

**PERANCANGAN COMPUTER ASSISTED TEST (CAT)
MENGUNAKAN BAHASA PEMOGRAMAN NODE.JS DAN DATABASE
POSTGRESQL BERBASIS WEB/MOBILE PADA UNIVERSITAS PUTRA
INDONESIA “YPTK” PADANG**

SKRIPSI

*Untuk Memenuhi Sebagian Persyaratan
Mencapai Gelar Sarjana Komputer
Sistem Informasi Strata I*



Dosen Pembimbing I :
Silfia Andini ,M.Kom

Diajukan Oleh:
EGO DAFMA DASA
14101152610565

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PUTRA INDONESIA “YPTK” PADANG
PADANG
2017**

BAB III

ANALISA DAN PERANCANGAN

3.1 Analisa Sistem Yang Sedang Berjalan

Analisa sistem dapat diartikan sebagai kegiatan memahami sebuah sistem. Dengan memahami sebuah sistem, maka dapat diketahui tentang cara kerja sistem, kelebihan sistem dan kekurangan.

Dari hasil penelitian yang telah dilakukan di Universitas Putra Indonesia “YPTK” Padang baik secara langsung melalui *survey* maupun tidak langsung, maka dapat diketahui bahwa sistem pelaksanaan ujian di Universitas Putra Indonesia “YPTK” Padang masih manual. Dimana kertas sangat dibutuhkan sebagai media soal dan media jawaban. Proses pemeriksaan jawaban masih dilakukan secara manual oleh dosen sehingga cukup menyulitkan dosen mengelola jumlah kertas jawaban yang jumlahnya cukup banyak. Cara kerja sistem pelaksanaan ujian yang sedang berjalan pada Universitas Putra Indonesia “YPTK” Padang dapat dijelaskan sebagai berikut :

1. Dosen membuat soal untuk pelaksanaan ujian pada aplikasi *office* seperti *Microsoft word* atau sejenisnya.
2. Kemudian dosen memberikan soal tersebut kepada panitia pelaksana ujian dan nantinya soal tersebut dicetak berdasarkan jumlah mahasiswa yang ikut ujian.
3. Lembar jawaban berupa kertas doublefolio juga disediakan bagi peserta ujian.

4. Mahasiswa datang ujian sesuai dengan jadwal yang telah ditentukan.
5. Pada saat ujian, mahasiswa diwajibkan untuk menunjukkan kartu ujian.
6. Mahasiswa mengerjakan soal dan mengumpulkan lembar jawaban jika ujian telah selesai dilaksanakan.
7. Dosen menerima lembar jawaban dari panitia ujian dan melakukan pemeriksaan terhadap lembar jawaban tersebut dengan menuliskan nilai ujian langsung pada lembar jawaban mahasiswa.

3.1.1 Analisa *Input*, Proses dan *Output*

Pada sebuah sistem, terdapat 3 komponen utama yaitu *input* yang merupakan masukkan kesistem, proses yang merupakan pengolahan *input* dan *output* yang merupakan hasil dari proses. *Input*, proses dan *output* yang ditemukan pada sistem pelaksanaan ujian Universitas Putra Indonesia “YPTK” Padang dapat dijelaskan sebagai berikut :

1. Analisa *input*

Input yang ditemukan adalah data mahasiswa, data dosen, data mata kuliah, data kuliah, data soal dan data jawaban.

2. Analisa proses

Proses yang ditemukan adalah proses pemeriksaan jawaban peserta ujian oleh dosen. Proses pemeriksaan jawaban dilakukan secara manual oleh dosen dimana dosen memiliki lembar jawaban peserta dan mengecek lembar jawaban tersebut satu per satu.

3. Analisa *output*

Ouput yang ditemukan adalah laporan hasil ujian peserta ujian. Laporan ini didapat setelah dosen melakukan pemeriksaan terhadap lembar jawaban peserta ujian.

3.1.2 Evaluasi Sistem Yang Sedang Berjalan

Berdasarkan analisa diatas, diketahui bahwa soal yang telah dibuat dosen dicetak pada kertas dan mahasiswa menjawab soal tersebut menggunakan kertas *doublefolio* sebagai media lembar jawaban. Kendala yang ditemukan pada sistem lama dapat dijelaskan sebagai berikut :

1. Pada sistem lama, kertas sangat dibutuhkan. Karena lembar soal dan jawaban menggunakan kertas. Kejadian kurangnya lembar jawaban atau lembaran soal bisa menjadi masalah jika jumlah kertas yang akan digunakan tidak diprediksi dengan baik.
2. Pada beberapa kasus, dosen tidak bisa membaca tulisan mahasiswa karena setiap mahasiswa memiliki tulisan yang berbeda-beda sehingga hal tersebut membingungkan dosen dalam memberikan nilai akhir.
3. Dosen sedikit kewalahan dalam mengelola lembar jawaban mahasiswa karena rata-rata dalam sebuah kelas, terdapat lebih dari 30 mahasiswa dan seorang dosen tidak hanya mengajar sebuah kelas dengan mata kuliah yang sama. Sehingga dosen akan cukup kewalahan dalam mengelola lembar jawaban mahasiswa yang jumlahnya cukup banyak.

3.1.3 Usulan Sistem Baru

Berdasarkan evaluasi sistem yang telah dijelaskan sebelumnya, akan dijelaskan usulan sistem baru yang akan dikembangkan :

1. Merancang sebuah aplikasi *computer assisted test* yang dapat mempermudah pelaksanaan ujian.
2. Aplikasi tersebut dikembangkan menggunakan bahasa pemrograman *javascript* dengan bantuan *nodejs*. Karena pada pelaksanaan ujian, akan terjadi keadaan dimana banyak mahasiswa akan mengakses aplikasi *computer assisted test* dalam waktu bersamaan. Dan *nodejs* mampu menangani hal tersebut lebih baik karena cara kerjanya yang bersifat *non blocking* sehingga penggunaan *resource* dapat lebih efisien.

3.2 Desain Sistem Baru

Salah satu tujuan dari pengembangan sistem baru yaitu membenahi kekurangan yang ada pada sistem lama. Dengan demikian, desain sistem baru haruslah lebih baik daripada sistem lama baik dari segi biaya maupun dari segi efisiensi.

Desain sistem baru terdiri atas 2 (dua) bagian yaitu desain sistem secara global atau secara umum dan desain sistem secara terinci.

3.2.1 Desain Secara Global

Desain secara global diartikan sebagai gambaran secara umum kepada pengguna tentang sistem yang akan dikembangkan. Desain sistem secara global merupakan persiapan untuk membangun desain yang lebih rinci. Desain yang

tidak dapat dijelaskan secara global nantinya akan dijelaskan pada desain terinci. Dan pada penelitian ini, penulis akan menggunakan *Unified Modelling Language* atau UML dalam mendesain sistem secara global.

