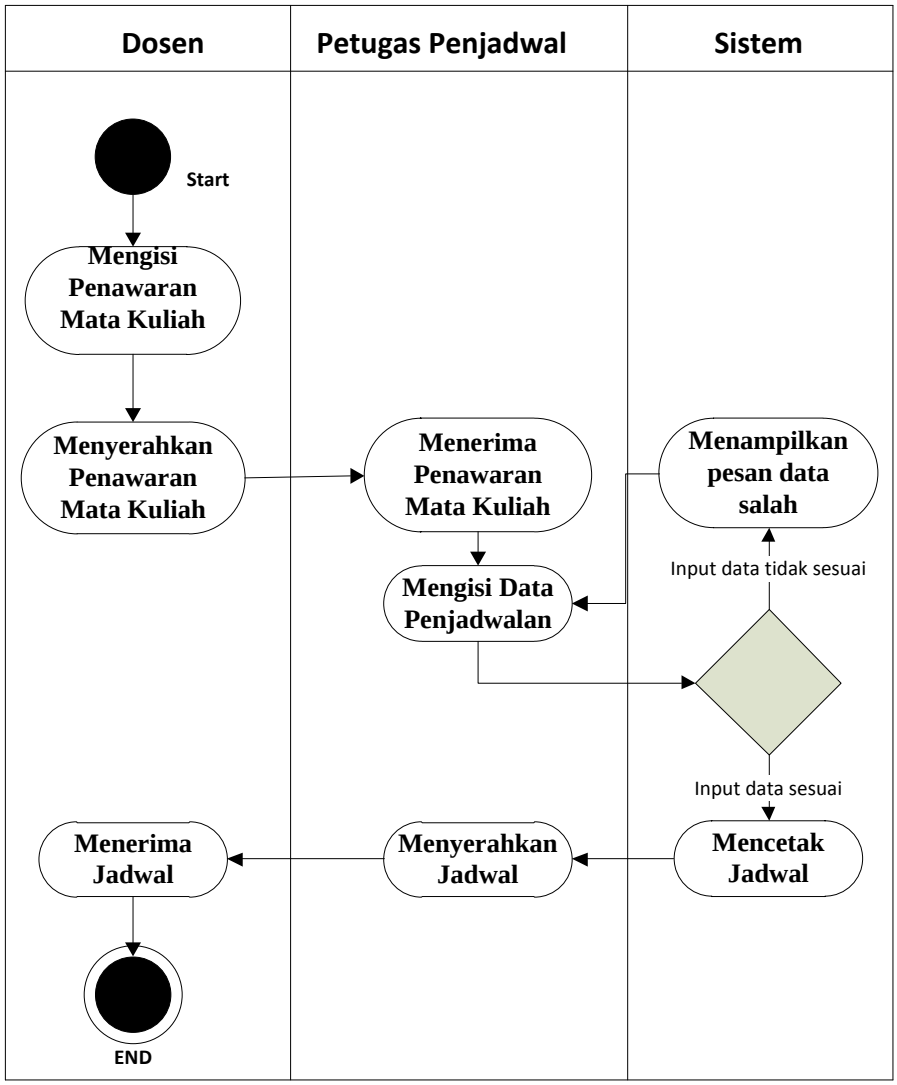
<table>
<tr><th colspan="2">Skenario Normal</th></tr>
<tr><th>Aksi- Aktor</th><th>Reaksi Sistem</th></tr>
<tr><td>1. Aktor mengisi field kosong pada form yang meminta data dosen dan kelas.</td><td></td></tr>
<tr><td>2. Aktor men-submit data dengan menekan sebuah button untuk autentikasi.</td><td></td></tr>
<tr><td></td><td>3. Memeriksa kebenaran data yang digunakan aktor.</td></tr>
<tr><td></td><td>4. Sistem :
a. Menerima data penjadwalan.
b. Menolak data penjadwalan.</td></tr>
<tr><td colspan="2">Kondisi Akhir Skenario Normal:

1. Aplikasi:
a. Mengenerate hasil data penjadwalan.
b. Menampilkan form untuk pengisian data penjadwalan kembali.</td></tr>
<tr><th colspan="2">Skenario Abnormal-1</th></tr>
<tr><th>Aksi- Aktor</th><th>Reaksi Sistem</th></tr>
<tr><td>1. Aktor men-submit data dengan menekan sebuah button tanpa mengisi data.</td><td></td></tr>
<tr><td></td><td>2. Memeriksa kebenaran data yang digunakan aktor.</td></tr>
<tr><td></td><td>3. Menampilkan pesan informasi bahwa data masih kosong.</td></tr>
<tr><td colspan="2">Kondisi Akhir Skenario Abnormal-1:

Aplikasi menampilkan form untuk pengisian data penjadwalan kembali.</td></tr>
<tr><th colspan="2">Catatan</th></tr>
<tr><td colspan="2">Untuk pengisian data penjadwalan, applikasi hanya akan memeriksa apakah data yang dimasukkan akan bentrok atau tidak, yang memperbaiki dan men-generate data penjadwalan adalah aktor, sehingga proses ini memakan waktu yang cukup lama.</td></tr>
</table>

4.1.1.5 Activity Diagram

Pada bagian ini akan digambarkan dokumentasi alur kerja pada sistem yang sedang berjalan yang bertujuan untuk melihat alur proses sistem yang sedang berjalan



Ga

mbar 4. 3 Activity Diagram Sistem Penjadwalan yang sedang berjalan

4.1.2 Evaluasi Sistem Yang Sedang Berjalan

Setelah dilakukan penelitian pada proses penjadwalan di Program Studi Sistem Informasi Unikom, terdapat beberapa kelemahan yang ditemukan pada sistem yang sedang berjalan. Kelemahan-kelemahan tersebut adalah sebagai berikut:

1. Sistem yang dilakukan masih memasangkan jadwal secara manual di mana ada applikasi yang berfungsi untuk melakukan pengecekan jadwal agar tidak berbentrokan.
2. Penjadwal sering kesulitan dalam memasangkan jadwal dari dosen karena adanya dosen yang hanya bisa mengajar pada waktu-waktu tertentu atau mengajar lebih dari satu mata kuliah.
3. Penjadwal sering kesuliatan dalam penentuan ruang kelas karena adanya ruangan-ruangan tertentu yang hanya tersedia pada waktu tertentu.
4. Penjadwal sering kesulitan dalam pemasangan dosen, ruang kuliah, dan mata kuliah terkhusus pada kelas beasiswa unggulan dan international program.
5. Pembuatan jadwal kuliah yang ada kerap memakan waktu yang cukup lama karena sistem yang ada hanya melakukan pengecekan bentrokan, bukan memperbaiki atau menyusun jadwal kuliah.
6. Akibat dari jadwal kuliah yang lama dikeluarkan, maka sering kali mahasiswa harus melakukan perbaikan *form* rencana studi, dan dosen juga sering terlambat mendapatkan jadwal sehingga pada minggu-minggu awal sering ada dosen yang tidak masuk.

Melihat kasus yang terjadi diatas untuk itu penulis mencoba memberikan solusi untuk mengatasi permasalahan tersebut, yakni dengan membangun sebuah sistem yang dapat menangani:

1. Adanya matakuliah yang tidak boleh dijadwalkan pada saat yang bersamaan.
2. Adanya keinginan dosen untuk mengajar pada waktu-waktu tertentu.
3. Adanya keterbatasan akan waktu dan ruang kelas.
4. Adanya pembagian jenis kelas mahasiswa dan tipe ruang kelas.
5. Adanya dosen yang mengajar lebih dari 1 matakuliah pada semester yang sama.
6. Adanya pemilihan untuk dosen mengajar di kelas-kelas tertentu
7. Dosen tidak boleh mengajar lebih dari 6 sks secara berturut-turut

4.2 Perancangan Sistem

Pada tahap perancangan sistem ini akan dijelaskan mengenai perancangan sistem pada objek yang digunakan, perancangan arsitektur program yang akan dibuat, perancangan tampilan dan perancangan menu.

4.2.1 Tujuan Perancangan Sistem yang Diusulkan

Perancangan sistem merupakan suatu kegiatan pengembangan prosedur dan proses yang sedang berjalan untuk menghasilkan sesuatu yang baru atau memperbaharui sistem yang ada untuk meningkatkan kinerja sistem itu sendiri, agar dapat memenuhi hasil yang diinginkan. Rancangan sistem yang baru, akan diterapkan suatu kegiatan untuk menemukan dan mengembangkan metoda, prosedur dan proses suatu data agar tujuan dari suatu organisasi dapat tercapai.

Adapun tujuan dari perancangan sistem ini adalah untuk menghasilkan suatu rancangan sistem yang memperbaharui sistem yang sudah ada untuk memaksimalkan proses penjadwalan yang sedang berjalan.

4.2.2 Gambaran Umum Sistem yang Diusulkan

Gambaran umum tentang sistem yang diusulkan pada proses perancangan ini adalah suatu sistem yang digunakan untuk membuat jadwal kuliah atau sistem penjadwalan dengan harapan mampu menangani permasalahan yang ada pada sistem yang sedang berjalan pada Program Studi Sistem Informasi Unikom, terkhusus pada penanganan optimalisasi waktu yang dibutuhkan untuk pembuatan jadwal. Sistem ini diharapkan akan membantu efektifitas dan optimalisasi waktu pada saat pembuatan jadwal kuliah. Sistem informasi ini diharapkan dapat

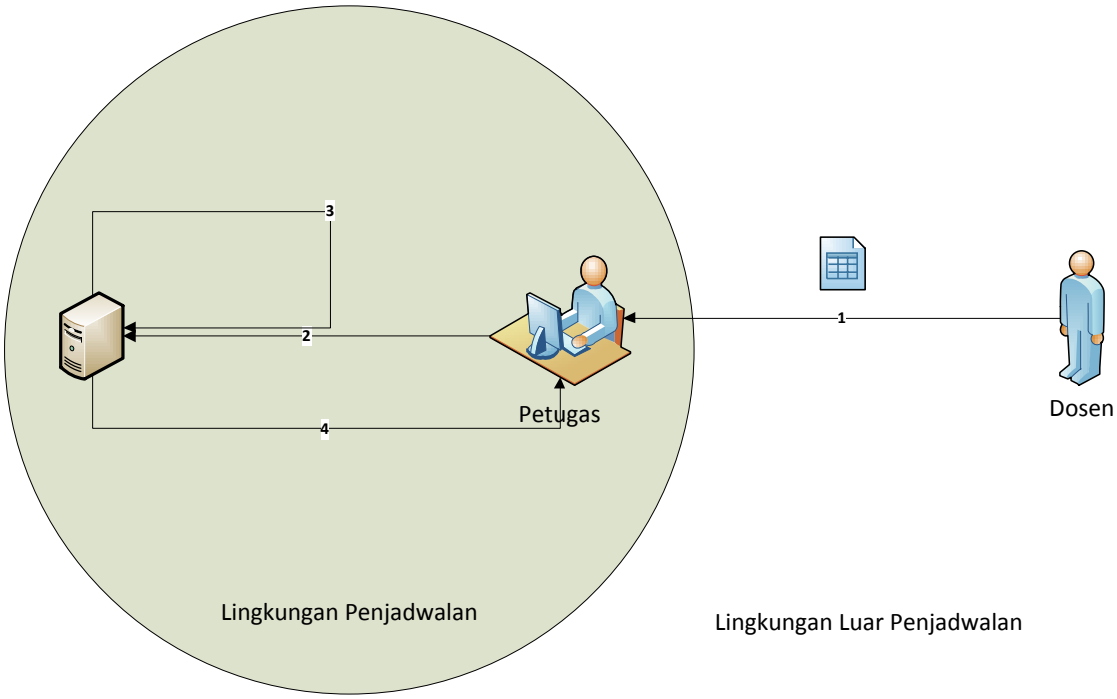
membantu mengatasi masalah yang ada, dan dapat menghasilkan informasi yang cepat, tepat dan akurat.

4.2.3 Perancangan Prosedur yang Diusulkan

Perancangan Prosedur merupakan awal dari pembuatan sistem yang akan dibuat, dimana dapat dilihat proses-proses apa saja yang nantinya diperlukan dalam pembuatan suatu sistem. Sedangkan perancangan prosedur yang diusulkan merupakan tahap untuk memperbaiki atau meningkatkan efisiensi kerja. Tahap perancangan sistem yang digambarkan merupakan tahap perancangan untuk membangun suatu sistem dan mengkonfigurasikan komponen-komponen perangkat lunak dan perangkat keras sehingga menghasilkan sistem yang baik. Sistem yang dirancang tersebut menjadi satu komponen. Tahapan perancangan prosedur ini akan dijelaskan dengan menggunakan pemodelan sistem informasi berorientasi objek dengan UML.

4.2.3.1 Proses Bisnis

Proses bisnis dari target sistem adalah sebagai berikut:



Gambar 4. 4 Proses Bisnis Target Sistem

Keterangan:

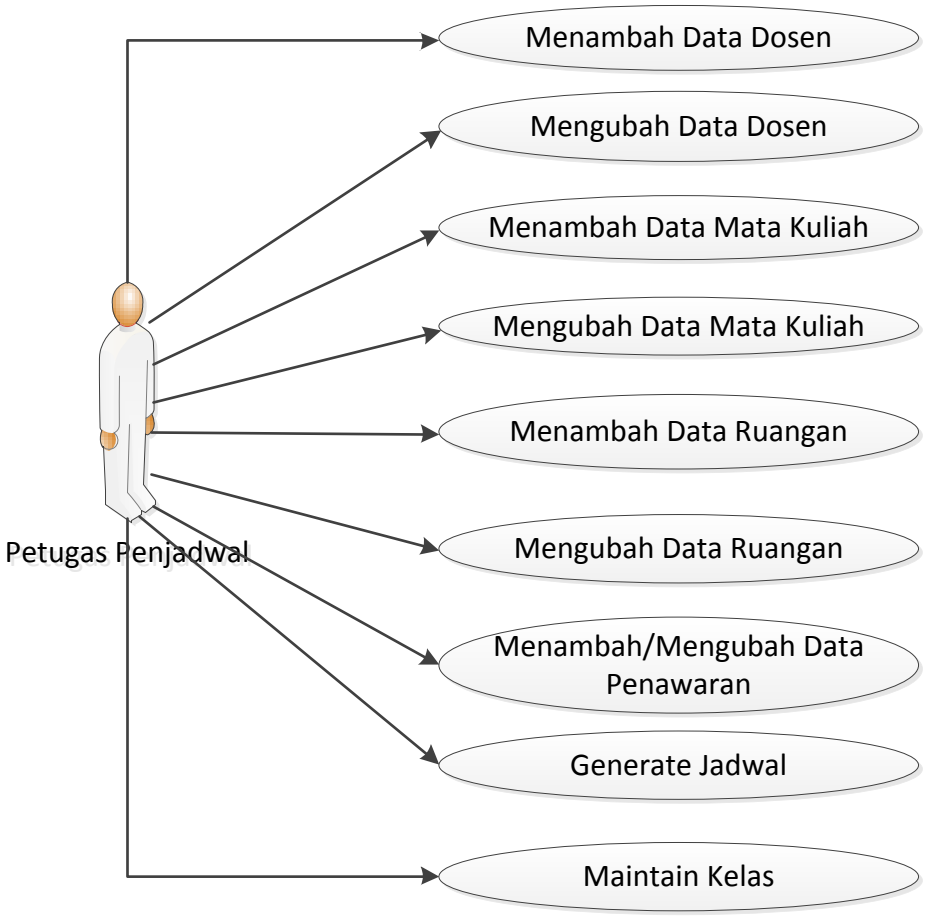
1. Dosen Mengisi data penawaran matakuliah dan menyerahkannya pada petugas penjadwalan kuliah.
2. Petugas memasukkan data jadwal kuliah sesuai dengan data penawaran matakuliah dari dosen.
3. Sistem akan men-*generate* jadwal kuliah sesuai dengan data penjadwalan yang ada.
4. Sistem akan mencetak data penjadwalan kuliah berupa Jadwal Kuliah, BAP (Berita Acara Perkuliahan), dan Tugas Mengajar.

4.2.3.2 Service Time

Dengan sistem penjadwalan yang ditargetkan, proses penjadwalan akan mampu mengoptimalisasi waktu yang dibutuhkan dalam penjadwalan. Hal ini dikarenakan penjadwal tidak perlu lagi menentukan jadwal kuliah untuk setiap dosen, karena target sistem akan langsung men-generate penjadwalan kuliah.

4.2.3.3 Use Case Diagram

Use case diagram (diagram *use case*) adalah diagram yang menyajikan interaksi antara use case dan actor. Dimana actor dapat berupa orang, peralatan atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun. Use case menggambarkan fungsionalitas sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai.



Gambar 4. 5 *Use Case* Diagram untuk Target Sistem

4.2.3.4 Skenario *Use Case*

Skenario *Use Case* digunakan untuk memudahkan dalam menganalisa skenario yang akan kita gunakan pada fase-fase selanjutnya dengan melakukan penilaian terhadap skenario tersebut.