3.2.1.1 *Usecase Diagram*

Usecase diagram merupakan gambaran dari aktifitas-aktifitas yang akan dilakukan oleh pengguna sistem. *Usecase diagram* juga memberikan gambaran tentang bagaimana sistem berinteraksi dengan orang yang berada diluar sistem atau biasa disebut sebagai aktor. Daftar aktor pada desain sistem baru dapat dijelaskan pada Tabel 3.1 :

Tabel 3.1: Definisi Aktor

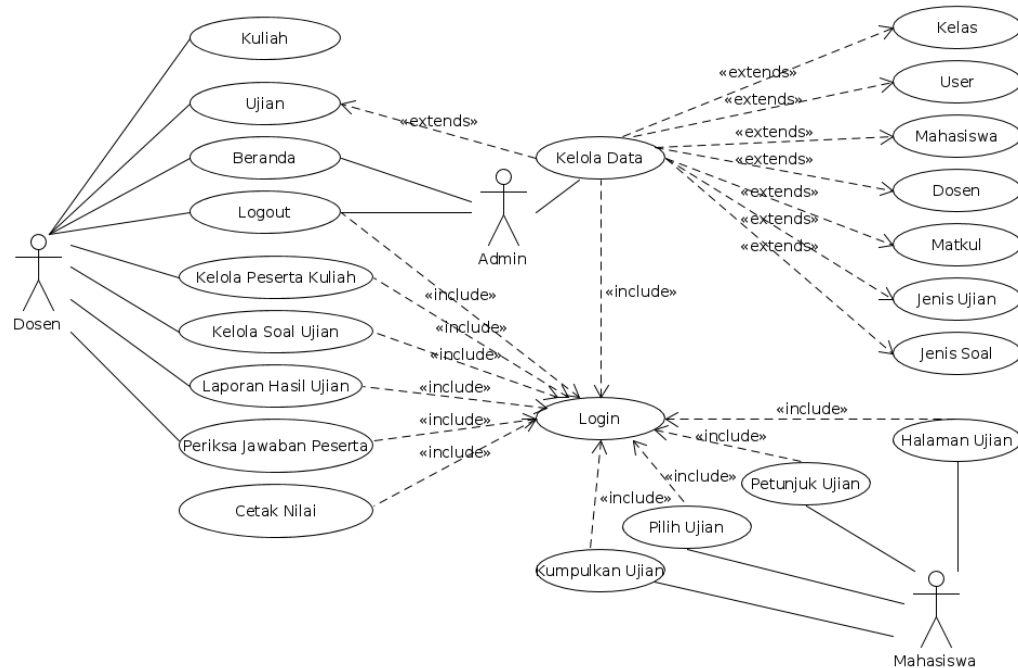
No	Aktor	Deskripsi
1	Admin	Aktor yang mengelola sistem.
2	Dosen	Aktor yang akan memproses hasil ujian serta melihat hasil akhir ujian mahasiswa.
3	Mahasiswa	Aktor yang akan mengikuti ujian.

Adapun daftar *usecase* beserta aktornya pada desain sistem baru adalah sebagai berikut :

1. Beranda : Admin, Dosen
2. Kelola User : Admin
3. Kelola Mahasiswa : Admin
4. Kelola Dosen : Admin
5. Kelola Matkul : Admin
6. Kelola Jenis Soal : Admin
7. Kelola Jenis Ujian : Admin

8. Kelola Kelas : Admin
9. Kelola Ujian : Dosen
10. Kelola Soal Ujian : Dosen
11. Kelola Peserta Kuliah : Dosen
12. Hasil Ujian : Dosen
13. Periksa Jawaban Peserta : Dosen
14. Login : Admin, Dosen, Mahasiswa
15. Petunjuk Ujian : Mahasiswa
16. Halaman Ujian : Mahasiswa
17. Kumpulkan Ujian : Mahasiswa
18. Pilih Ujian : Mahasiswa
19. Logout :Admin, Dosen
20. Kelola Kuliah : Dosen
21. Cetak Nilai : Dosen

Berdasarkan daftar *usecase* diatas, dapat digambarkan *usecase diagram* seperti pada Gambar 3.1 :



Gambar 3.1 Usecase Diagram

3.2.1.2 Skenario Usecase

Skenario *usecase* merupakan gambaran detail sebuah proses yang terjadi pada suatu *usecase*, dimana aktifitas aktor dan sistem dalam menanggapi aktifitas aktor akan dijelaskan pada skenario *usecase*.

Pada skenario *usecase*, dijelaskan tentang kondisi awal aktor terhadap *usecase*, kemudian penjelasan detail dari skenario sukses sebuah *usecase* dan dilanjutkan dengan penjelasan skenario gagal sebuah *usecase*.

3.2.1.2.1 Skenario Usecase Beranda

Usecase beranda merupakan *usecase* yang diakses saat proses *login* berhasil. Beranda hanya akan dijumpai oleh aktor admin dan dosen. Agar lebih jelas mengenai skenario *usecase* beranda dapat dilihat pada Tabel 3.2 :

Tabel 3.2 Skenario *Usecase* Beranda

Identifikasi	
No	1
Nama	Beranda
Tujuan	Aktor melihat beranda
Deskripsi	Aktor mengakses beranda
Tipe :-	
Aktor : Admin, Dosen	
Skenario Utama	
Kondisi awal : Login	
Aksi aktor	Reaksi Sistem
Entri username dan password	Sistem cek username dan password
	Sistem menampilkan halaman beranda
Aktor melihat halaman beranda	
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Entri username dan password	Sistem cek username dan password
	Sistem gagal menampilkan halaman beranda
Entri username dan password kembali	Sistem cek username dan password
	Sistem menampilkan halaman beranda
Aktor melihat halaman beranda	
Kondisi Akhir : Aktor berhasil melihat halaman beranda	

3.2.1.2.2 Skenario *Usecase* Kelola User

Usecase kelola user bertujuan untuk mengelola data pengguna sistem dimana hanya pengguna terdaftar yang dapat masuk ke sistem. Jika muncul suatu keadaan dimana dibutuhkan lebih banyak admin, maka proses penambahan admin

baru dapat dilakukan di *usecase* kelola user. Agar lebih jelas mengenai skenario *usecase* kelola user dapat dilihat pada Tabel 3.3 :

Tabel 3.3 Skenario Usecase Kelola User

Identifikasi	
No	2
Nama	Kelola user
Tujuan	Aktor mengatur data user
Deskripsi	Aktor membaca,mengedit,membuat dan menghapus data user
Tipe :-	
Aktor : Admin	
Skenario Utama	
Kondisi awal : Halaman beranda	
Aksi aktor	Reaksi Sistem
Aktor mengelola data user(simpan,tambah,hapus,edit,cari)	Validasi data user dan sistem menyimpan data kelola user
Submit/Ok	Informasi kelola data user
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Aktor mengelola data user(simpan,tambah,hapus,edit) kembali	Validasi data user kelola user
Submit/Ok kembali	Informasi kelola data user
Kondisi Akhir : Aktor berhasil mengelola data user	