4.2.3.4.1 Skenario Menambah Data Dosen

Tabel 4. 2 Tabel skenario *use case* menambah data dosen

Judul	Menambah data dosen
No. *Use case*	UC-JADWAL-01
Deskripsi *Use Case*	Aktor menambahkan data dosen.
Aktor	Penjadwal
Trigger	Aktor memilih menu untuk menambahkan data dosen.
Kondisi Awal	Aplikasi menampilkan sebuah *form* untuk menambahkan data dosen.
Skenario Normal	
Aksi- Aktor	**Reaksi Sistem**
1. Aktor mengisi *field* kosong pada *form* yang data dari dosen yang akan ditambahkan.	
2. Aktor men-*submit* data dosen yang ingin ditambahkan.	3. Sistem memeriksa apakah *field-field* pada *form* pengisian dosen telah diisi dan sesuai.
	4. Data dosen ditambahkan pada *database*.
Kondisi Akhir Skenario Normal: Data dosen yang baru dimasukkan tampil pada list data dosen	
Skenario Abnormal-1	
Aksi- Aktor	**Reaksi Sistem**

1. Aktor men-*submit* data dengan menekan sebuah *button* tanpa mengisi *field-field* pada form.	2. *Form* pengisian data dosen ditampilkan dan ada pesan kesalahan yang menyatakan *field* harus diisi.
	3. Data dosen tidak ditambahkan pada *database.*
Kondisi Akhir Skenario Abnormal-1: Aplikasi menampilkan *form* pengisian data dosen.	
Catatan	

4.2.3.4.2 Skenario mengubah data dosen

Tabel 4. 3 Skenario *use case* mengubah data dosen

Judul	Mengubah data dosen
No. *Use case*	UC-JADWAL-02
Deskripsi *Use case*	Aktor mengubah data dosen.
Aktor	Penjadwal
Trigger	Aktor menekan salah satu dari data dosen yang akan dirubah.
Kondisi Awal	Aplikasi menampilkan sebuah *form* untuk mengubah data dosen.
Skenario Normal	
Aksi- Aktor	**Reaksi Sistem**
1. Aktor mengubah data pada *field* yang diinginkan.	

2. Aktor men-*submit* data dosen yang ingin ditambahkan.	3. Sistem memeriksa apakah *field-field* pada *form* pengisian dosen telah diisi dan sesuai.
	4. Data dosen dirubah pada *database*.
Kondisi Akhir Skenario Normal: Data dosen yang baru dimasukkan tampil pada list data dosen	
Skenario Abnormal-1	
Aksi- Aktor	**Reaksi Sistem**
1. Aktor men-*submit* data dengan menekan sebuah *button* di mana ada *field-field* pada form yang seharusnya diisi tetapi dikosongkan.	2. *Form* pengisian data dosen ditampilkan dan ada pesan kesalahan yang menyatakan *field* harus diisi.
	3. Data dosen tidak ditambahkan pada *database.*
Kondisi Akhir Skenario Abnormal-1: Aplikasi menampilkan *form* untuk mengubah data dosen.	
Catatan	

4.2.3.4.3 Menambah data mata kuliah

Tabel 4. 4 Skenario *use case* menambah data mata kuliah

Judul	Menambah mata kuliah
No. *Use case*	UC-JADWAL-03
Deskripsi *Use Case*	Aktor menambahkan data mata kuliah.
Aktor	Penjadwal

<table>
<tr><td>Trigger</td><td colspan="2">Aktor memilih menu untuk menambahkan data mata kuliah.</td></tr>
<tr><td>Kondisi Awal</td><td colspan="2">Aplikasi menampilkan sebuah form untuk menambahkan data mata kuliah.</td></tr>
<tr><td colspan="3">Skenario Normal</td></tr>
<tr><td colspan="2">Aksi- Aktor</td><td>Reaksi Sistem</td></tr>
<tr><td colspan="2">1. Aktor mengisi field kosong pada form yang data dari mata kuliah yang akan ditambahkan.</td><td></td></tr>
<tr><td colspan="2">2. Aktor men-submit data mata kuliah yang ingin ditambahkan.</td><td>3. Sistem memeriksa apakah field-field pada form pengisian mata kuliah telah diisi dan sesuai.</td></tr>
<tr><td colspan="2"></td><td>4. Data mata kuliah ditambahkan pada database.</td></tr>
<tr><td colspan="3">Kondisi Akhir Skenario Normal:
Data mata kuliah yang baru dimasukkan tampil pada list data mata kuliah</td></tr>
<tr><td colspan="3">Skenario Abnormal-1</td></tr>
<tr><td colspan="2">Aksi- Aktor</td><td>Reaksi Sistem</td></tr>
<tr><td colspan="2">1. Aktor men-submit data dengan menekan sebuah button tanpa mengisi field-field pada form.</td><td>2. Form pengisian data mata kuliah ditampilkan dan ada pesan kesalahan yang menyatakan field harus diisi.</td></tr>
<tr><td colspan="2"></td><td>3. Data mata kuliah tidak ditambahkan pada database.</td></tr>
<tr><td colspan="3">Kondisi Akhir Skenario Abnormal-1:
Aplikasi menampilkan form pengisian data mata kuliah.</td></tr>
<tr><td colspan="3">Catatan</td></tr>
</table>

4.2.3.4.4 Mengubah data mata kuliah

Tabel 4. 5 Skenario *use case* mengubah data mata kuliah

Judul	Mengubah data mata kuliah
No. *Usecase*	UC-JADWAL-04
Deskripsi *Use Case*	Aktor mengubah data mata kuliah.
Aktor	Penjadwal
Trigger	Aktor menekan salah satu dari data mata kuliah yang akan dirubah.
Kondisi Awal	Aplikasi menampilkan sebuah *form* untuk mengubah data mata kuliah.
Skenario Normal	
Aksi- Aktor	**Reaksi Sistem**
1. Aktor mengubah data pada *field* yang diinginkan.	
2. Aktor men-*submit* data mata kuliah yang ingin ditambahkan.	3. Sistem memeriksa apakah *field-field* pada *form* pengisian mata kuliah telah diisi dan sesuai.
	4. Data mata kuliah dirubah pada *database*.
Kondisi Akhir Skenario Normal: Data mata kuliah yang baru dimasukkan tampil pada list data mata kuliah	
Skenario Abnormal-1	
Aksi- Aktor	**Reaksi Sistem**

1. Aktor men-*submit* data dengan menekan sebuah *button* di mana ada *field-field* pada form yang seharusnya diisi tetapi dikosongkan.	2. *Form* pengisian data mata kuliah ditampilkan dan ada pesan kesalahan yang menyatakan *field* harus diisi.
	3. Data mata kuliah tidak ditambahkan pada *database*.
Kondisi Akhir Skenario Abnormal-1: Aplikasi menampilkan *form* untuk mengubah data mata kuliah.	
Catatan	

4.2.3.4.5 Menambah data ruangan

Tabel 4. 6 Skenario *use case* menambah data ruangan

Judul	Menambah ruangan
No. *Use case*	UC-JADWAL-05
Deskripsi *Use Case*	Aktor menambahkan data ruangan.
Aktor	Penjadwal
Trigger	Aktor memilih menu untuk menambahkan data ruangan.
Kondisi Awal	Aplikasi menampilkan sebuah *form* untuk menambahkan data ruangan.
Skenario Normal	
Aksi- Aktor	**Reaksi Sistem**

1. Aktor mengisi *field* kosong pada *form* yang data dari ruangan yang akan ditambahkan.	
2. Aktor men-*submit* data ruangan yang ingin ditambahkan.	3. Sistem memeriksa apakah *field-field* pada *form* pengisian ruangan telah diisi dan sesuai.
	4. Data ruangan ditambahkan pada *database*.
Kondisi Akhir Skenario Normal: Data ruangan yang baru dimasukkan tampil pada list data ruangan	
Skenario Abnormal-1	
Aksi- Aktor	**Reaksi Sistem**
1. Aktor men-*submit* data dengan menekan sebuah *button* tanpa mengisi *field-field* pada form.	2. *Form* pengisian data ruangan ditampilkan dan ada pesan kesalahan yang menyatakan *field* harus diisi.
	3. Data ruangan tidak ditambahkan pada *database.*
Kondisi Akhir Skenario Abnormal-1: Aplikasi menampilkan *form* pengisian data ruangan.	
Catatan	

4.2.3.4.6 Mengubah data ruangan

Tabel 4. 7 Skenario *use case* mengubah ruangan

Judul	Mengubah data mata ruangan
No. *Use case*	UC-JADWAL-06
Deskripsi *Use Case*	Aktor mengubah data ruangan.
Aktor	Penjadwal
Trigger	Aktor menekan salah satu dari data ruangan yang akan dirubah.
Kondisi Awal	Aplikasi menampilkan sebuah *form* untuk mengubah data ruangan.
Skenario Normal	
Aksi- Aktor	**Reaksi Sistem**
1. Aktor mengubah data pada *field* yang diinginkan.	
2. Aktor men-*submit* data ruangan yang ingin ditambahkan.	3. Sistem memeriksa apakah *field-field* pada *form* pengisian ruangan telah diisi dan sesuai.
	4. Data ruangan dirubah pada *database*.
Kondisi Akhir Skenario Normal: Data ruangan yang baru dimasukkan tampil pada list data ruangan	
Skenario Abnormal-1	
Aksi- Aktor	**Reaksi Sistem**

1. Aktor men-*submit* data dengan menekan sebuah *button* di mana ada *field-field* pada form yang seharusnya diisi tetapi dikosongkan.	2. *Form* pengisian data ruangan ditampilkan dan ada pesan kesalahan yang menyatakan *field* harus diisi.
	3. Data ruangan tidak ditambahkan pada *database.*
Kondisi Akhir Skenario Abnormal-1: Aplikasi menampilkan *form* untuk mengubah data ruangan.	
Catatan	

4.2.3.4.7 Menambah/mengubah data penawaran

Tabel 4. 8 Skenario *use case* menambah/mengubah data penawaran

Judul	Menambah/mengubah penawaran
No. *Use case*	UC-JADWAL-07
Deskripsi *Use case*	Aktor menambahkan/mengubah data penawaran.
Aktor	Penjadwal
Trigger	Aktor memilih menu untuk menambahkan/mengubah data penawaran.
Kondisi Awal	Aplikasi menampilkan sebuah *form* untuk menambahkan/mengubah data penawaran.
Skenario Normal	
Aksi- Aktor	**Reaksi Sistem**

5. Aktor mengisi *field* pada *form* yang data dari penawaran yang akan ditambahkan/dirubah.	
6. Aktor men-*submit* data penawaran yang ingin ditambahkan/dirubah.	7. Sistem memeriksa apakah *field-field* pada *form* pengisian ruangan telah diisi dan sesuai.
	8. Data penawaran ditambahkan/dirubah pada *database*.
Kondisi Akhir Skenario Normal: Data penawaran yang baru dimasukkan tampil pada list data ruangan	
Skenario Abnormal-1	
Aksi- Aktor	**Reaksi Sistem**
Kondisi Akhir Skenario Abnormal-1:	
Catatan	
Use case untuk menambah dan mengubah data penawaran jadwal kuliah disatukan karena *form* yang digunakan merupakan form yang sama.	

4.2.3.4.8 Generate Jadwal

Tabel 4. 9 Skenario *use case Generate* jadwal

Judul	*Generate* Jadwal
No. *Use case*	UC-JADWAL-08
Deskripsi *Use case*	Aktor men-*generate* jadwal kuliah.
Aktor	Penjadwal
Trigger	Aktor menekan tombol Generate Jadwal.
Kondisi Awal	Aplikasi menampilkan list penawaran jadwal kuliah.
Skenario Normal	
Aksi- Aktor	**Reaksi Sistem**
1. Aktor menekan tombol Generate Jadwal	
	2. Sistem men-*generate* jadwal kuliah sesuai dengan data yang ada.
	3. Sistem menyimpan data jadwal kuliah ke *database.*
Kondisi Akhir Skenario Normal: Data jadwal kuliah yang digenerate tersimpan di *database.*	
Skenario Abnormal-1	
Aksi- Aktor	**Reaksi Sistem**

Kondisi Akhir Skenario Abnormal-1:
Catatan

4.2.3.4.9 Maintain Kelas

Tabel 4. 10 Skenario use case Maintain kelas

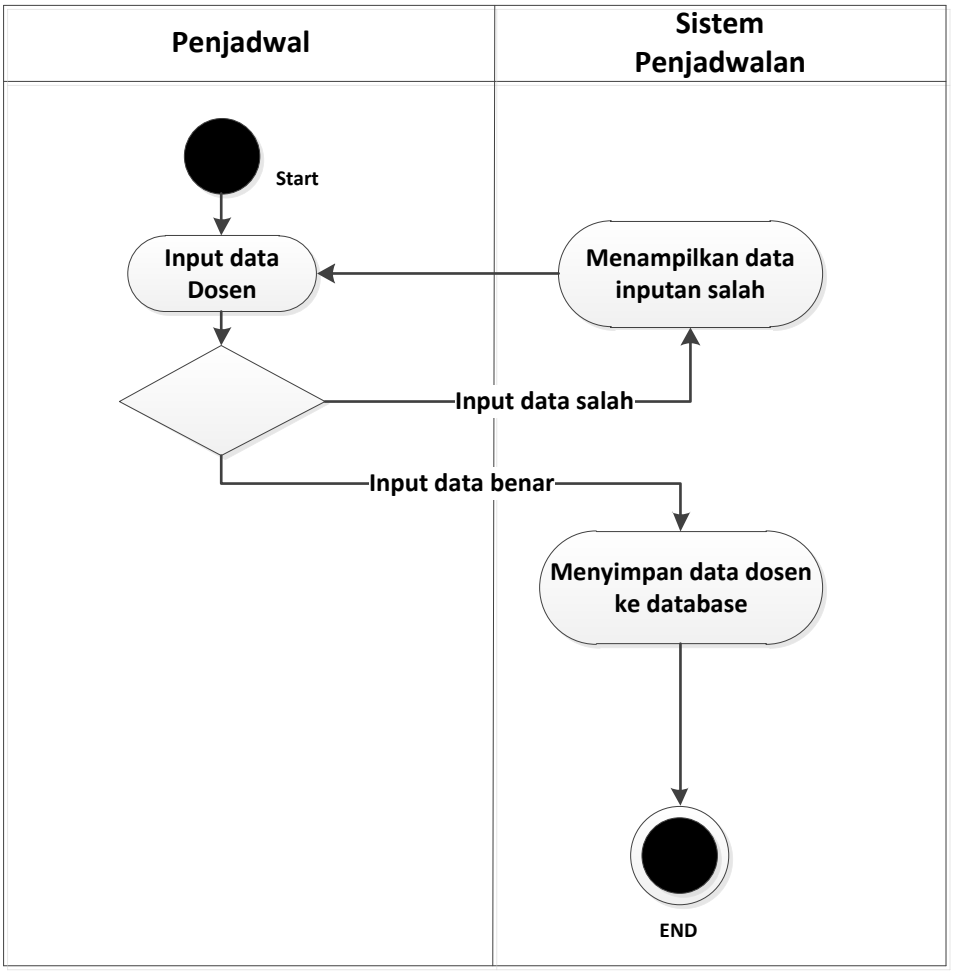
Judul	Maintain Kelas
No. *Use case*	UC-JADWAL-09
Deskripsi *Use case*	Aktor memaintain kelas yang ada.
Aktor	Penjadwal
Trigger	Aktor memilih menu kelas.
Kondisi Awal	Aplikasi menampilkan list kelas yang tersedia.
Skenario Normal	
Aksi- Aktor	**Reaksi Sistem**
1. Aktor menekan simpan	
	2. Sistem mengupdate data kelas pada *database*.
Kondisi Akhir Skenario Normal: Data kelas yang dirubah tersimpan di *database.*	
Skenario Abnormal-1	

Aksi- Aktor	Reaksi Sistem
Kondisi Akhir Skenario Abnormal-1:	
Catatan	

4.2.3.5 *Activity Diagram*

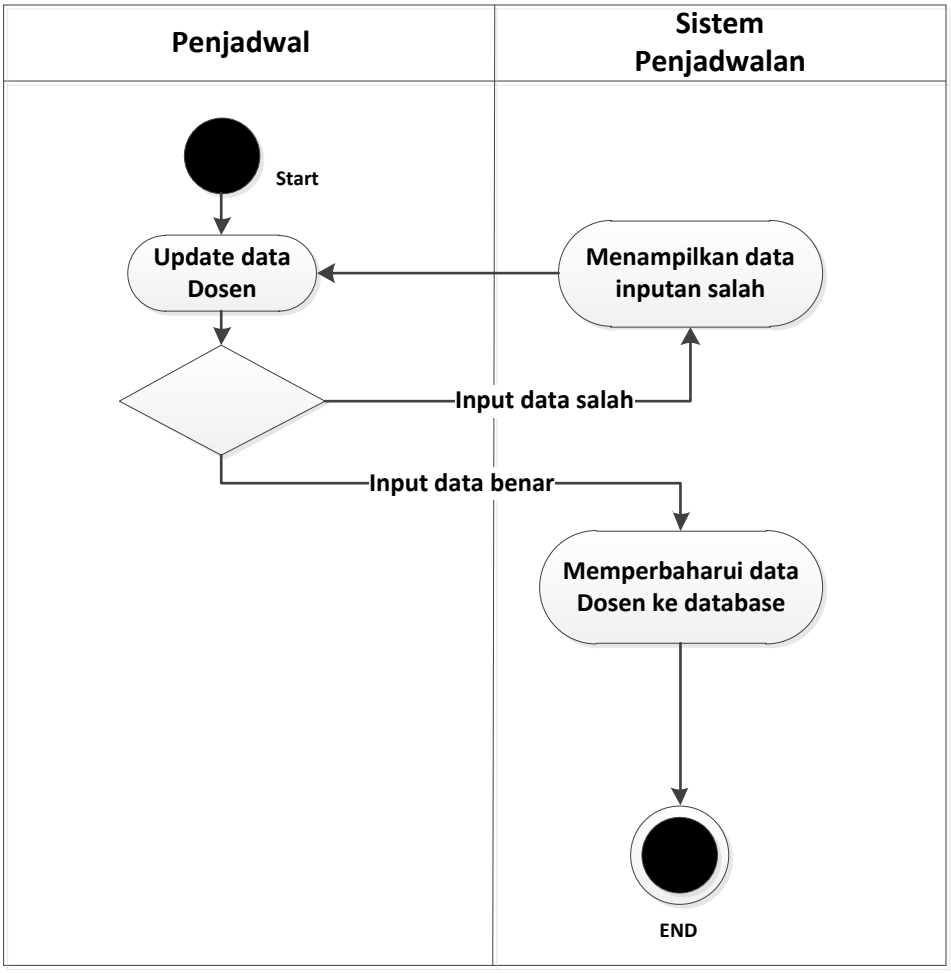
Activity diagram (diagram aktivitas) adalah diagram yang menggambarkan aliran fungsionalitas dari sistem. Pada tahap pemodelan bisnis, diagram aktivitas dapat digunakan untuk menunjukkan aliran kerja bisnis *(business work flow).* Dapat juga digunakan untuk menggambarkan aliran kejadian *(flow of events).*

4.2.3.5.1 Activity Diagram Menambah Data Dosen



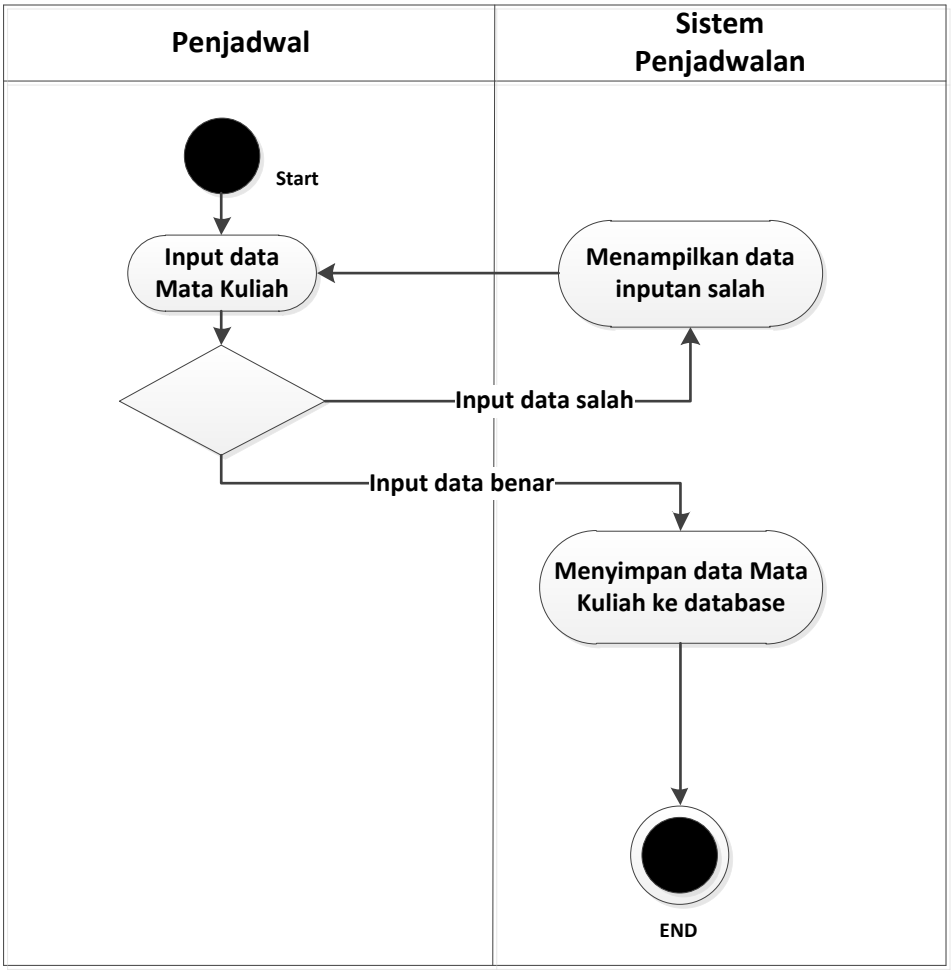
Gambar 4. 6 *Activity Diagram* Menambah Data Dosen

4.2.3.5.2 Activity Diagram Mengubah Data Dosen



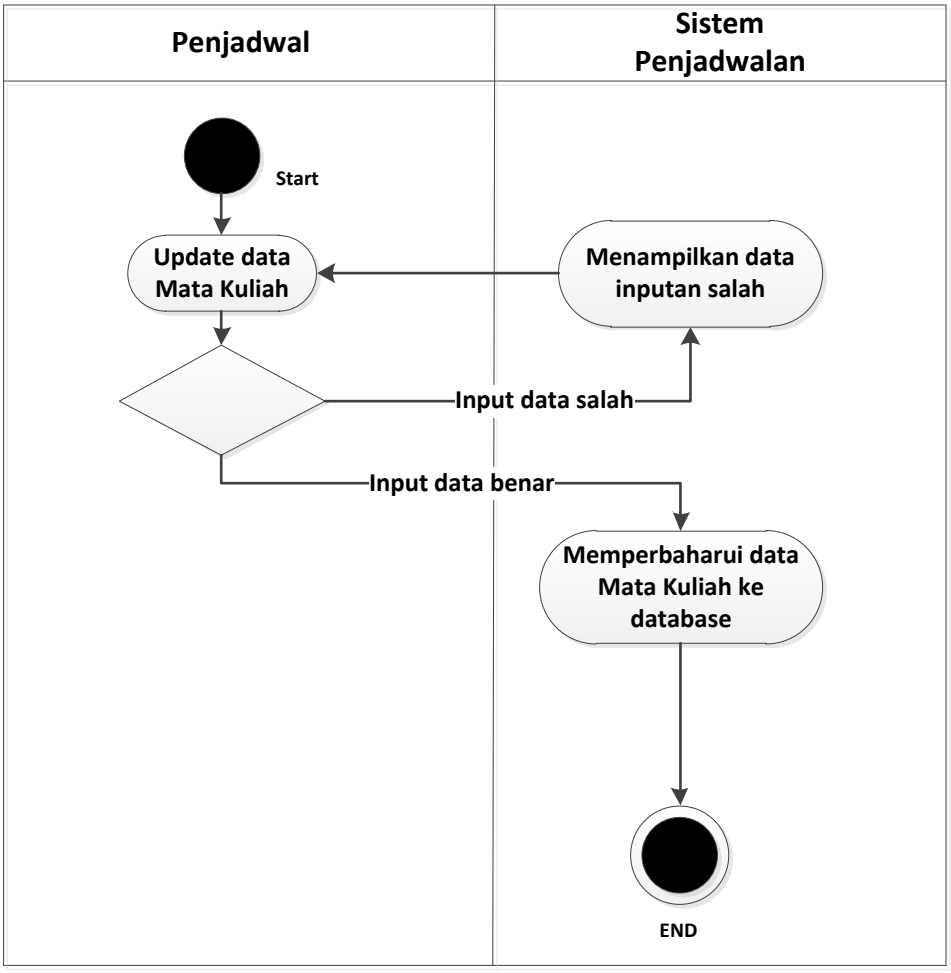
Gambar 4. 7 *Activity Diagram* Mengubah Data Dosen

4.2.3.5.3 Activity Diagram Menambah Data Mata Kuliah



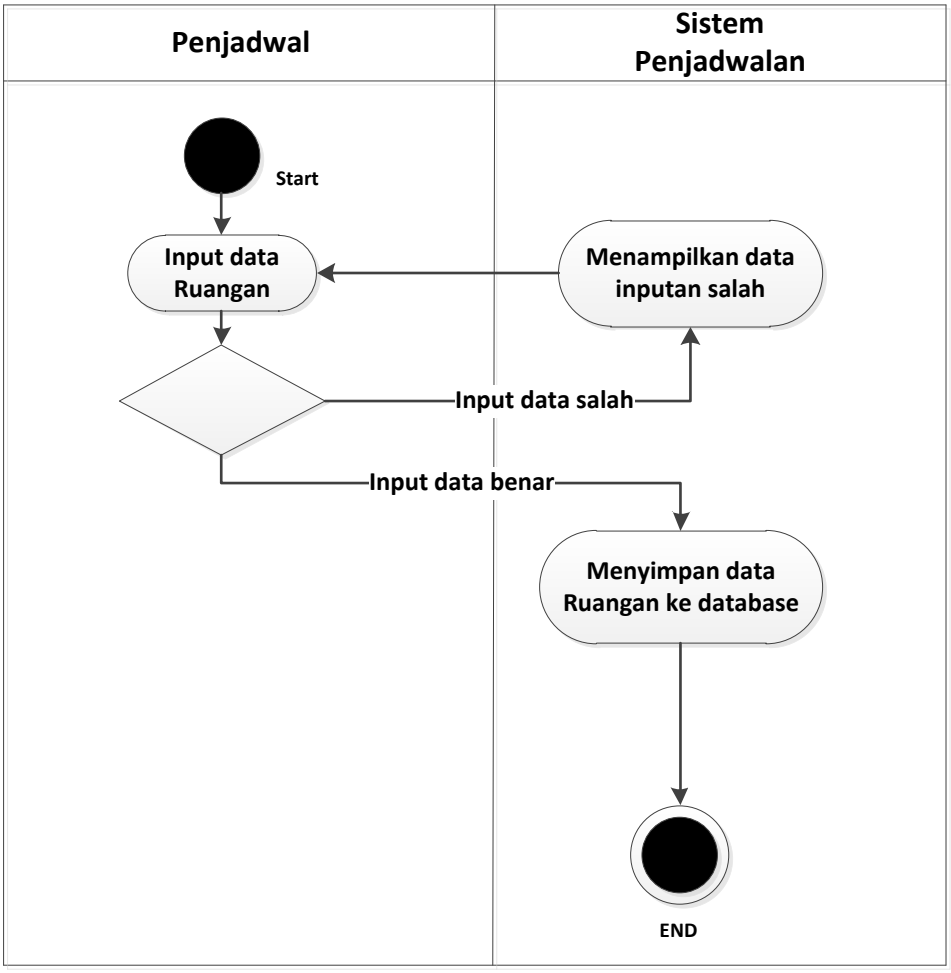
Gambar 4. 8 *Activity Diagram* Menambah Data Mata Kuliah

4.2.3.5.4 Activity Diagram Mengubah Data Mata Kuliah



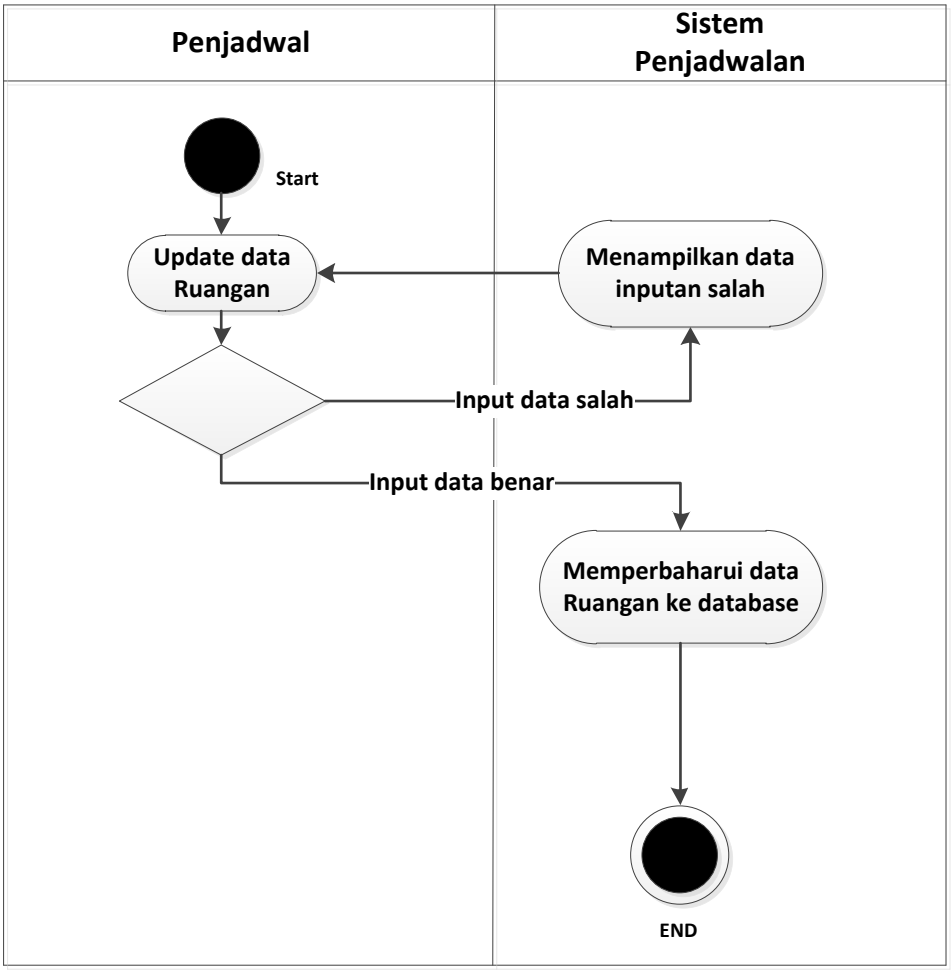
Gambar 4. 9 *Activity Diagram* Mengubah Data Mata Kuliah

4.2.3.5.5 Activity Diagram Menambah Data Ruangan



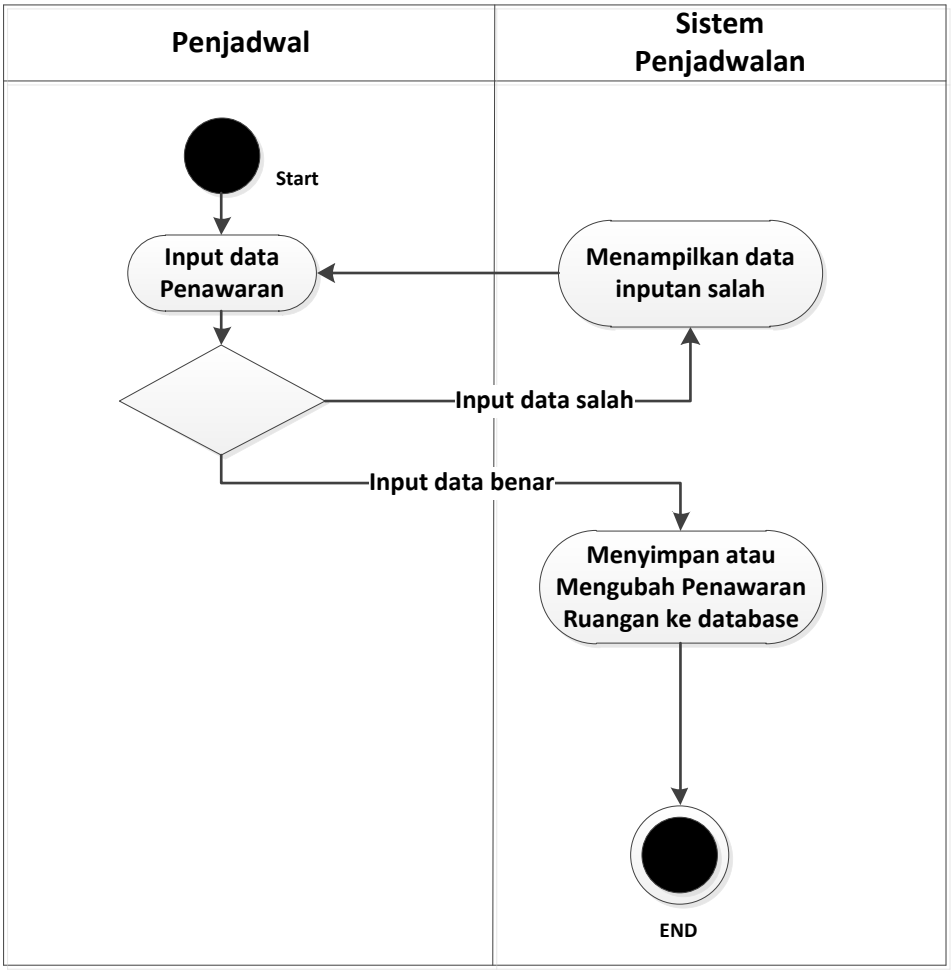
Gambar 4. 10 *Activity Diagram* Menambah Data Ruangan