3.2.1.2.3 Skenario Usecase Kelola Mahasiswa

Usecase kelola mahasiswa bertujuan untuk mengelola data mahasiswa. Data mahasiswa dikelola oleh admin dan jika terdapat mahasiswa baru, maka data mahasiswa baru tersebut dapat ditambahkan pada *usecase* kelola mahasiswa. Agar

lebih jelas mengenai skenario *usecase* kelola mahasiswa dapat dilihat pada Tabel

3.4 :

Tabel 3.4 Skenario *Usecase* Kelola Mahasiswa

Identifikasi	
No	3
Nama	Kelola mahasiswa
Tujuan	Aktor mengatur data mahasiswa
Deskripsi	Aktor membaca, mengedit, membuat dan menghapus data mahasiswa
Tipe :-	
Aktor : Admin	
Skenario Utama	
Kondisi awal : Halaman beranda	
Aksi aktor	Reaksi Sistem
Aktor mengelola data mahasiswa(simpan,tambah,hapus,edit,cari)	Validasi data mahasiswa dan sistem menyimpan data kelola mahasiswa
Submit/Ok	Informasi kelola data mahasiswa
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Aktor mengelola data mahasiswa(simpan,tambah,hapus,edit) kembali	Validasi data mahasiswa kelola mahasiswa
Submit/Ok kembali	Informasi kelola data mahasiswa
Kondisi Akhir : Aktor berhasil mengelola data mahasiswa	

3.2.1.2.4 Skenario *Usecase* Kelola Dosen

Usecase kelola dosen bertujuan untuk mengelola data dosen. Admin bertugas mengelola data dosen tersebut. Jika terdapat dosen baru, maka admin

harus menambahkan data dosen baru tersebut. Agar lebih jelas mengenai skenario *usecase* kelola dosen dapat dilihat pada Tabel 3.5 :

Tabel 3.5 Skenario *Usecase* Kelola Dosen

Identifikasi	
No	4
Nama	Kelola dosen
Tujuan	Aktor mengatur data dosen
Deskripsi	Aktor membaca,mengedit,membuat dan menghapus data dosen
Tipe :-	
Aktor : Admin	
Skenario Utama	
Kondisi awal : Halaman beranda	
Aksi aktor	Reaksi Sistem
Aktor mengelola data dosen(simpan,tambah,hapus,edit,cari)	Validasi data dosen dan sistem menyimpan data kelola dosen
Submit/Ok	Informasi kelola data dosen
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Aktor mengelola data dosen(simpan,tambah,hapus,edit) kembali	Validasi data dosen kelola dosen
Submit/Ok kembali	Informasi kelola data dosen
Kondisi Akhir : Aktor berhasil mengelola data dosen	

3.2.1.2.5 Skenario *Usecase* Kelola Matkul

Usecase kelola mata kuliah bertujuan untuk mengelola data mata kuliah. Admin bertugas mengelola data mata kuliah tersebut. Jika terdapat mata kuliah

baru, maka admin harus menambahkan data mata kuliah baru tersebut. Agar lebih jelas mengenai skenario *usecase* kelola mata kuliah dapat dilihat pada Tabel 3.6 :

Tabel 3.6 Skenario *Usecase* Kelola Matkul

Identifikasi	
No	5
Nama	Kelola matkul
Tujuan	Aktor mengatur data matkul
Deskripsi	Aktor membaca,mengedit,membuat dan menghapus data matkul
Tipe :-	
Aktor : Admin	
Skenario Utama	
Kondisi awal : Halaman beranda	
Aksi aktor	Reaksi Sistem
Aktor mengelola data matkul(simpan,tambah,hapus,edit,cari)	Validasi data matkul dan sistem menyimpan data kelola matkul
Submit/Ok	Informasi kelola data matkul
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Aktor mengelola data matkul(simpan,tambah,hapus,edit) kembali	Validasi data matkul kelola matkul
Submit/Ok kembali	Informasi kelola data matkul
Kondisi Akhir : Aktor berhasil mengelola data matkul	

3.2.1.2.6 Skenario *Usecase* Kelola Jenis Soal

Usecase kelola jenis soal bertujuan untuk mengelola data jenis soal. Admin bertugas mengelola data jenis soal tersebut. Jika terdapat jenis soal baru, maka admin harus menambahkan data jenis soal baru tersebut jika diperlukan. Agar

lebih jelas mengenai skenario *usecase* kelola jenis soal dapat dilihat pada Tabel 3.7 :

Tabel 3.7 Skenario *Usecase* Kelola Jenis Soal

Identifikasi	
No	6
Nama	Kelola jenis soal
Tujuan	Aktor mengatur data jenis soal
Deskripsi	Aktor membaca,mengedit,membuat dan menghapus data jenis soal
Tipe :-	
Aktor : Admin	
Skenario Utama	
Kondisi awal : Halaman beranda	
Aksi aktor	Reaksi Sistem
Aktor mengelola data jenis soal(simpan,tambah,hapus,edit,cari)	Validasi data jenis soal dan sistem menyimpan data kelola jenis soal
Submit/Ok	Informasi kelola data jenis soal
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Aktor mengelola data jenis soal(simpan,tambah,hapus,edit) kembali	Validasi data jenis soal kelola jenis soal
Submit/Ok kembali	Informasi kelola data jenis soal
Kondisi Akhir : Aktor berhasil mengelola data jenis soal	

3.2.1.2.7 Skenario *Usecase* Kelola Jenis Ujian

Usecase kelola jenis ujian bertujuan untuk mengelola data jenis ujian. Admin bertugas mengelola data jenis ujian tersebut. Jika terdapat jenis ujian baru,

maka admin harus menambahkan data jenis ujian baru tersebut. Agar lebih jelas mengenai skenario *usecase* kelola jenis ujian dapat dilihat pada Tabel 3.8 :

Tabel 3.8 Skenario *Usecase* Kelola Jenis Ujian

Identifikasi	
No	7
Nama	Kelola jenis ujian
Tujuan	Aktor mengatur data jenis ujian
Deskripsi	Aktor membaca,mengedit,membuat dan menghapus data jenis ujian
Tipe :-	
Aktor : Admin	
Skenario Utama	
Kondisi awal : Halaman beranda	
Aksi aktor	Reaksi Sistem
Aktor mengelola data jenis ujian(simpan,tambah,hapus,edit,cari)	Validasi data jenis ujian dan sistem menyimpan data kelola jenis ujian
Submit/Ok	Informasi kelola data jenis ujian
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Aktor mengelola data jenis ujian(simpan,tambah,hapus,edit) kembali	Validasi data jenis ujian kelola jenis ujian
Submit/Ok kembali	Informasi kelola data jenis ujian
Kondisi Akhir : Aktor berhasil mengelola data jenis ujian	