4.2.3.5.6 Activity Diagram Mengubah Data Ruangan



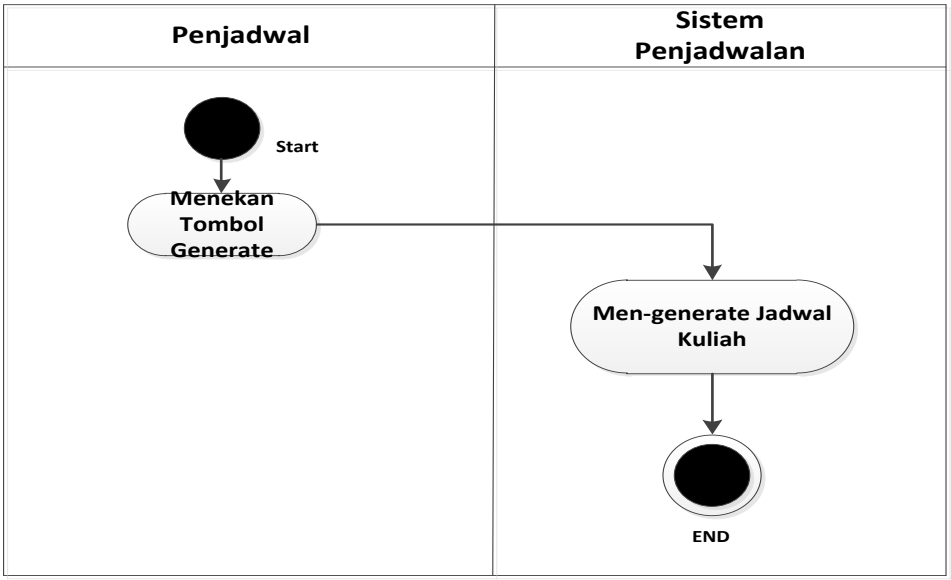
Gambar 4. 11 *Activity Diagram* Mengubah Data Ruangan

4.2.3.5.7 Activity Diagram Menambah/Mengubah Data Penawaran



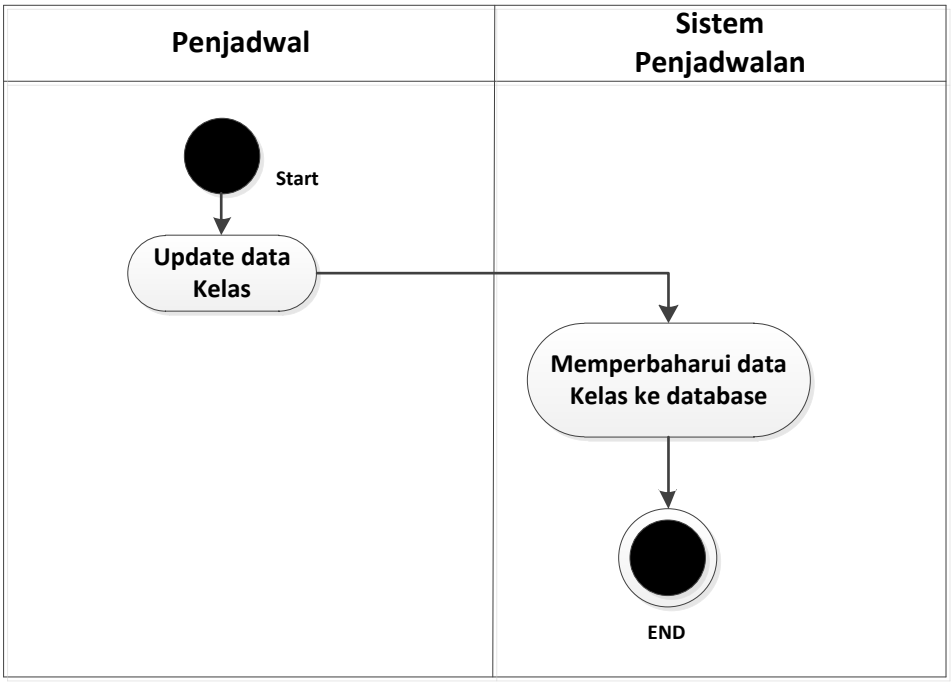
Gambar 4. 12 *Activity Diagram* Menambah atau Mengubah Data Penawaran

4.2.3.5.8 Activity Diagram Generate Jadwal



Gambar 4. 13 *Activity Diagram Generate* Jadwal Kuliah

4.2.3.5.9 Activity Diagram Maintain Kelas

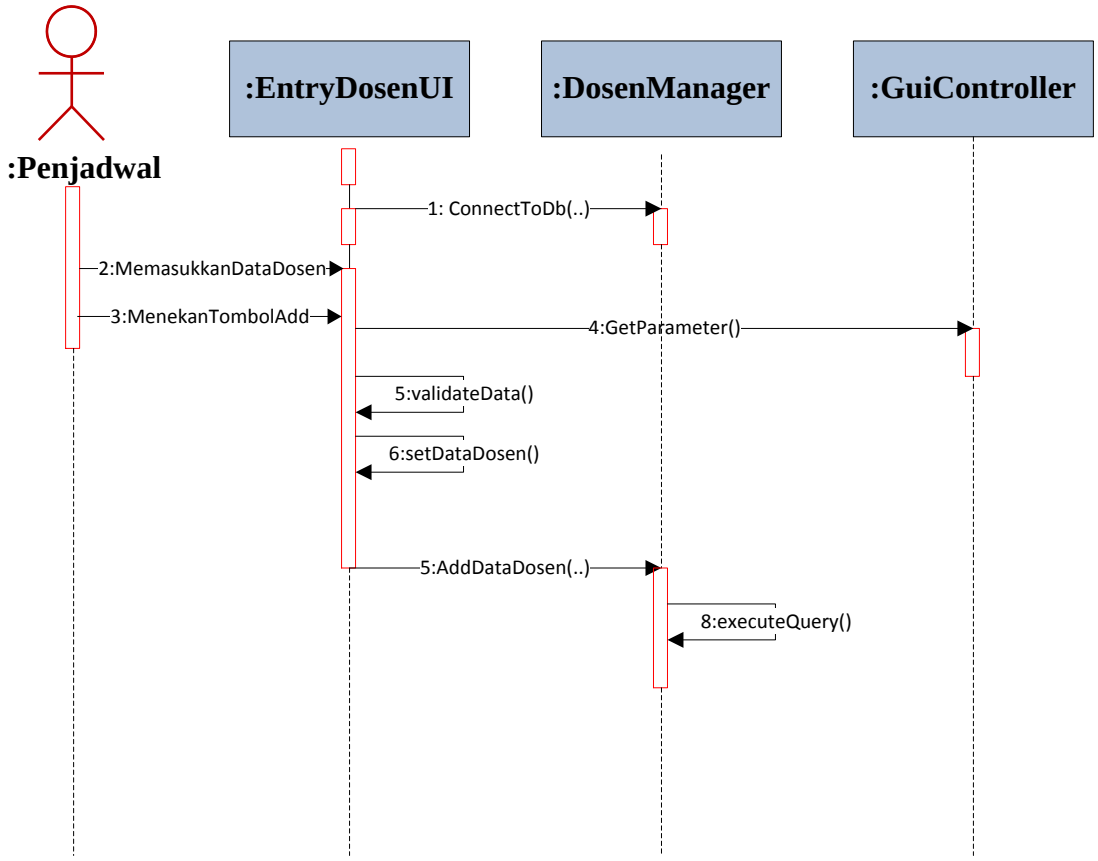


Gambar 4. 14 *Activity Diagram* Maintain Data Kelas

4.2.3.6 Sequence Diagram

Pada setiap *sequence diagram* terdapat aksi aktor yang pertama sekali adalah terhadap *interface*. *Sequence Diagram* digunakan untuk menggambarkan interaksi antar objek dalam waktu yang berurutan. Tetapi pada dasarnya *sequence diagram* digunakan dalam lapisan abstraksi model objek. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antar object, juga interaksi antar objek, dan menunjukkan sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. Komponen utama *sequence diagram* terdiri atas objek yang dituliskan dengan kotak segiempat bernama, pesan diwakili oleh garis dengan tanda panah, dan waktu yang ditunjukkan dengan proses vertikal. Berikut adalah *sequence diagram* yang ada pada penjadwalan.

4.2.3.6.1 Sequence Diagram menambah dosen



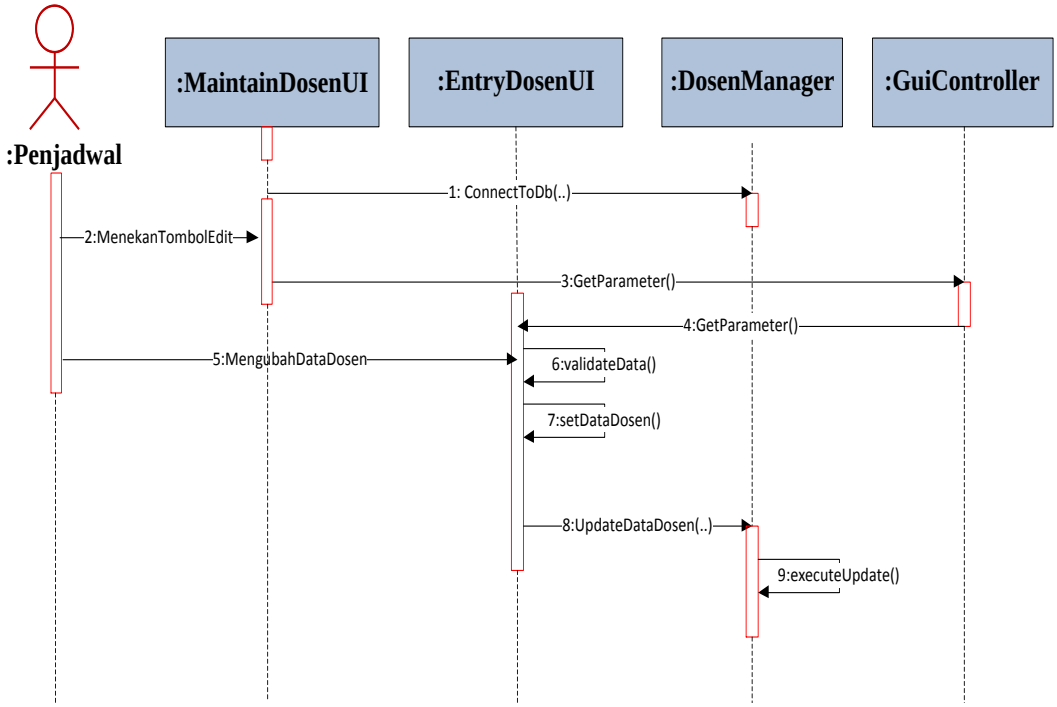
Gambar 4. 15 *Sequence diagram* menambah dosen

Keterangan:

Pada Gambar 4.7 dapat dilihat bahwa terdapat 3 kelas yang saling berinteraksi dalam urutan untuk menambah data dosen, yaitu :

1. GUIDosen
2. DosenManager
3. GuiControler

4.2.3.6.2 Sequence Diagram mengubah dosen



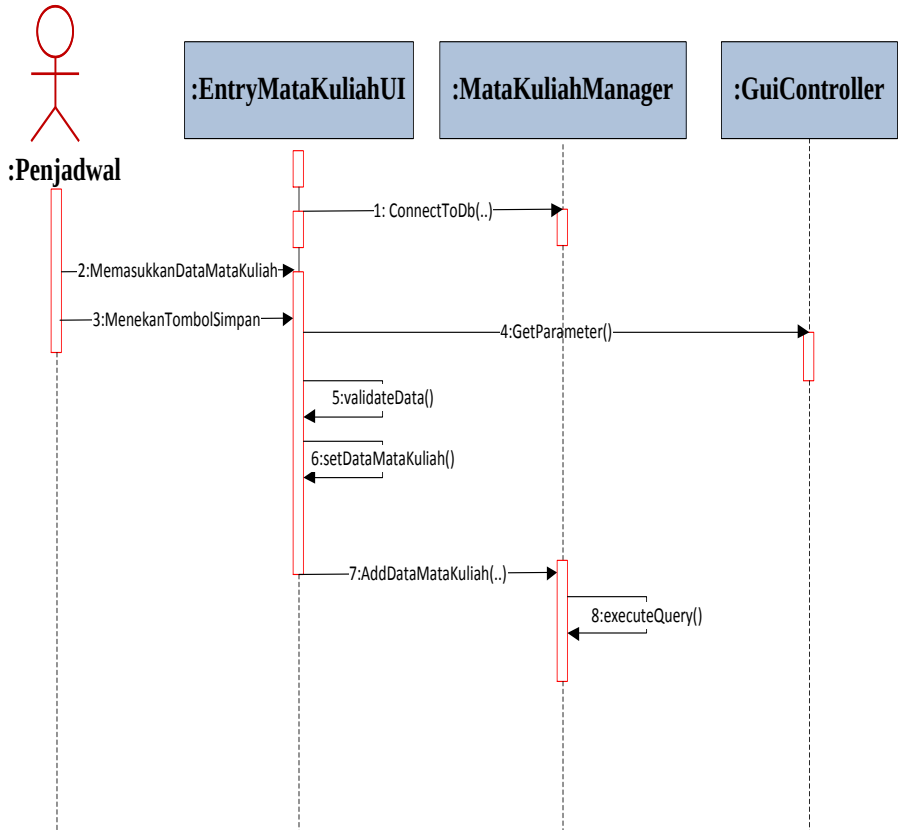
Gambar 4. 16 Sequence diagram mengubah dosen

Keterangan:

Pada Gambar 4.8 dapat dilihat bahwa terdapat 3 kelas yang saling berinteraksi dalam urutan untuk mengubah data dosen, yaitu :

1. MaintainDosenUI
2. EntryDosenUI
3. DosenManager
4. GuiControler

4.2.3.6.3 Sequence Diagram menambah mata kuliah



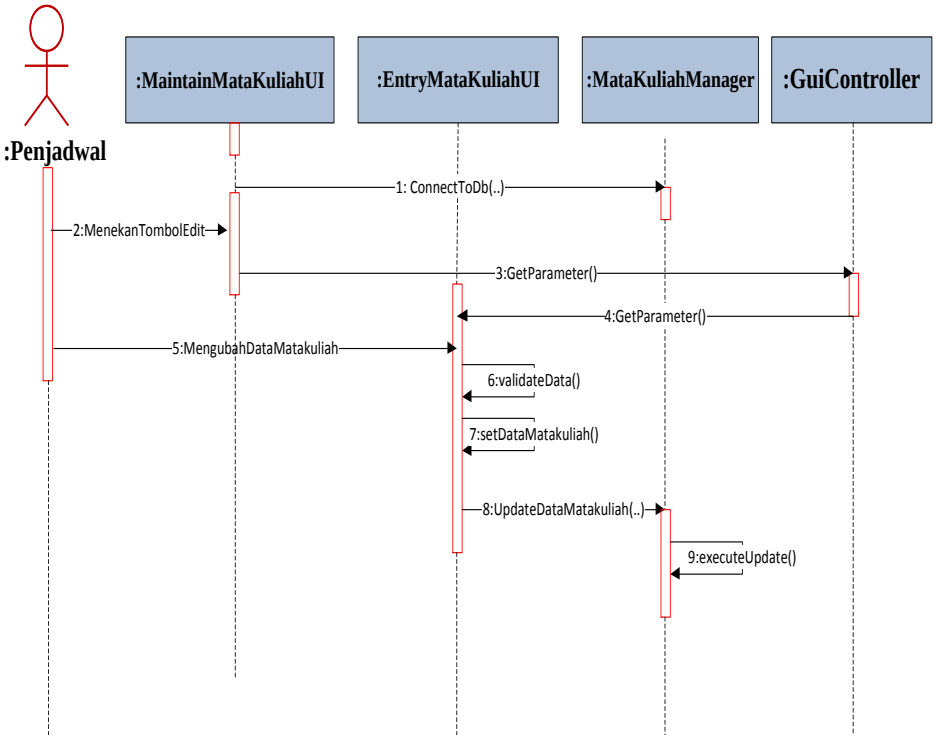
Gambar 4. 17 sequence diagram menambah mata kuliah

Keterangan:

Pada Gambar 4.9 dapat dilihat bahwa terdapat 3 kelas yang saling berinteraksi dalam urutan untuk menambah data mata kuliah, yaitu :