3.2.1.2.8 Skenario *Usecase* Kelola Kelas

Usecase kelola kelas bertujuan untuk mengelola data kelas. Admin bertugas mengelola data kelas tersebut. Jika terdapat kelas baru, maka admin harus

menambahkan data kelas baru tersebut. Agar lebih jelas mengenai skenario *usecase* kelola kelas dapat dilihat pada Tabel 3.9 :

Tabel 3.9 Skenario *Usecase* Kelola Kelas

Identifikasi	
No	8
Nama	Kelola Kelas
Tujuan	Aktor mengatur data Kelas
Deskripsi	Aktor membaca,mengedit,membuat dan menghapus data Kelas
Tipe :-	
Aktor : Admin	
Skenario Utama	
Kondisi awal : Halaman beranda	
Aksi aktor	Reaksi Sistem
Aktor mengelola data Kelas(simpan,tambah,hapus,edit,cari)	Validasi data Kelas dan sistem menyimpan data kelola Kelas
Submit/Ok	Informasi kelola data Kelas
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Aktor mengelola data Kelas(simpan,tambah,hapus,edit) kembali	Validasi data Kelas kelola Kelas
Submit/Ok kembali	Informasi kelola data Kelas
Kondisi Akhir : Aktor berhasil mengelola data Kelas	

3.2.1.2.9 Skenario *Usecase* Kelola Ujian

Usecase kelola ujian dilakukan oleh aktor dosen. Dosen dapat membuat ujian baru dan nanti ujian tersebut akan di ikuti oleh mahasiswa sesuai dengan

kuliah yang di ambil. Agar lebih jelas mengenai skenario *usecase* kelola ujian dapat dilihat pada Tabel 3.10 :

Tabel 3.10 Skenario *Usecase* Kelola Ujian

Identifikasi	
No	9
Nama	Kelola ujian
Tujuan	Aktor mengatur data ujian
Deskripsi	Aktor membaca,mengedit,membuat dan menghapus data ujian
Tipe :-	
Aktor : Admin, Dosen	
Skenario Utama	
Kondisi awal : Halaman beranda	
Aksi aktor	Reaksi Sistem
Aktor mengelola data ujian(simpan,tambah,hapus,edit,cari)	Validasi data ujian dan sistem menyimpan data kelola ujian
Submit/Ok	Informasi kelola data ujian
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Aktor mengelola data ujian(simpan,tambah,hapus,edit) kembali	Validasi data ujian kelola ujian
Submit/Ok kembali	Informasi kelola data ujian
Kondisi Akhir : Aktor berhasil mengelola data ujian	

3.2.1.2.10 Skenario *Usecase* Kelola Soal Ujian

Usecase kelola soal ujian dilakukan oleh dosen. Dosen dapat menambahkan soal ujian baru dengan tipe soal essai dan pilihan ganda. Agar lebih jelas mengenai skenario *usecase* kelola soal ujian dapat dilihat pada tabel 3.11 :

Tabel 3.11 Skenario *Usecase* Kelola Soal Ujian

Identifikasi	
No	10
Nama	Kelola soal ujian
Tujuan	Aktor mengatur data soal ujian
Deskripsi	Aktor membaca,mengedit,membuat dan menghapus data soal ujian
Tipe :-	
Aktor : Dosen	
Skenario Utama	
Kondisi awal : Halaman beranda	
Aksi aktor	Reaksi Sistem
Aktor mengelola data soal ujian(simpan,tambah,hapus,edit,cari)	Validasi data soal ujian dan sistem menyimpan data kelola soal ujian
Submit/Ok	Informasi kelola data soal ujian
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Aktor mengelola data soal ujian(simpan,tambah,hapus,edit) kembali	Validasi data soal ujian kelola soal ujian
Submit/Ok kembali	Informasi kelola data soal ujian
Kondisi Akhir : Aktor berhasil mengelola data soal ujian	

3.2.1.2.11 Skenario *Usecase* Kelola Peserta Kuliah

Usecase kelola peserta kuliah bertujuan untuk mengelola peserta mahasiswa yang mengikuti suatu kuliah. Peserta kuliah nantinya akan otomatis menjadi peserta ujian. Agar lebih jelas mengenai skenario *usecase* kelola peserta kuliah dapat dilihat pada Tabel 3.12 :

Tabel 3.12 Skenario Usecase Kelola Peserta Kuliah

Identifikasi	
No	11
Nama	Kelola peserta kuliah
Tujuan	Aktor mengatur data peserta kuliah
Deskripsi	Aktor membaca, mengedit, membuat dan menghapus data peserta kuliah
Tipe :-	
Aktor : Dosen	
Skenario Utama	
Kondisi awal : Halaman beranda	
Aksi aktor	Reaksi Sistem
Aktor mengelola data peserta kuliah(simpan,tambah,hapus,cari)	Validasi data peserta kuliah dan sistem menyimpan data kelola peserta kuliah
Submit/Ok	Informasi kelola data peserta kuliah
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Aktor mengelola data peserta kuliah(simpan,tambah,hapus,edit) kembali	Validasi data peserta kuliah kelola peserta kuliah
Submit/Ok kembali	Informasi kelola data peserta kuliah
Kondisi Akhir : Aktor berhasil mengelola data peserta kuliah	

3.2.1.2.12 Skenario Usecase Hasil Ujian

Usecase hasil ujian merupakan kegiatan dosen melihat hasil ujian peserta kuliah. Dosen masih perlu melakukan pemeriksaan jawaban jika tipe soal ujian adalah esai. Sedangkan tipe soal pilihan ganda akan menghasilkan nilai pada halaman hasil ujian secara otomatis. Agar lebih jelas mengenai skenario *usecase* hasil ujian dapat dilihat pada Tabel 3.13 :

Tabel 3.13 Skenario *Usecase* Hasil Ujian

Identifikasi	
No	12
Nama	Hasil Ujian
Tujuan	Aktor mengetahui hasil ujian mahasiswa
Deskripsi	Aktor membaca laporan hasil ujian mahasiswa
Tipe :-	
Aktor : Dosen	
Skenario Utama	
Kondisi awal : Halaman beranda	
Aksi aktor	Reaksi Sistem
Klik menu laporan hasil ujian	Sistem menampilkan laporan hasil ujian
Aktor mengentri kata kunci untuk pencarian hasil ujian mahasiswa	Sistem menampilkan hasil pencarian hasil ujian
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Klik menu laporan hasil ujian	Sistem tidak menampilkan laporan ujian
Klik kembali laporan hasil ujian	Sistem menampilkan laporan hasil ujian
Kondisi Akhir : Aktor berhasil melihat laporan hasil ujian	

3.2.1.2.13 Skenario *Usecase* Periksa Jawaban Peserta

Skenario *usecase* berikut menjelaskan proses dosen dalam melakukan pemeriksaan jawaban peserta. Pemeriksaan jawaban hanya berlaku jika tipe soal ujian adalah essai. Agar lebih jelas mengenai skenario *usecase* periksa jawaban peserta dapat dilihat pada Tabel 3.14 :

Tabel 3.14 Skenario Usecase Periksa Jawaban Peserta

Identifikasi	
No	13
Nama	Periksa Jawaban
Tujuan	Aktor memeriksa jawaban ujian peserta
Deskripsi	Aktor membaca dan memberikan nilai bobot paa setiap jawaban peserta
Tipe :-	
Aktor : Dosen	
Skenario Utama	
Kondisi awal : Halaman Kelola Ujian	
Aksi aktor	Reaksi Sistem
Klik menu peserta ujian	Sistem menampilkan daftar peserta ujian
Klik tombol periksa jawaban	Sistem menampilkan halaman periksa jawaban beserta jawaban peserta
Aktor memberikan nilai bobot disetiap jawaban peserta dan mengklik tombol simpan	Sistem menyimpan nilai bobot
Klik tombol simpan hasil periksa	Sistem menyimpan hasil pemeriksaan jawaban peserta
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Klik menu peserta ujian	Sistem tidak menampilkan peserta ujian
Klik kembali menu peserta ujian	Sistem menampilkan peserta ujian
Klik menu periksa jawaban	Sistem menampilkan halaman periksa jawaban
Klik simpan hasil periksa	Sistem tidak berhasil menyimpan hasil periksa
Klik kemmbali simpan hasil periksa	Sistem berhasil menyimpan hasil periksa
Kondisi Akhir : Aktor berhasil memeriksa jawaban peserta	

3.2.1.2.14 Skenario Usecase Login

Usecase login diakses oleh semua aktor. Karena hampir semua halaman pada aplikasi yang akan dikembangkan hanya bisa diakses oleh aktor mahasiswa, dosen dan admin. Agar lebih jelas mengenai skenario *usecase login* dapat dilihat pada Tabel 3.15 :

Tabel 3.15 Skenario Usecase Login

Identifikasi	
No	14
Nama	Login
Tujuan	Aktor masuk kesistem
Deskripsi	Aktor mengentri informasi yang dibutuhkan untuk masuk kesistem
Tipe :-	
Aktor : Admin, Dosen, Mahasiswa	
Skenario Utama	
Kondisi awal : Halaman login	
Aksi aktor	Reaksi Sistem
Entri username dan password serta klik tombol login	Sistem mengecek username dan password
	Sistem menemukan username dan password yang cocok
	Sistem mengarahkan aktor kedalam sistem
Aktor berada didalam sistem	
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Entri username dan password	Sistem cek username dan password
	Sistem tidak berhasil mengecek username dan password
Entri kembali username dan	Sistem berhasil mengecek username

password	dan password
Kondisi Akhir : Aktor berhasil login	

3.2.1.2.15 Skenario *Usecase* Petunjuk Ujian

Usecase ini akan diakses oleh mahasiswa yang akan mengikuti ujian. Halaman ini bertujuan untuk memberi sedikit gambaran tentang penggunaan aplikasi *computer assisted test*. Agar lebih jelas mengenai skenario *usecase* petunjuk ujian dapat dilihat pada Tabel 3.16 :

Tabel 3.16 Skenario *Usecase* Petunjuk Ujian

Identifikasi	
No	15
Nama	Petunjuk Ujian
Tujuan	Aktor mengetahui petunjuk ujian
Deskripsi	Aktor membaca petunjuk ujian
Tipe :-	
Aktor : Mahasiswa	
Skenario Utama	
Kondisi awal : Halaman login	
Aksi aktor	Reaksi Sistem
Login dengan username dan password	Sistem membawa aktor kehalaman petunjuk ujian
	Sistem menampilkan halaman petunjuk ujian
Aktor membaca petunjuk ujian	
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Login dengan username dan password	Sistem membawa aktor kehalaman petunjuk ujian
	Sistem gagal menampilkan petunjuk

	ujian
Login kembali	Sistem berhasil menampilkan petunjuk ujian
Kondisi Akhir : Aktor berhasil mengetahui petunjuk ujian	

3.2.1.2.16 Skenario *Usecase* Halaman Ujian

Usecase halaman ujian diakses oleh mahasiswa. Mahasiswa akan menjawab soal ujian pada halaman ini. Hanya mahasiswa yang sudah terdaftar yang dapat mengikuti ujian. Agar lebih jelas mengenai skenario *usecase* halaman ujian dapat dilihat pada Tabel 3.17 :

Tabel 3.17 Skenario *Usecase* Halaman Ujian

Identifikasi	
No	16
Nama	Halaman Ujian
Tujuan	Aktor mengikuti ujian
Deskripsi	Aktor menjawab pertanyaan yang diujikan
Tipe :-	
Aktor : Mahasiswa	
Skenario Utama	
Kondisi awal : Halaman petunjuk ujian	
Aksi aktor	Reaksi Sistem
Klik tombol masuk kehalaman ujian	Sistem menampilkan halaman ujian
Aktor membaca pertanyaan dan memilih atau mengetik jawaban dan mengklik tombol simpan jawaban	Sistem menyimpan jawaban peserta
Klik tombol kumpulkan ujian	Sistem menyimpan semua jawaban peserta
Skenario Gagal	

Aksi Aktor	Reaksi Sistem
Aktor membaca dan menjawab serta mengklik simpan jawaban	Sistem gagal menyimpan jawaban
Aktor klik kembali simpan jawaban	Sistem berhasil menyimpan jawaban
Aktor klik kumpulkan ujian	Sistem tidak berhasil mengumpulkan ujian
Klik kembali kumpulkan ujian	Sistem berhasil mengumpulkan ujian
Kondisi Akhir : Ujian peserta terlaksana	

3.2.1.2.17 Skenario *Usecase* Kumpulkan Ujian

Usecase ini dilakukan mahasiswa setelah semua soal terjawab atau waktu ujian sudah habis. Semua jawaban yang sudah dikumpulkan akan diproses untuk menghasilkan nilai ujian mahasiswa. Agar lebih jelas mengenai skenario *usecase* kumpulkan ujian dapat dilihat pada Tabel 3.18 :

Tabel 3.18 Skenario *Usecase* Periksa Kumpulkan Ujian

Identifikasi	
No	17
Nama	Kumpulkan Ujian
Tujuan	Aktor mengumpulkan ujian
Deskripsi	Aktor mengumpulkan ujian setelah selesai ujian
Tipe :-	
Aktor : Dosen	
Skenario Utama	
Kondisi awal : Halaman ujian	
Aksi aktor	Reaksi Sistem
Klik tombol kumpulkan ujian	Sistem mengumpulkan jawaban peserta dan menyimpannya
Skenario Gagal	

Aksi Aktor	Reaksi Sistem
Klik tombol kumpulkan ujian	Sistem gagal mengumpulkan jawaban peserta dan menyimpannya
Klik tombol kumpulkan ujian kembali	Sistem mengumpulkan jawaban peserta dan menyimpannya
Kondisi Akhir : Aktor berhasil logout	