1. EntryMatakuliahUI
2. MataKuliahManager
3. GuiControler

4.2.3.6.4 Sequence Diagram mengubah mata kuliah



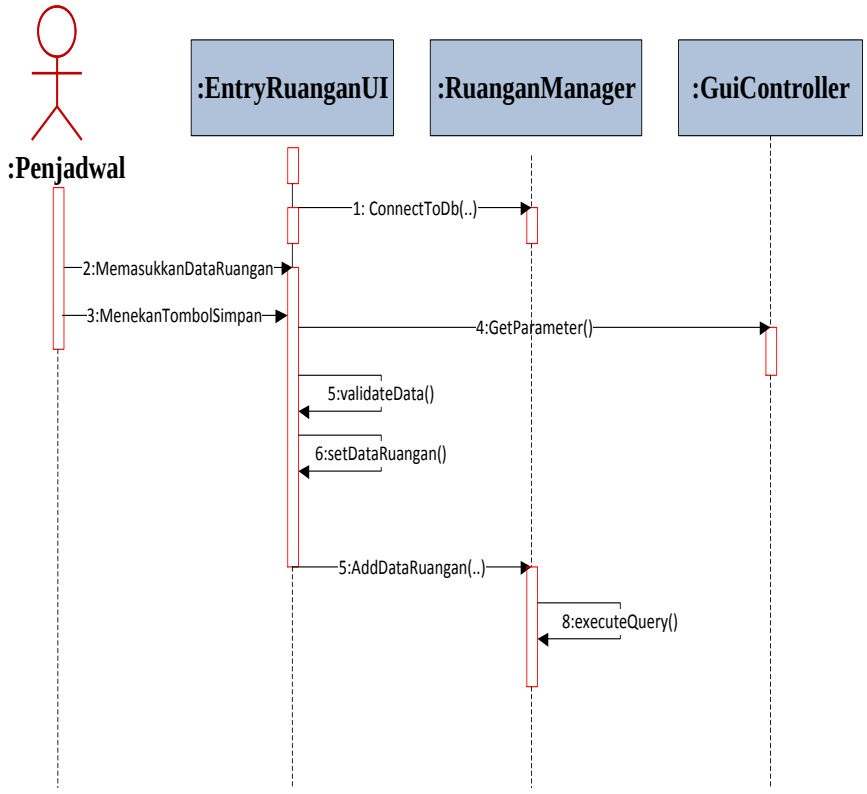
Gambar 4. 18 sequence diagram mengubah mata kuliah

Keterangan:

Pada Gambar 4.10 dapat dilihat bahwa terdapat 4 kelas yang saling berinteraksi dalam urutan untuk mengubah data mata kuliah, yaitu :

1. MaintainMatakuliahUI
2. EntryMatakuliahUI
3. MataKuliahManager
4. GuiControler

4.2.3.6.5 Sequence Diagram menambah ruangan



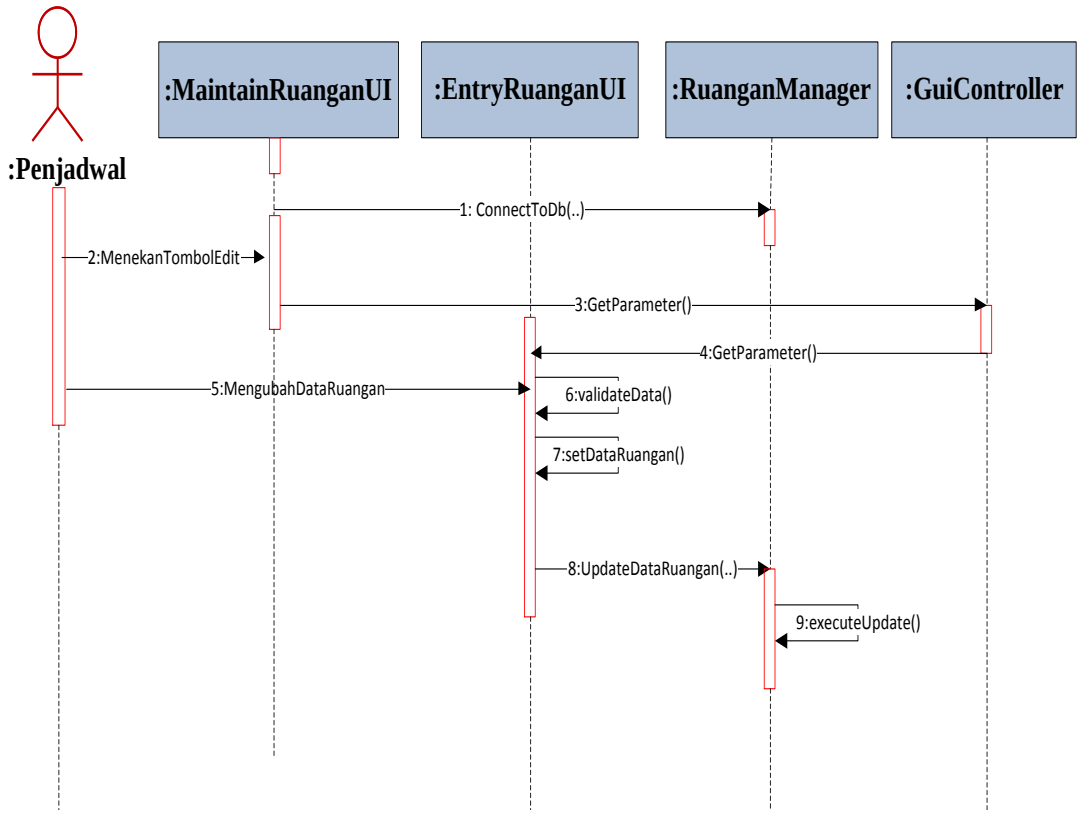
Gambar 4. 19 *sequence diagram* menambah ruangan

Keterangan:

Pada Gambar 4.11 dapat dilihat bahwa terdapat 3 kelas yang saling berinteraksi dalam urutan untuk menambah data ruangan, yaitu :

1. EntryRuanganUI
2. RuanganManager
3. GuiControler

4.2.3.6.6 Sequence Diagram mengubah ruangan



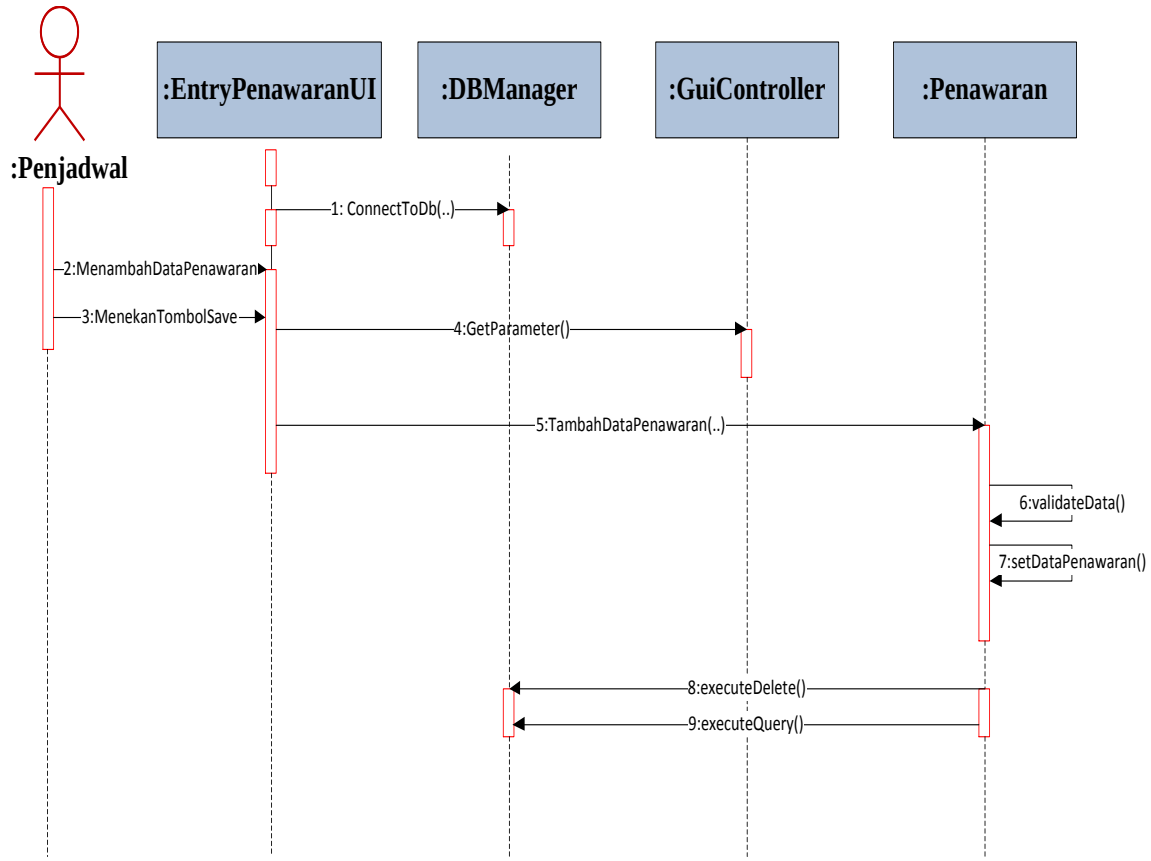
Gambar 4. 20 *sequence diagram* mengubah ruangan

Keterangan:

Pada Gambar 4.12 dapat dilihat bahwa terdapat 4 kelas yang saling berinteraksi dalam urutan untuk mengubah data ruangan, yaitu :

1. MaintainRuanganUI
2. EntryRuanganUI
3. RuanganManager
4. GuiControler

4.2.3.6.7 Sequence Diagram menambah/mengubah penawaran



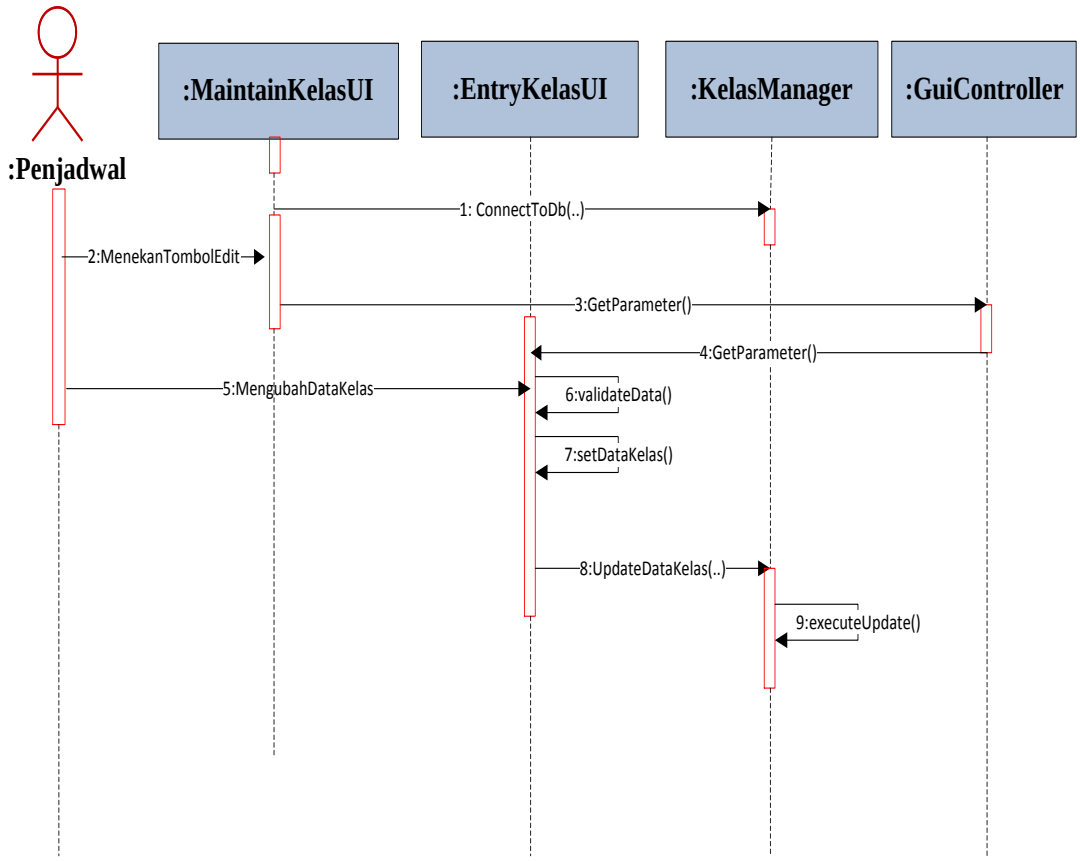
Gambar 4. 21 sequence diagram menambah/mengubah ruangan

Keterangan:

Pada Gambar 4.13 dapat dilihat bahwa terdapat 4 kelas yang saling berinteraksi dalam urutan untuk menambah/mengubah data ruangan, yaitu :

1. EntryPenawaranUI
2. DBManager
3. GuiControler
4. Penawaran

4.2.3.6.8 Sequence Diagram Maintain Kelas



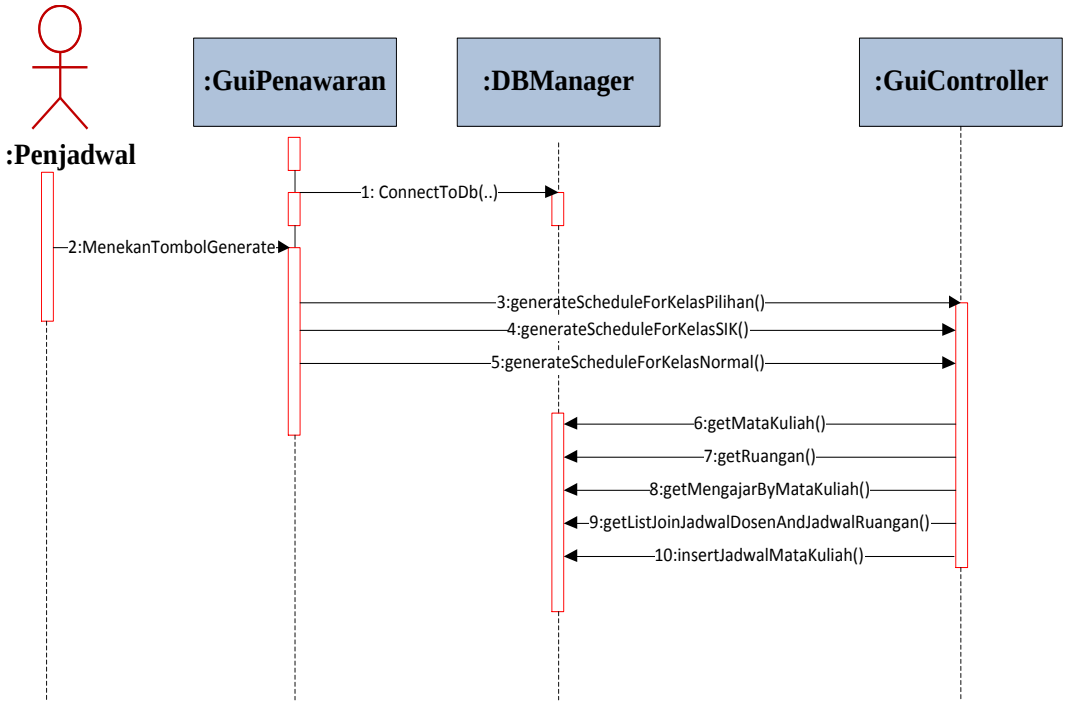
Gambar 4. 22 *sequence diagram* memaintain Kelas

Keterangan:

Pada Gambar 4.13 dapat dilihat bahwa terdapat 4 kelas yang saling berinteraksi dalam urutan untuk menambah/mengubah data ruangan, yaitu:

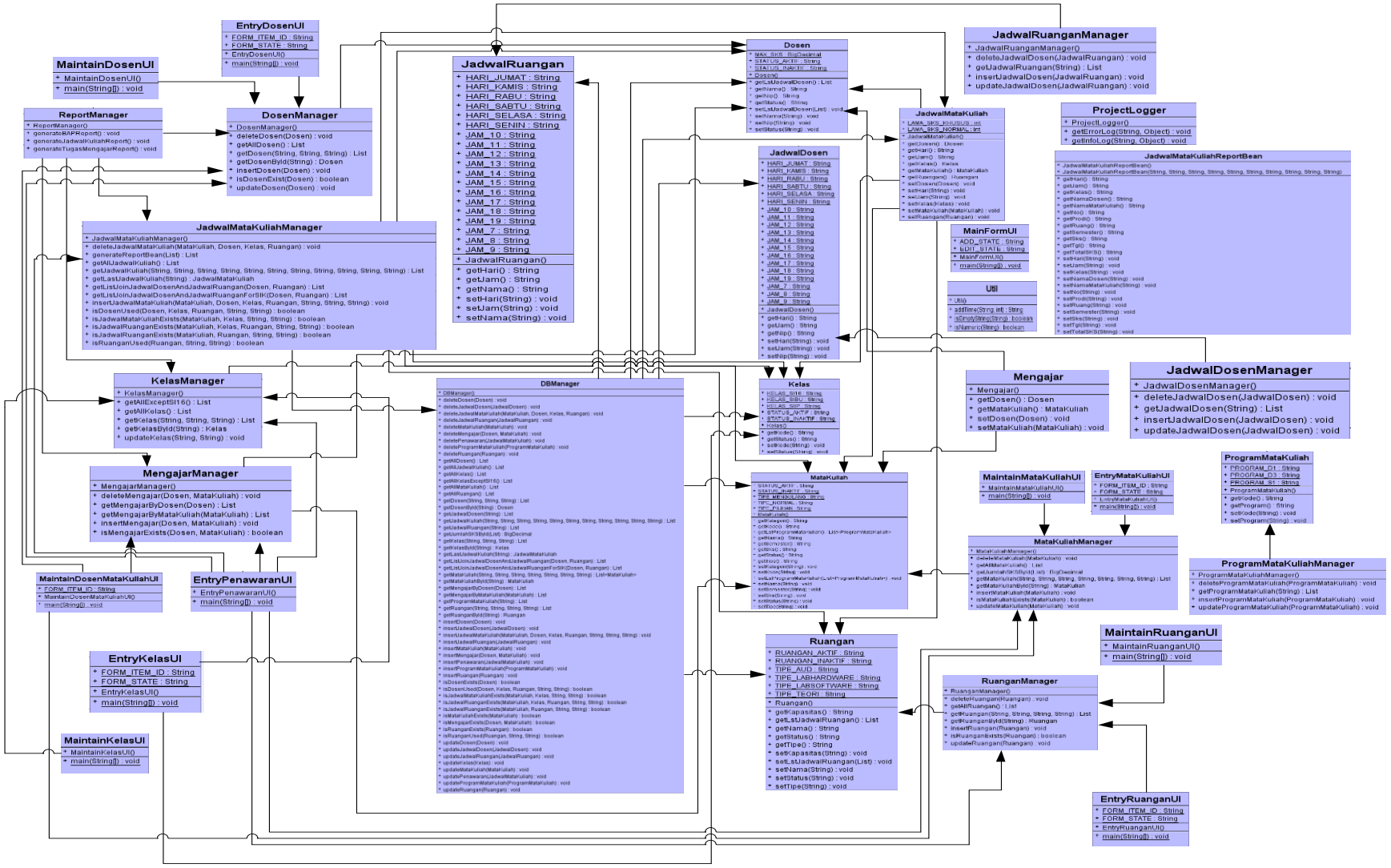
1. MaintainRuanganUI
2. EntryRuanganUI
3. RuanganManager
4. GuiControler

4.2.3.6.9 Sequence Diagram generate jadwal



4.2.3.7 Class Diagram

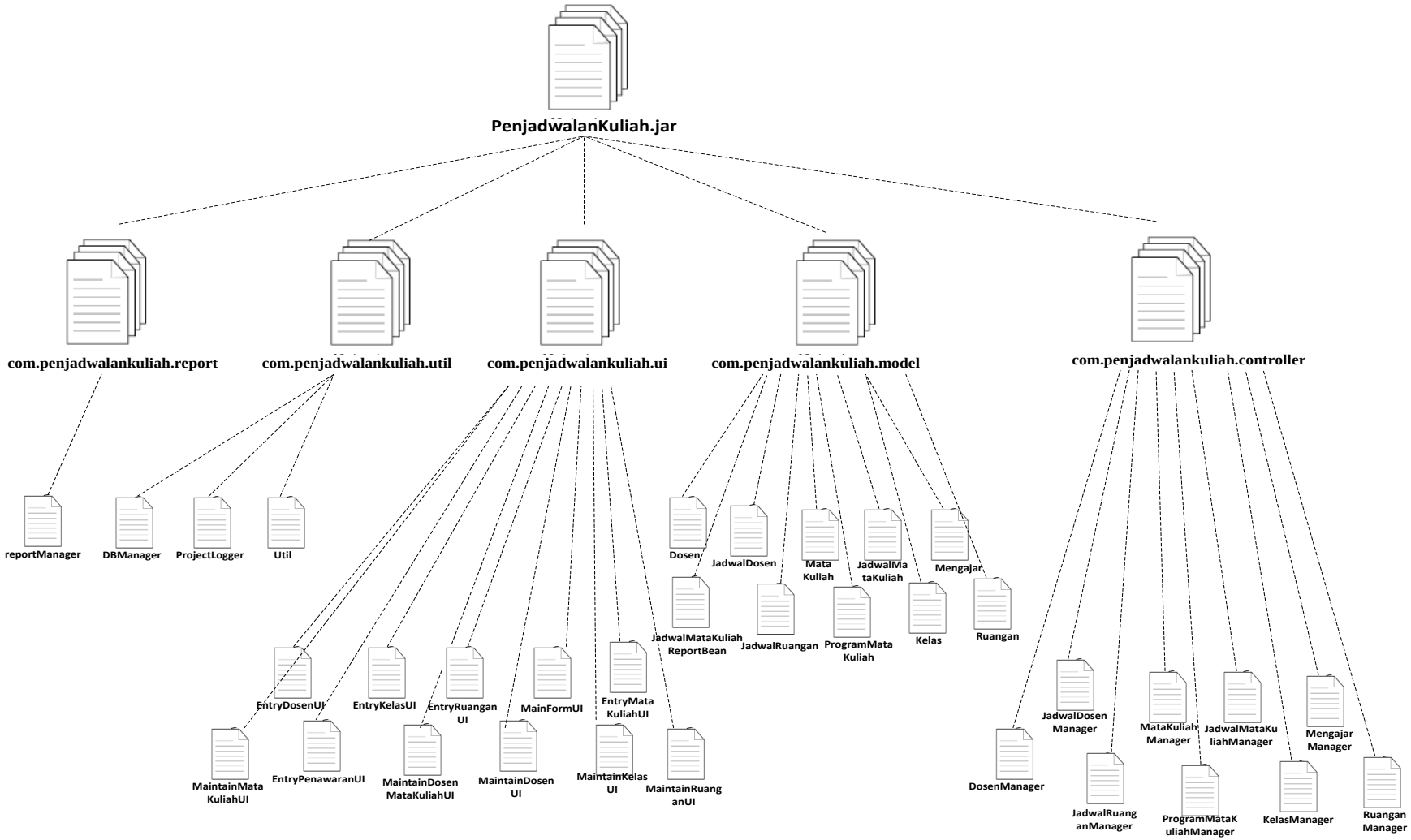
Menggambarkan struktur statis *class* di dalam sistem. *Class* merepresentasikan sesuatu yang ditangani oleh sistem. Dengan melihat karakteristik sistem pemasaran produk dari bagian penjualan beserta proses-proses yang terjadi, maka dapat dibuat *Class Diagram* Berikut *Class Diagram* Sistem Penjadwalan yang diusulkan pada Program Studi Sitem Informasi Unikom.



Gambar 4. 23 Class Diagram

4.2.3.8 Component Diagram

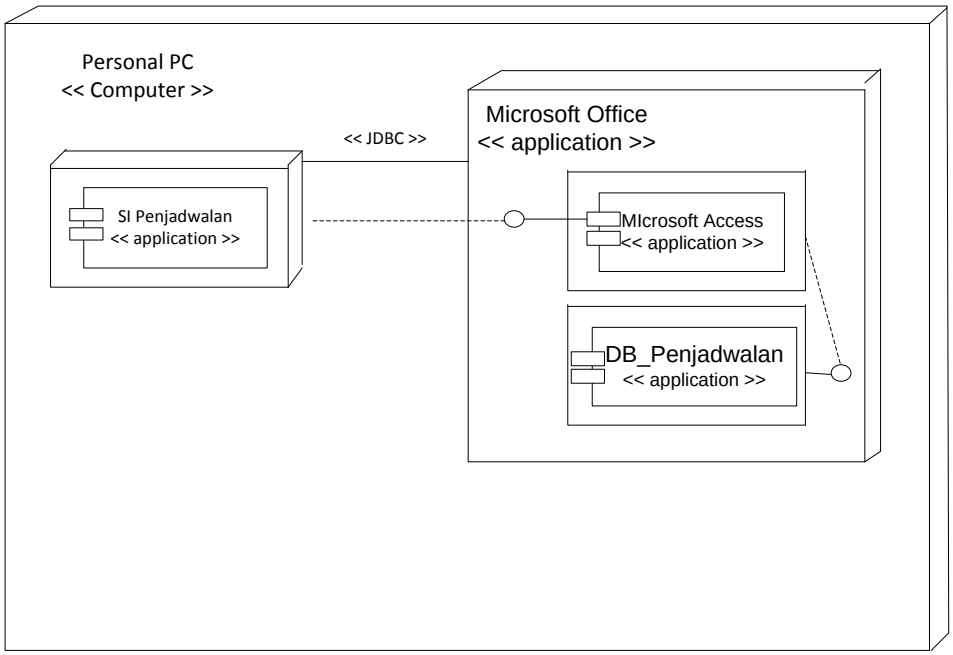
Component diagram menggambarkan struktur dan hubungan antar komponen perangkat lunak, termasuk ketergantungan (*dependency*) di antaranya. *Component* piranti lunak adalah modul berisi *code*, baik *berisi source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time, link time,* maupun *run time.* Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain. Berikut ini adalah *component diagram* yang dibutuhkan :



Gambar 4. 24 Component Diagram

4.2.3.9 Deployment Diagram

Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan. Diagram ini memuat simpul-simpul beserta komponen-komponen yang ada didalamnya. *Deployment diagram* berhubungan dengan diagram komponen dimana *deployment diagram* memuat satu atau lebih komponen-komponen.



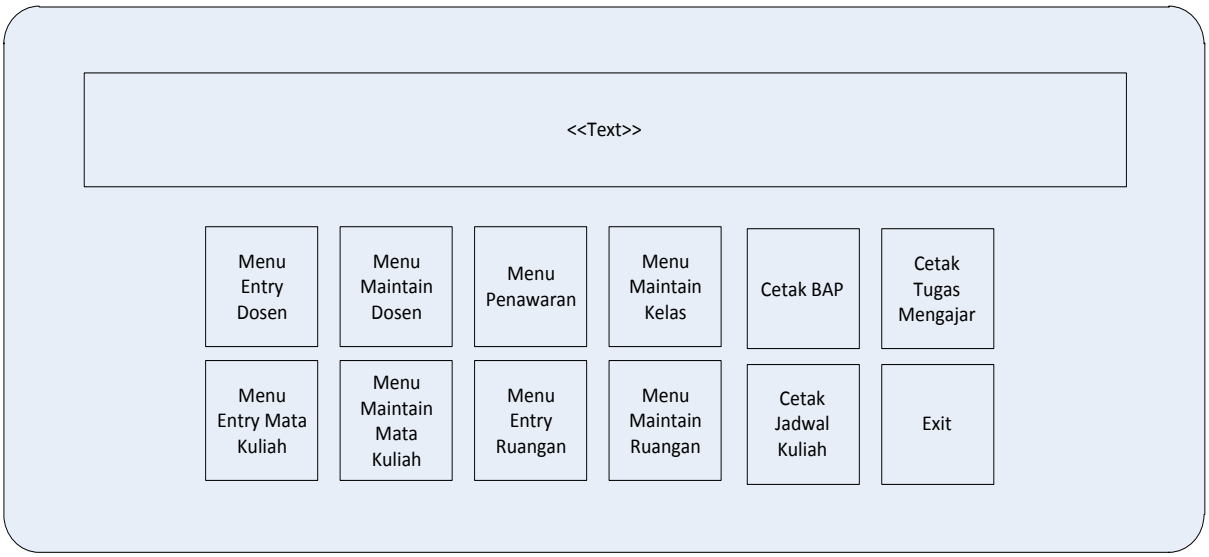
Gambar 4. 25 *Deployment diagram* sistem penjadwalan target

4.2.4 Perancangan Antar Muka

Perancangan antar muka merupakan perancangan yang dibuat sebelum program aplikasi dibuat, perancangan antar muka pada sistem informasi Penjadwalan untuk Program Studi Sistem Informasi UNIKOM adalah sebagai berikut:

4.2.4.1 Perancangan Struktur Menu

Perancangan menu dibuat sebagai alat antar muka dengan pengguna untuk memudahkan pengoperasian perangkat lunak. Berikut rancangan menu perangkat lunak ini :



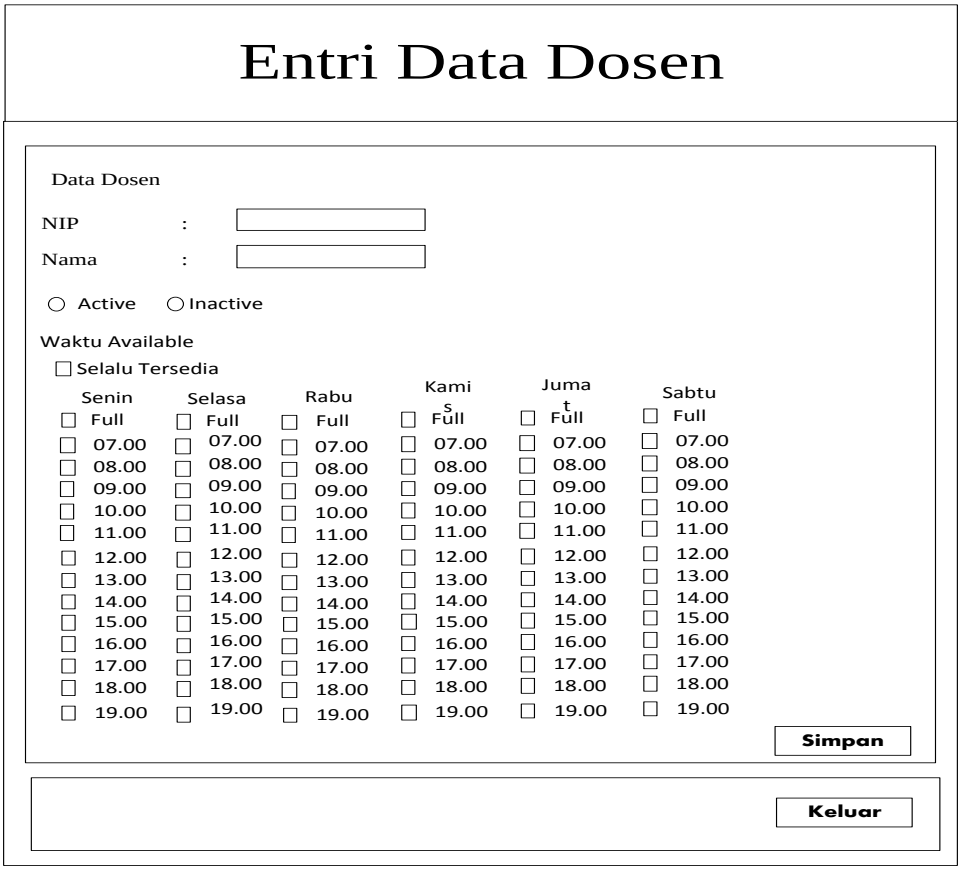
Gambar 4. 26 Perancangan struktur menu

4.2.4.2 Perancangan Input

Perancangan input merupakan perancangan tampilan yang akan digunakan guna memasukkan data pada sistem untuk kemudian diproses. Dalam perancangan input ini, data yang dimasukkan akan mempengaruhi hasil yang ditampilkan. Adapun perancangan-perancangan input yang ada dalam perancangan ini adalah:

1. Rancangan Entry data dosen

   Rancangan tampilan data dosen berfungsi untuk memasukkan data dosen pada sistem. Berikut rancangan data tampilan data dosen yang dibuat:



Gambar 4. 27 Desain tampilan untuk menambah data dosen

Keterangan:

a. *Textbox* NIP dan nama dosen digunakan untuk mengisi data dari dosen.

b. *Radio button* active dan inactive berfungsi untuk menentukan status dosen aktif mengajar atau tidak.

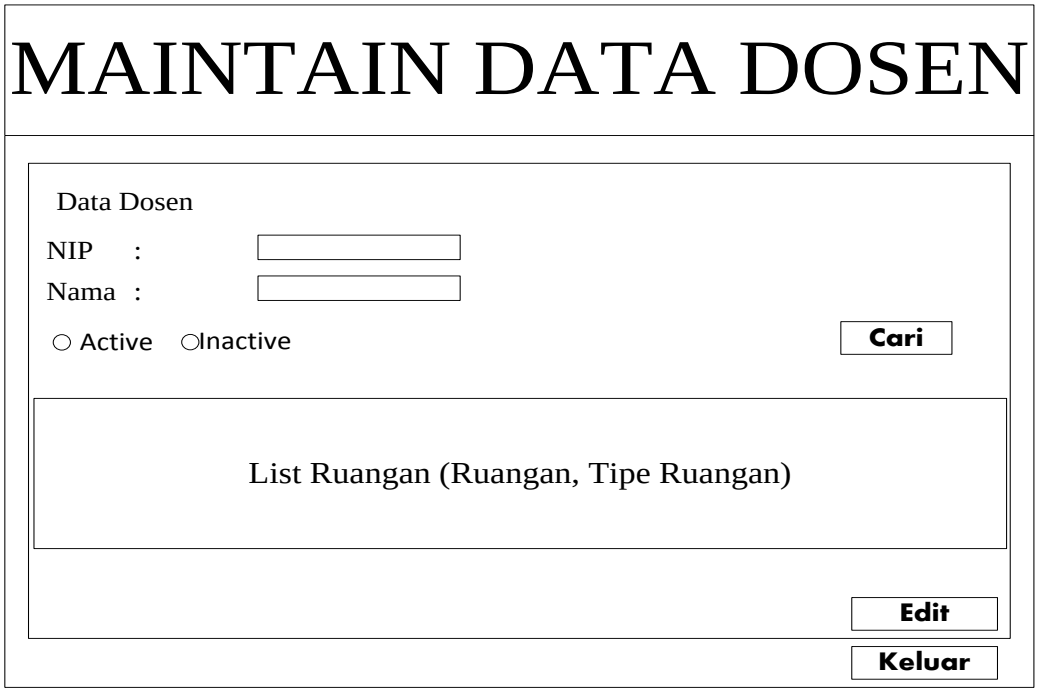
c. *Checkbox* untuk waktu *available* menentukan kapan ketersediaan waktu dosen untuk mengajar. Memilih selalu tersedia akan mencentang semua waktu, memilih full akan mencentang semua waktu yang tersedia pada hari tersebut.

d. Tombol simpan berfungsi untuk memasukkan data dosen yang baru.

e. Tombol keluar digunakan untuk kembali ke halaman utama.