3.2.1.2.18 Skenario *Usecase* Pilih Ujian

Usecase pilih ujian muncul setelah mahasiswa *login*. Pada *usecase* ini, mahasiswa dihadapkan pada daftar ujian yang bisa di ikuti. Mahasiswa hanya dapat melaksanakan 1 (satu) ujian dalam waktu bersamaan. Agar lebih jelas mengenai skenario *usecase* pilih ujian dapat dilihat pada Tabel 3.19 :

Tabel 3.19 Skenario *Usecase* Pilih Ujian

Identifikasi	
No	18
Nama	Pilih ujian
Tujuan	Aktor memilih ujian
Deskripsi	Aktor memilih ujian yang akan diikuti
Tipe :-	
Aktor : Mahasiswa	
Skenario Utama	
Kondisi awal : Halaman login	
Aksi aktor	Reaksi Sistem
Entri username dan password	Sistem cek username dan password
	Sistem menampilkan halaman pilih ujian
Aktor memilih ujian	
Skenario Gagal	

Aksi Aktor	Reaksi Sistem
Entri username dan password	Sistem cek username dan password
	Sistem gagal menampilkan halaman pilih ujian
Entri username dan password kembali	Sistem cek username dan password
	Sistem menampilkan halaman pilih ujian
Aktor memilih ujian	
Kondisi Akhir : Aktor berhasil memilih ujian	

3.2.1.2.19 Skenario Usecase Logout

Usecase ini dilakukan oleh admin dan dosen. Aktor dianjurkan untuk melakukan *logout* saat mereka tidak akan menggunakan aplikasi *computer assisted test* dalam waktu tertentu agar lebih aman terhadap orang yang tidak bertanggung jawab. Agar lebih jelas mengenai skenario *usecase logout* dapat dilihat pada Tabel 3.20 :

Tabel 3.20 Skenario Usecase Periksa Jawaban Peserta

Identifikasi	
No	19
Nama	Logout
Tujuan	Aktor keluar dari sistem
Deskripsi	Aktor keluar dari sistem setelah menggunakan sistem
Tipe :-	
Aktor : Admin, Dosen	
Skenario Utama	
Kondisi awal : Halaman beranda	
Aksi aktor	Reaksi Sistem

Klik tombol logout	Sistem membawa aktor keluar sistem
	Sistem menampilkan halaman login
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Klik tombol logout	Sistem gagal membawa aktor keluar sistem
Klik tombol logout kemmbali	Sistem berhasil membawa aktor keluar
Kondisi Akhir : Aktor berhasil logout	

3.2.1.2.20 Skenario *Usecase* Kelola Kuliah

Usecase ini dilakukan oleh dosen untuk mengelola kuliah yang di ampu. Kuliah yang telah ditambahkan nantinya akan muncul pada saat dosen ingin menambahkan ujian. Agar lebih jelas mengenai skenario *usecase* kelola kuliah dapat dilihat pada Tabel 3.21 :

Tabel 3.21 Skenario *Usecase* Kelola Kuliah

Identifikasi	
No	20
Nama	Kelola kuliah
Tujuan	Aktor mengatur datakuliah
Deskripsi	Aktor membaca,mengedit,membuat dan menghapus datakuliah
Tipe :-	
Aktor : Dosen	
Skenario Utama	
Kondisi awal : Halaman beranda	
Aksi aktor	Reaksi Sistem
Aktor mengelola data kuliah (simpan,tambah,hapus,cari)	Validasi datakuliah dan sistem menyimpan data kelolakuliah

Submit/Ok	Informasi kelola datakuliah
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Aktor mengelola datakuliah(simpan,tambah,hapus,edit) kembali	Validasi datakuliah kelolakuliah
Submit/Ok kembali	Informasi kelola datakuliah
Kondisi Akhir : Aktor berhasil mengelola data kuliah	

3.2.1.2.21 Skenario *Usecase* Cetak Nilai

Usecase ini bertujuan untuk mencetak hasil ujian mahasiswa dalam bentuk *file* seperti *file excel*. Nantinya *file* tersebut bisa digunakan oleh dosen untuk diolah lebih lanjut. Agar lebih jelas mengenai skenario *usecase* cetak nilai dapat dilihat pada Tabel 3.22 :

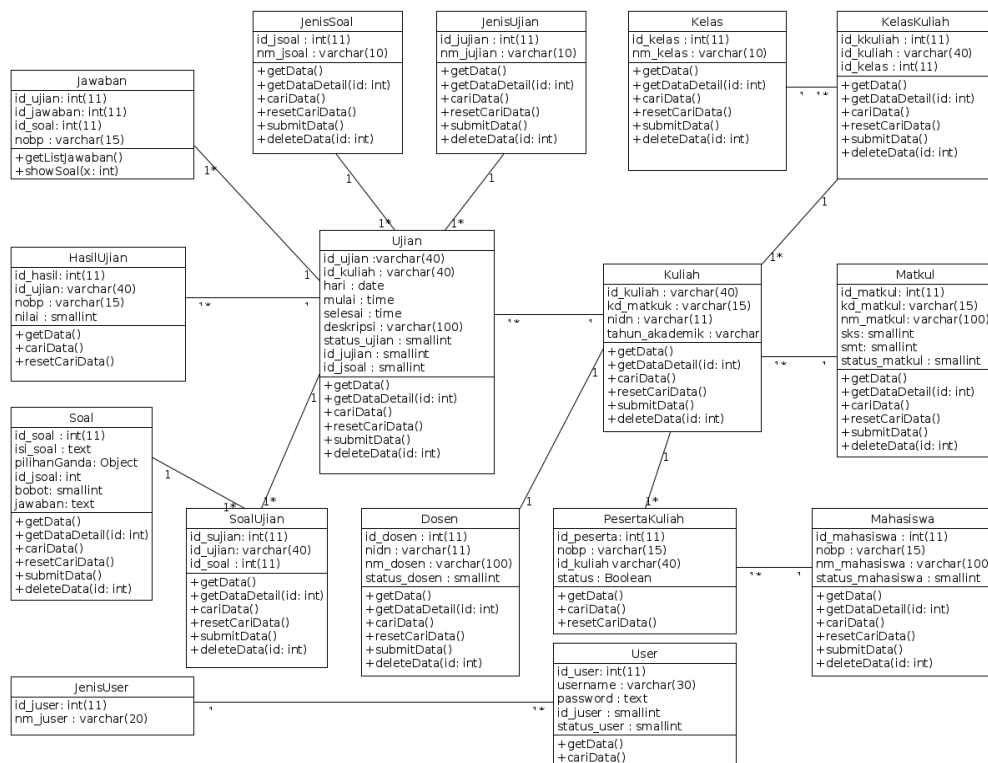
Tabel 3.22 Skenario *Usecase* Cetak Nilai

Identifikasi	
No	21
Nama	Cetak Nilai
Tujuan	Aktor Mencetak Nilai kedalam bentuk file
Deskripsi	Aktor mencetak nilai ujian mahasiswa
Tipe :-	
Aktor : Dosen	
Skenario Utama	
Kondisi awal : Halaman Hasil Ujian	
Aksi aktor	Reaksi Sistem
Aktor mengklik tombol cetak dan memilih pilihan file untuk dicetak	Sistem memberikan file nilai yang diminta aktor

Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Aktor mengklik tombol cetak dan memilih pilihan file untuk dicetak Sistem menampilkan nilai hasil ujian	Sistem gagal memberikan file nilai yang diminta aktor
Aktor mengklik kembali tombol cetak dan memilih pilihan file untuk dicetak	Sistem memberikan file nilai yang diminta aktor
Kondisi Akhir : Aktor berhasil mendapatkan file nilai mahasiswa	

3.2.1.3 Class Diagram

Class diagram menggambarkan struktur *class* serta deskripsi *class* dan *object* serta hubungan antar data-data yang terdapat pada basis data. Agar lebih jelas mengenai *class diagram* pada aplikasi yang akan dikembangkan dapat dilihat pada Gambar 3.2 :



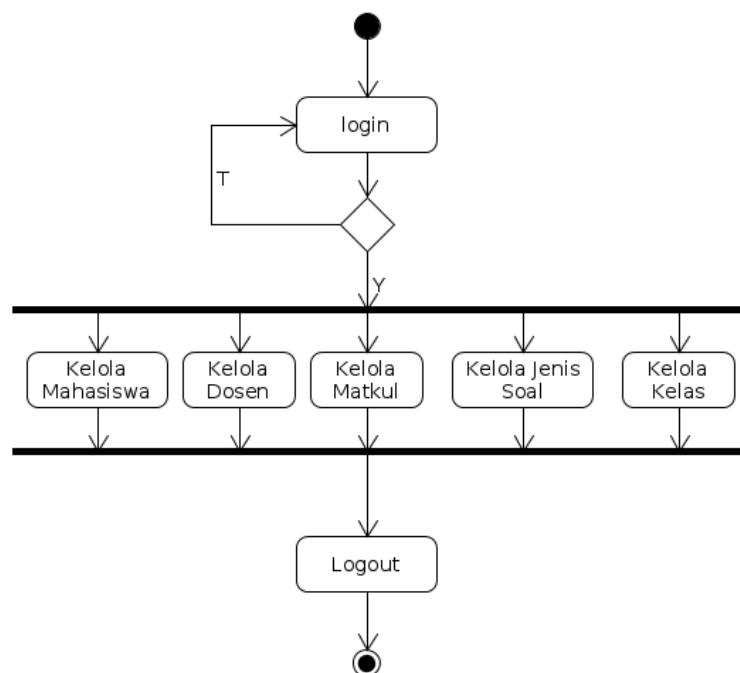
Gambar 3.2 Class Diagram

3.2.1.4 Activity Diagram

Sesuai dengan namanya, *activity diagram* menggambarkan aktivitas dari seorang aktor terhadap sebuah sistem. Aktivitas tersebut biasanya berupa *use case* yang sudah dibahas sebelumnya. Agar lebih jelas mengenai *activity diagram* pada penelitian ini dapat dijelaskan sebagai berikut :

3.2.1.4.1 Activity Diagram Admin

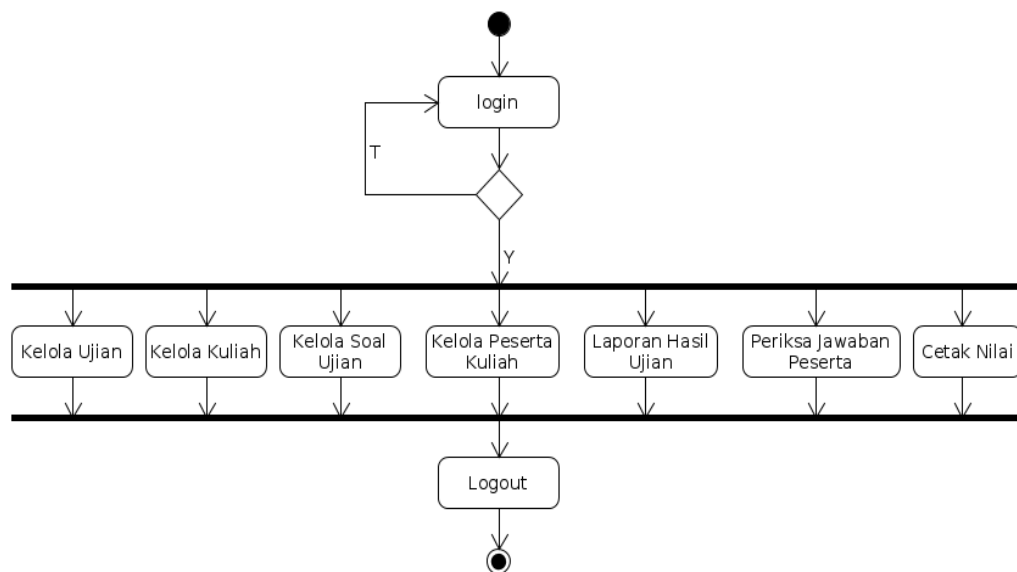
Seorang admin biasanya mengelola data utama dari sebuah sistem. Data utama tersebut nantinya akan digunakan oleh aktor lainnya. Adapun *activity diagram* seorang admin dapat digambarkan seperti Gambar 3.3 :



Gambar 3.3 Activity Diagram Admin

3.2.1.4.2 Activity Diagram Dosen

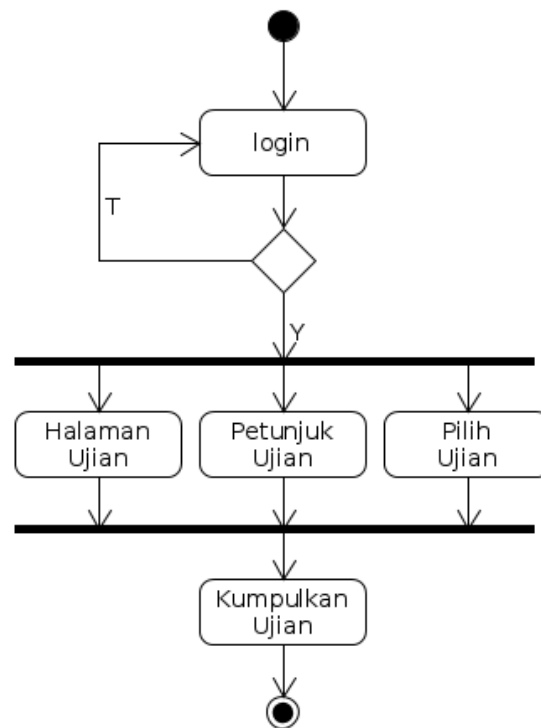
Pada sistem baru, dosen nantinya bertugas mengelola kuliah dan ujian yang telah Ia daftarkan. Dan juga dosen akan melakukan pemeriksaan terhadap jawaban peserta ujian. Adapun *activity diagram* seorang dosen dapat digambarkan seperti Gambar 3.4 :