2. Rancangan Maintain data Dosen

Rancangan tampilan maintain data dosen berfungsi untuk mencari data dosen pada sistem untuk kemudian di edit. Berikut rancangan data tampilan data dosen yang dibuat:



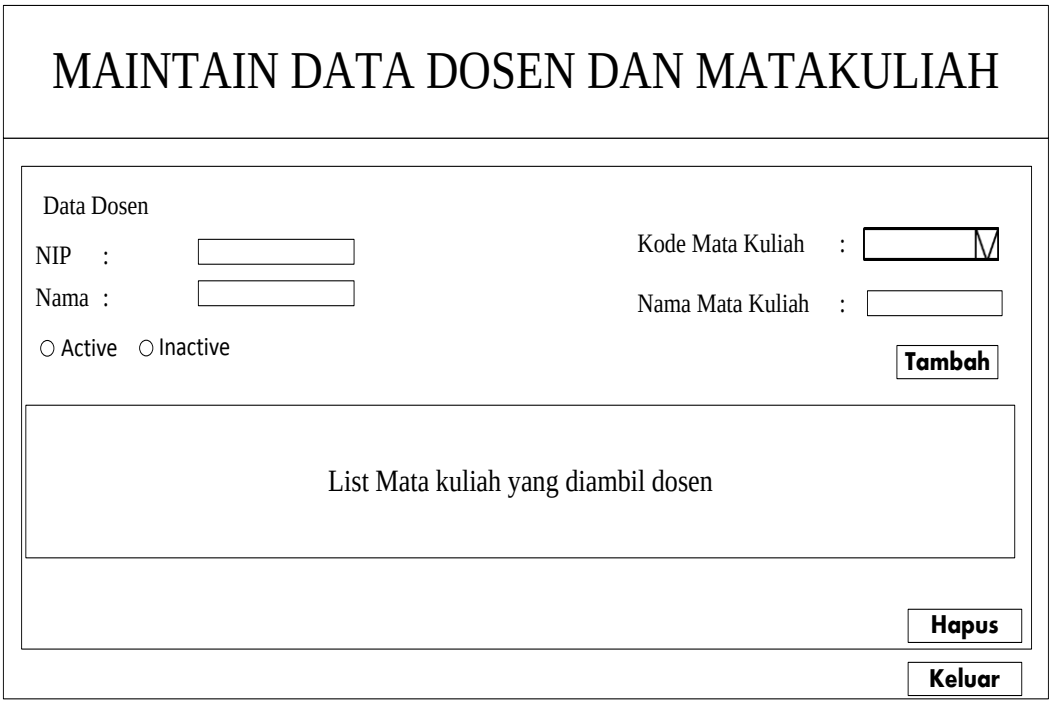
Gambar 4. 28 Desain tampilan untuk maintain data dosen 1

Keterangan:

a. *Textbox* NIP dan nama dosen digunakan untuk mengisi data dari dosen yang ingin dicari.

b. *Radio button* active dan inactive berfungsi untuk menentukan status dosen aktif mengajar atau tidak yang ingin dicari.
c. Tombol Cari digunakan untuk mencari dosen berdasarkan data yang diisi.
d. Tombol keluar digunakan untuk kembali ke halaman utama.

Setelah data dosen dipilih kemudian data dari dosen tersebut ingin dirubah maka akan dimunculkan tampilan berikut.



Gambar 4. 29 Desain tampilan untuk maintain data dosen 2

Keterangan:

a. Tombol Tambah berfungsi untuk menambahkan mata kuliah yang diajarkan oleh dosen yang bersangkutan. *Combo box* mata kuliah berisikan list matakuliah yang ada. Pada saat *combo box* dipilih, *textbox* nama matakuliah akan terisi secara otomatis berdasarkan matakuliah yang dipilih.

b. *Textbox* NIP dan nama dosen akan terisi otomatis sesuai dengan data dosen yang dipilih.

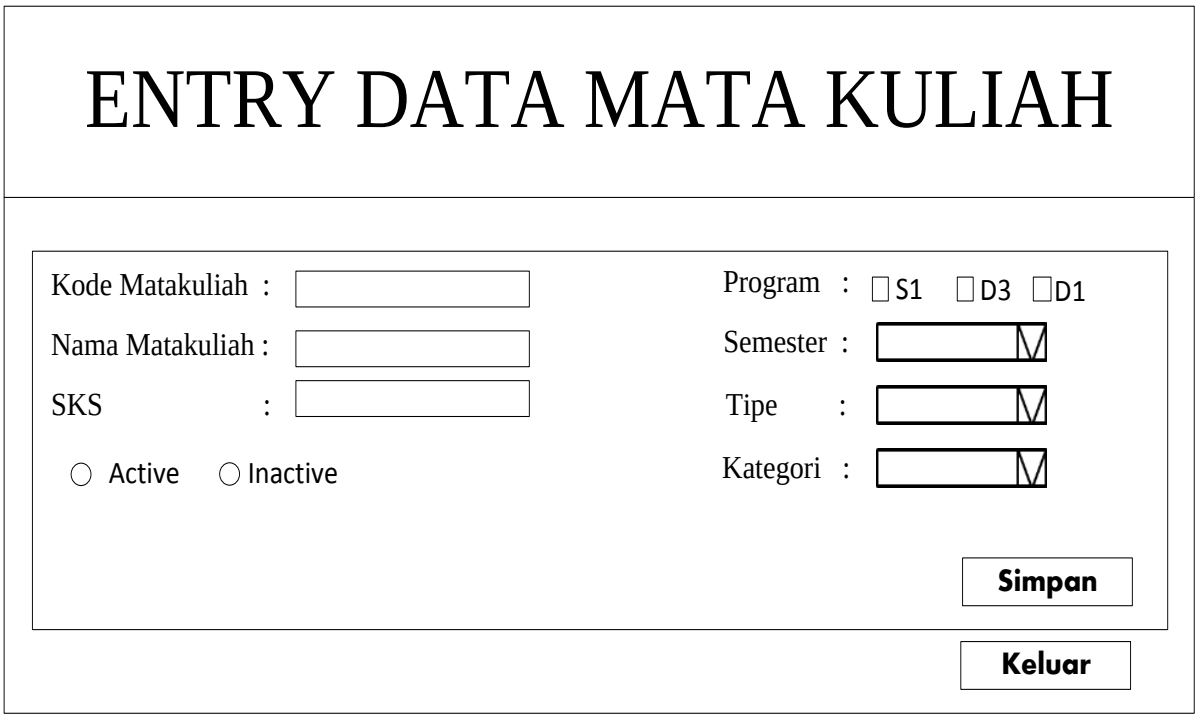
c. *Radio button* active dan inactive akan otomatis terpilih sesuai dengan data dosen yang dipilih.

d. Tombol hapus berfungsi untuk menghapus data mata kuliah yang diajarkan oleh dosen yang bersangkutan.

e. Tombol keluar digunakan untuk kembali ke halaman utama.

3. Rancangan Entry Matakuliah

Rancangan tampilan Entry data mata kuliah berfungsi untuk menambahkan data mata kuliah pada sistem. Berikut rancangan data tampilan entry data mata kuliah yang dibuat:



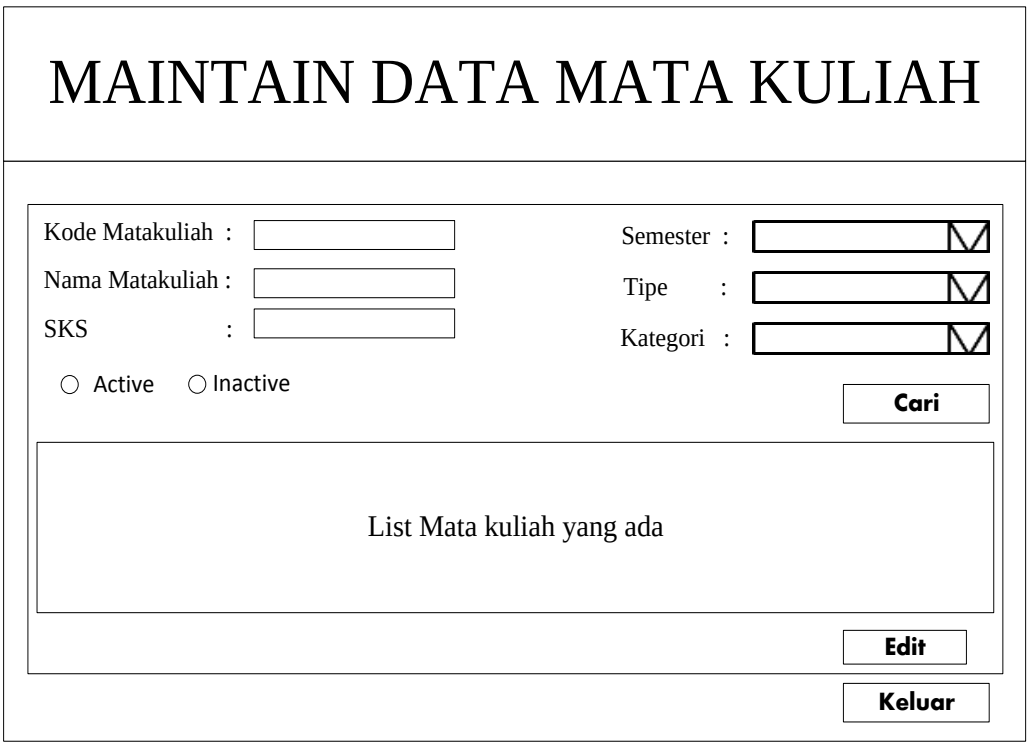
Gambar 4. 30 Desain tampilan untuk entry data mata kuliah

Keterangan:

a. *Textbox* Kode Matakuliah, nama kuliah, dan sks digunakan untuk mengisi data dari mata kuliah.
b. Radio button active dan inactive berfungsi untuk menentukan status mata kuliah aktif diajarkan atau tidak.
c. *Checkbox* program digunakan untuk menentukan matakuliah tersebut diajarkan pada program apa saja.
d. *Combobox* semester menentukan matakuliah tersebut diajarkan pada semester berapa (I, II, III, IV, V, VI, VII, VIII)
e. *Combobox* Tipe digunakan untuk menentukan tipe dari mata kuliah tersebut (Teori, Lab software, Lab Hardware, AUD).
f. *Combobox* Kategori digunakan untuk menentukan kategori dari mata kuliah (normal, pilihan, mengulang)
g. Tombol Simpan berfungsi untuk memasukkan data mata kuliah yang baru.
h. Tombol keluar digunakan untuk kembali ke halaman utama.

4. Rancangan Maintain data Mata Kuliah

Rancangan tampilan maintain data mata kuliah berfungsi untuk mengubah data mata kuliah pada sistem. Berikut rancangan data tampilan maintain data mata kuliah yang dibuat:



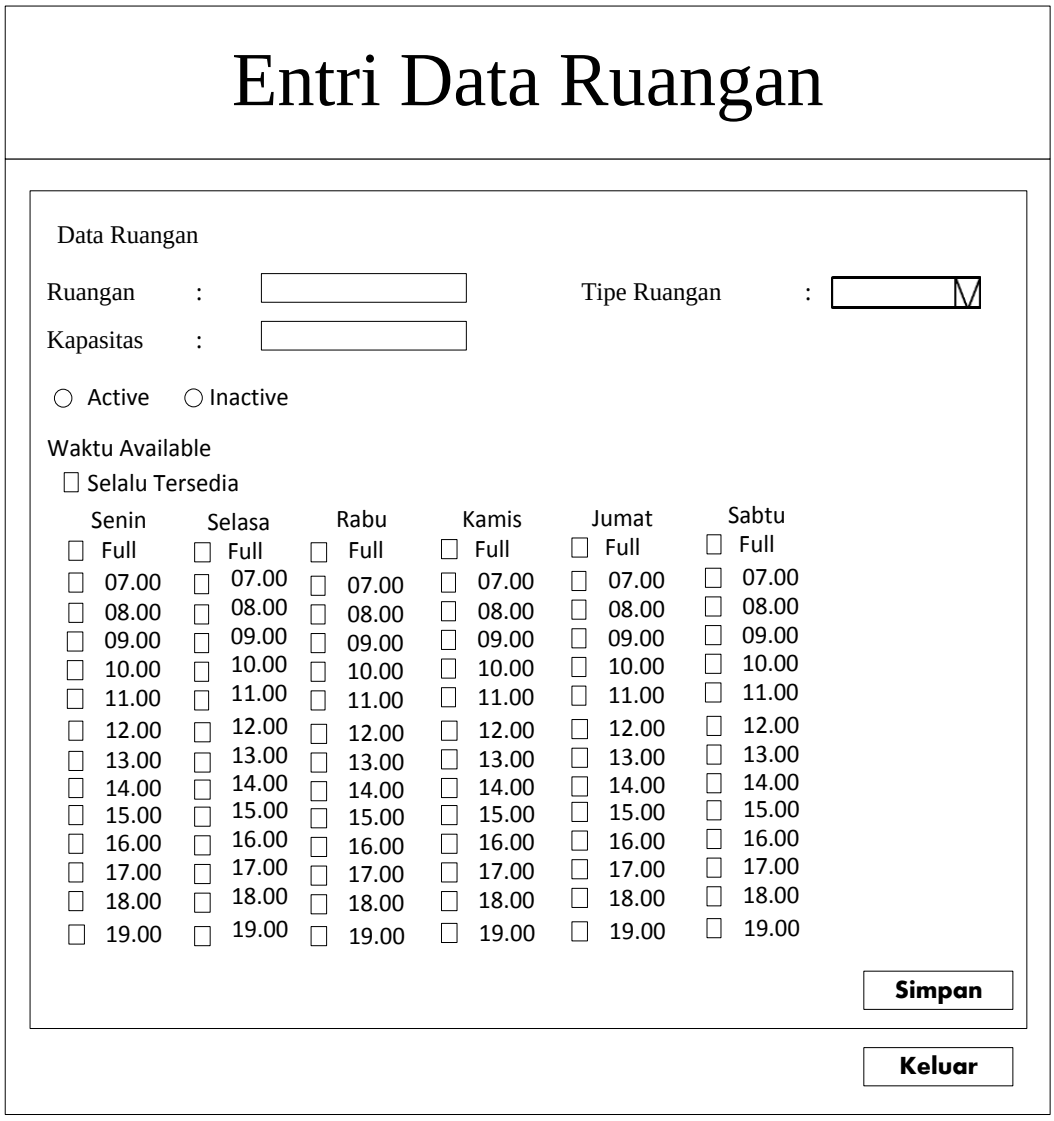
Gambar 4. 31 Desain tampilan maintain data mata kuliah

Keterangan:

a. *Textbox* Kode Matakuliah, nama kuliah, dan sks digunakan untuk mengisi data dari mata kuliah yang ingin dicari.
b. *Combobox* Semester, tipe, dan kategori dipilih untuk menentukan data mata kuliah yang ingin dicari.
c. *Listbox* menampilkan data mata kuliah sesuai dengan yang dicari.
d. Tombol Cari digunakan untuk mencari data mata kuliah sesuai dengan kondisi yang diinginkan.
e. Tombol Edit digunakan untuk mengedit data matakuliah yang dipilih dari *listbox* dan kemudian akan masuk kehalaman entry mata kuliah dengan kondisi data mata kuliah yang telah dipilih.
f. Tombol keluar digunakan untuk kembali ke halaman utama.

5. Rancangan Entry data Ruangan

Rancangan tampilan entry data ruangan berfungsi untuk menambahkan data ruangan pada sistem. Berikut rancangan data tampilan entry data mata ruangan yang dibuat:



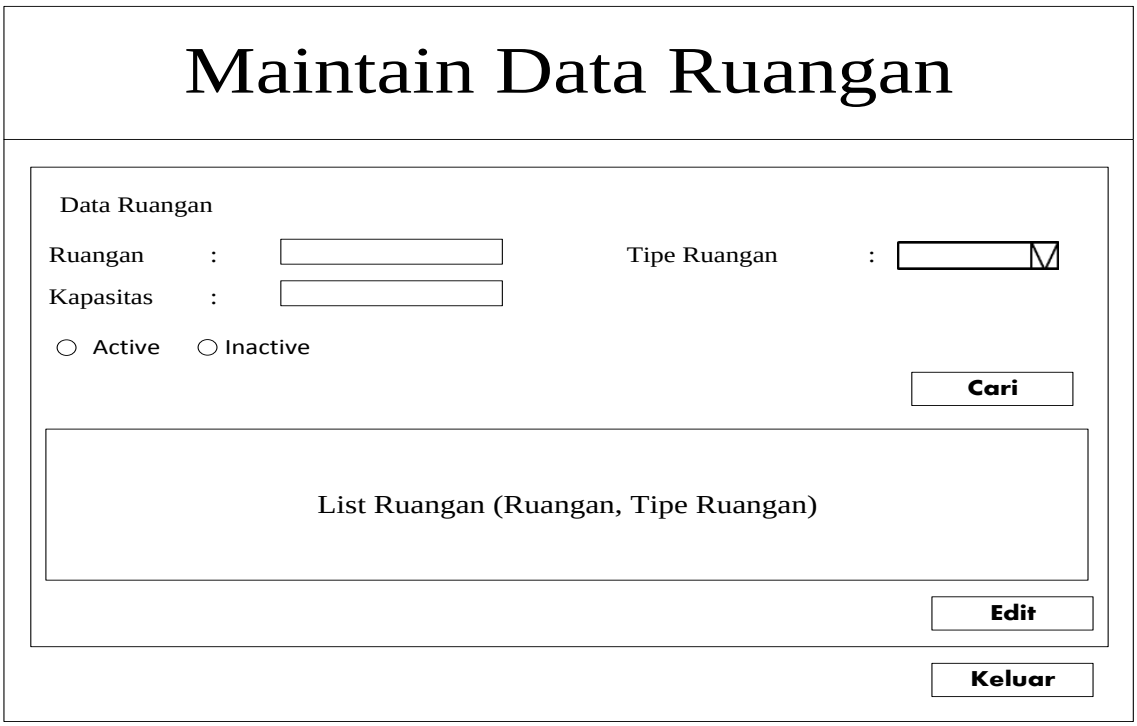
Gambar 4. 32 Desain tampilan untuk entry data ruangan

Keterangan:

a. *Textbox* Ruangan dan Kapasitas digunakan untuk mengisi data dari ruangan.
b. *Radio button* active dan inactive berfungsi untuk menentukan status ruangan aktif dipakai atau tidak.
c. *Combobox* Tipe Ruangan digunakan untuk menentukan tipe dari mata kuliah tersebut (Teori, Lab software, Lab Hardware, AUD).
d. *Checkbox* untuk waktu available menentukan kapan ketersediaan ruangan untuk dipakai. Memilih selalu tersedia akan mencentang semua waktu, memilih full akan mencentang semua waktu yang tersedia pada hari tersebut.
e. Tombol simpan berfungsi untuk memasukkan data ruangan yang baru.
f. Tombol keluar digunakan untuk kembali ke halaman utama.

6. Rancangan tampilan maintain data ruangan

   Rancangan tampilan maintain data ruangan berfungsi untuk mengubah data ruangan pada sistem. Berikut rancangan data tampilan maintain data ruangan yang dibuat:



Gambar 4. 33 Desain tampilan maintain data ruangan

Keterangan:

a. *Textbox* Ruangan dan Kapasitas digunakan untuk mengisi data dari ruangan yang ingin dicari.
b. *Radio button* active dan inactive berfungsi untuk menentukan status ruangan aktif dipakai atau tidak yang ingin dicari.
c. *Combobox* Tipe Ruangan digunakan untuk menentukan tipe dari mata kuliah tersebut (Teori, Lab software, Lab Hardware, AUD) yang ingin dicari.
d. Tombol cari digunakan untuk mencari data sesuai dengan kondisi yang diinginkan.

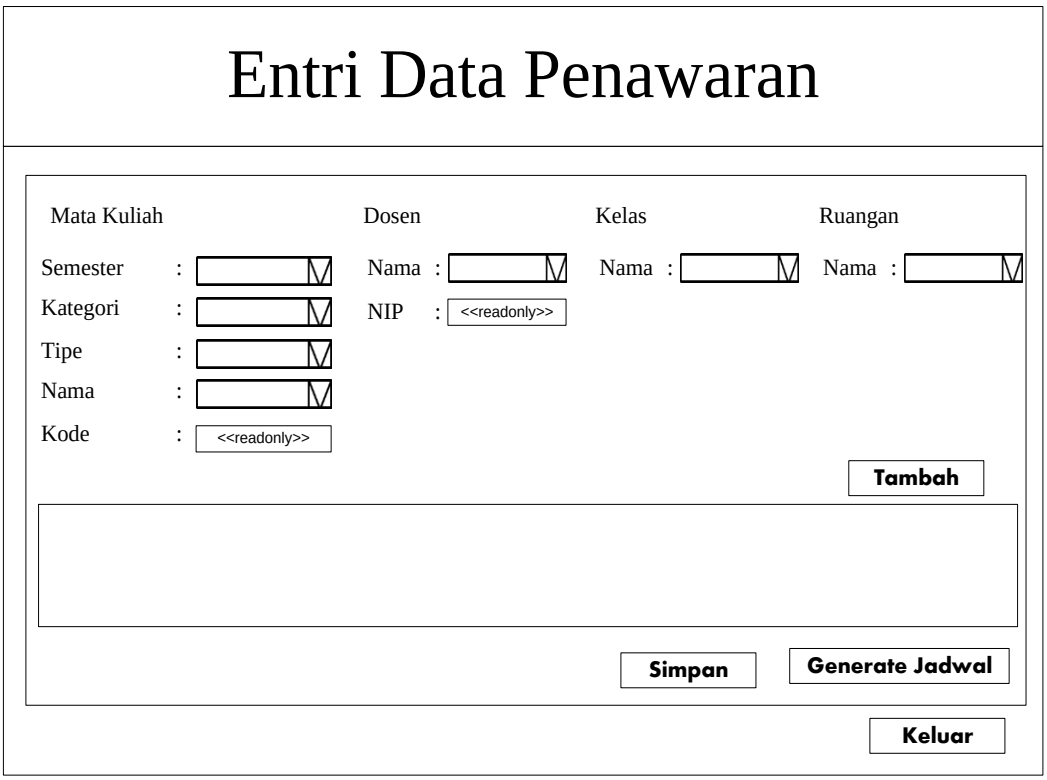
e. *Listbox* digunakan untuk menampilkan data ruangan sesuai dengan kondisi yang dicari.

f. Tombol Edit digunakan untuk mengedit data ruangan yang dipilih dari *listbox* dan kemudian akan masuk kehalaman entry ruangan dengan kondisi data ruangan yang telah dipilih.

g. Tombol keluar digunakan untuk kembali ke halaman utama.

7. Rancangan tampilan penawaran

Rancangan tampilan penawaran berfungsi untuk menentukan data penawaran mata kuliah pada dosen. Berikut rancangan data tampilan penawaran yang dibuat:



Gambar 4. 34 Desain tampilan untuk menambah data penawaran

Keterangan:

a. *Combobox* semester, kategori, tipe, nama digunakan untuk memilih data matakuliah yang ada (pada kelompok mata kuliah).
b. *Combobox* nama pada kelompok dosen digunakan untuk memilih data dosen yang tersedia.
c. *Combobox* nama pada kelas digunakan untuk memilih data kelas yang tersedia.
d. *Combobox* ruangan digunakan untuk memilih data ruangan yang tersedia sesuai dengan matakuliah yang dipilih.
e. Tombol tambah berfungsi untuk menyimpan data dosen yang memilih kelas tertentu.

f. Tombol Keluar berfungsi untuk keluar dari halaman penawaran ke halaman utama

g. Tombol Generate Jadwal berfungsi untuk men-generate jadwal kuliah

8. Rancangan tampilan maintain kelas

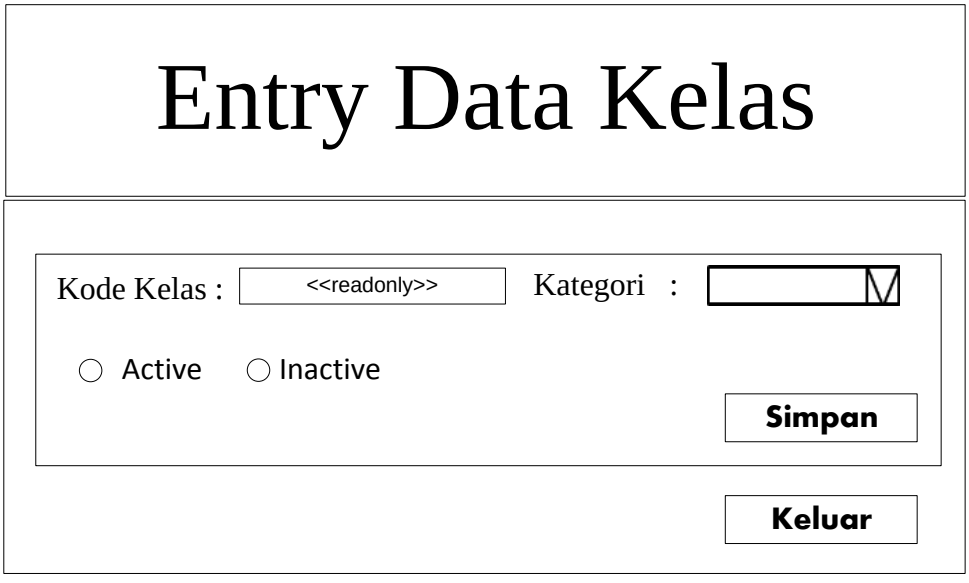
Rancangan tampilan maintain kelas berfungsi untuk mencari data kelas dan kemudian untuk menngubah datanya. Berikut rancangan data tampilan maintain kelas yang dibuat:



Gambar 4. 35 Desain tampilan untuk memaintain data kelas

Keterangan:

a. *Textbox* digunakan untuk menentukan kondisi kode kelas yang ingin dicari.
b. *Radio button* active dan inactive berfungsi untuk menentukan status kelas yang ingin dicari, aktif atau tidak.
c. *Combobox* kategori digunakan untuk memilih kategori kelas (Normal, Pilihan, Mengulang) yang ingin dicari
d. Tombol Cari digunakan untuk mencari data kelas yang diinginkan sesuai kondisi pencarian.
e. Tombol Edit digunakan untuk mengedit data kelas yang dipilih dari *listbox* dan kemudian akan masuk kehalaman entry data kelas dengan kondisi data kelas yang telah dipilih.
f. Tombol keluar digunakan untuk kembali ke halaman utama.



Gambar 4. 36 Desain tampilan entry data kelas

Keterangan:

a. *Textbox* kode kelas merupakan *textbox readonly* yang akan terisi secara otomatis sesuai dengan data kelas yang ingin diedit.

b. *Radio button* active dan inactive berfungsi untuk menentukan status kelas aktif atau tidak.

c. *Combobox* kategori digunakan untuk memilih kategori kelas (Normal, Pilihan, Mengulang).

d. Tombol Simpan digunakan untuk menyimpan data kelas yang telah diubah.

e. Tombol keluar digunakan untuk kembali ke halaman utama.

9. Rancangan tampilan BAP (Berita Acara Perkuliahan)

BAP Berita Acara Perkuliahan
Semester Ganjil – Tahun Akademik 2012-2013

Nama Dosen : Lusi Melian S.Si, MT
NIP : 41277026003
Prodi : Sistem Informasi

No.	Tgl.	Mata Kuliah	SKS	Sem	Kls/Prog	Prodi	Waktu Mengajar		Ruang	Materi	Pengesahan Pengajar oleh Mhs		
							Hari	Waktu					
1		Matematika	3	1	MI-2/D3	MI	Senin	09.25-11.30	5607				
2		Matematika	3	1	MI-1/D3	MI	Senin	11.30-13.45	5607				
3		Matematika1	3	1	SI-08/S1	SI	Selasa	09.25-11.30	5606				
4		Matematika1	3	1	SI-07/S1	SI	Rabu	10.00-12.15	4403				
5		Matematika1	3	1	SI-10/S1	SI	Kamis	09.15-11.30	4403				
6		Matematika1	3	1	SI-11/S1	SI	Kamis	11.30-13.45	5206				
7		Matematika1	3	1	SI-09/S1	SI	Jumat	09.15-11.30	5206				
		Total SKS/Minggu	**21**										
		SKS Wajib/Minggu											
		KelebihanSKS/Minggu											

Perhatian
1. BAP ini diserahkan ke Petugas Front Office selambat-lambatnya hari sabtu setiap minggunya
2. Tidak diperkenankan menambah mengurangi, merubah jadwal kuliah diatas tanpa izin tertulis dari WR-1 dengan tembusan ke Rektor, WR-II, Kaprodi dan Bendahara UNIKOM

Tembusan:
1. Yth, Wakil Rektor I
2. Yth, Wakil Rektor II
3. Yth, Dekan
4. Yth, Ketua Prodi ysb
5. Para Dosen di Prodi ysb
6. Yth, Bendahara UNIKOM
7. Arsip

BAP diserahkan ke petugas pada:
Hari: Jam :
Tanggal:

Petugas Dosen, Mengetahui, Wakil Rektor I

(……) Nama Jelas
Lusi Melian S.Si, MT
(----------)

Gambar 4. 37 Rancangan tampilan BAP

10. Rancangan tampilan Jadwal Kuliah

JADWAL KULIAH SEMESTER GANJIL TAHUN AKADEMIK 2012-2013

FAKULTAS TEKNIK & ILMU KOMPUTER

PROGRAM STUDI SISTEM INFORMASI

Kelas : SI-01

Semester : 1

Program : S1

No	Hari	Jam	Mata Kuliah	SKS	Dosen	Ruang
1.	Senin	07.00 – 09.15	Dasar Manajemen & Bisnis	3	Dadang Munandar SE, Msi	5206
2.	Senin	09.15 – 11.30	Matematika 1	3	Novrini Hasti, S.Si, MT	5606
3.	Selasa	07.00 – 09.15	Algoritma & Struktur Data 1	3	Tono Hartono S.Si, MT	5103
4.	Selasa	09.15 – 11.30	Pengantar Ilmu Komputer	3	Deasy Permatasari S.Si, MT	606
5.	Rabu	13.00 – 14.30	Komputer Aplikasi MI	2	Andri Sahata, S.Kom	Lab8
6.	Kamis	16.00 – 17.30	Komputer Aplikasi IT	2	Taufik Pontoh, S.Kom	Lab8
7.	Sabtu	07.00 – 09.30	Bahasa Inggris I	2	Annisa Rachmani, S.S, M.Hum	5606
			Total sks	**10**		

Gambar 4. 38 Rancangan tampilan Tugas Mengajar

11. Rancangan tampilan Tugas Mengajar

Tugas Mengajar

Semester Ganjil – Tahun Akademik 2012-2013

Nama Dosen : Lusi Melian S.Si, MT

NIP : 41277026003

Prodi : Sistem Informasi

No.	Tgl.	Mata Kuliah	SKS	Sem	Kls/Prog	Prodi	Waktu Mengajar		Ruang
							Hari	Waktu	
1		Matematika	3	1	MI-2/D3	MI	Senin	09.25-11.30	5607
2		Matematika	3	1	MI-1/D3	MI	Senin	11.30-13.45	5607
3		Matematika1	3	1	SI-08/S1	SI	Selasa	09.25-11.30	5606
4		Matematika1	3	1	SI-07/S1	SI	Rabu	10.00-12.15	4403
5		Matematika1	3	1	SI-10/S1	SI	Kamis	09.15-11.30	4403
6		Matematika1	3	1	SI-11/S1	SI	Kamis	11.30-13.45	5206
7		Matematika1	3	1	SI-09/S1	SI	Jumat	09.15-11.30	5206
		Total SKS/Minggu	**21**						
		SKS Wajib/Minggu							
		KelebihanSKS/Minggu							

Perhatian

1. BAP ini diserahkan ke Petugas Front Office selambat-lambatnya hari sabtu setiap minggunya
2. Tidak diperkenankan menambah mengurangi, merubah jadwal kuliah diatas tanpa izin tertulis dari WR-1 dengan tembusan ke Rektor, WR-II, Kaprodi dan Bendahara UNIKOM
3. Tunjangan kelebihan mengajar dibayar sesuai dengan pengesahan diatas

Tembusan:

1. Yth, Wakil Rektor I
2. Yth, Wakil Rektor II
3. Yth, Dekan
4. Yth, Ketua Prodi ysb
5. Para Dosen di Prodi ysb
6. Yth, Bendahara UNIKOM
7. Arsip

Rektor,

(----------)

Gambar 4. 39 Rancangan tampilan Tugas Mengajar