Gambar 3.4 Activity Diagram Dosen

3.2.1.4.3 Activity Diagram Mahasiswa

Mahasiswa merupakan aktor yang akan mengikuti ujian yang telah dibuat oleh seorang dosen. Adapun *activity diagram* seorang mahasiswa dapat digambarkan seperti Gambar 3.5 :



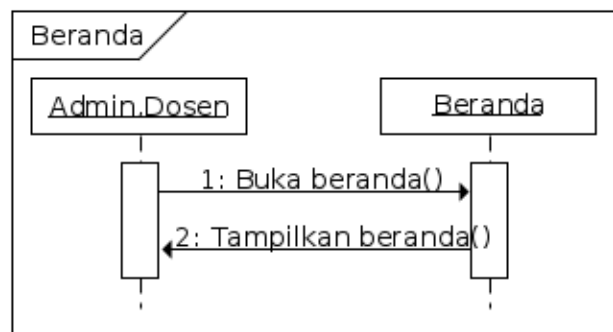
Gambar 3.5 Activity Diagram Mahasiswa

3.2.1.5 Sequence Diagram

Sequence diagram merupakan gambaran detail dari aktifitas seorang aktor pada sebuah *usecase*. Aktifitas-aktifitas itu sendiri dimulai dari aktor dan kemudian sistem membalas aktifitas tersebut. Bisa berupa menampilkan tampilan lain atau menampilkan informasi tertentu. Berikut adalah rancangan dari *sequence diagram* pada penelitian ini :

3.2.1.5.1 Sequence Diagram Beranda

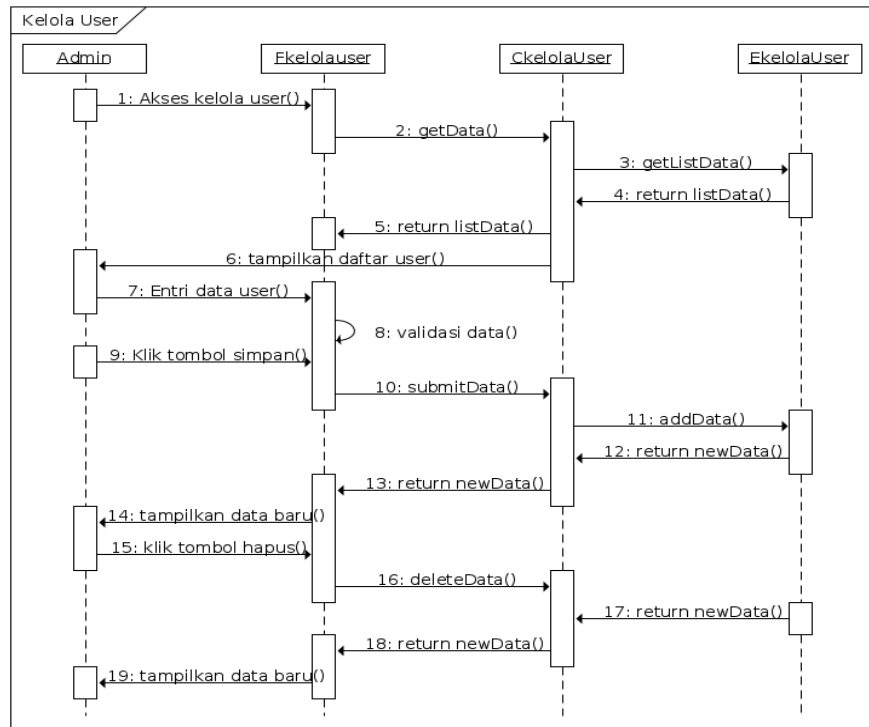
Diagram ini menggambarkan bagaimana aktor mengakses halaman beranda. Halaman ini nantinya akan muncul pertama kali saat dosen dan admin berhasil login. Adapun *sequence diagram* beranda dapat digambarkan seperti pada Gambar 3.6 :



Gambar 3.6 Sequence Diagram beranda

3.2.1.5.2 Sequence Diagram Kelola User

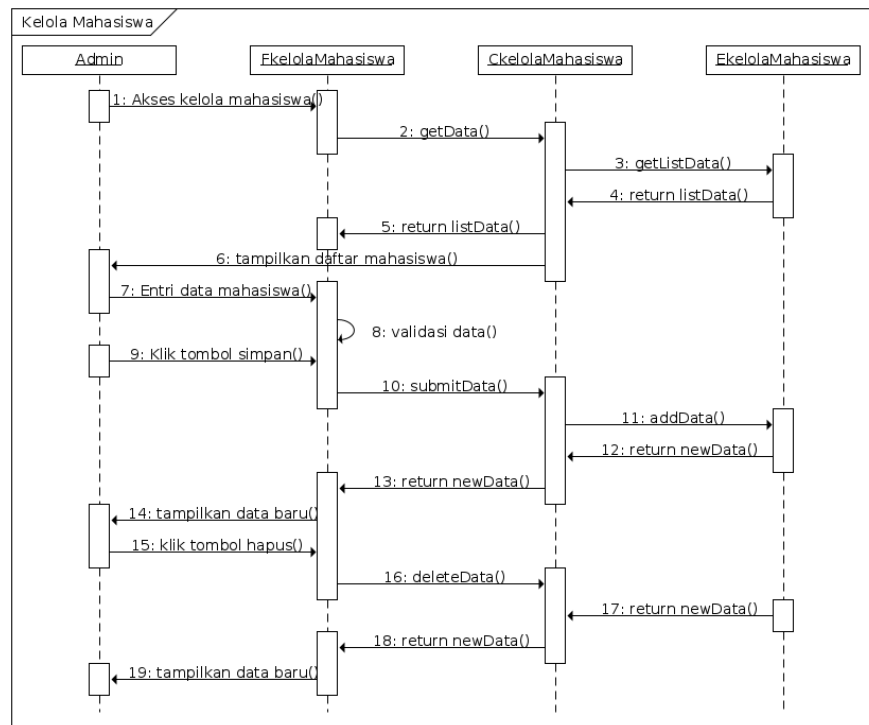
Sequence diagram ini menggambarkan tentang aktifitas admin dalam mengelola *user*. Admin dapat mengatur siapa saja yang dapat masuk kesistem melalui halaman ini. Adapun *sequence diagram* kelola user yang dapat digambarkan seperti pada Gambar 3.7 :



Gambar 3.7 Sequence Diagram Kelola User

3.2.1.5.3 Sequence Diagram Kelola Mahasiswa

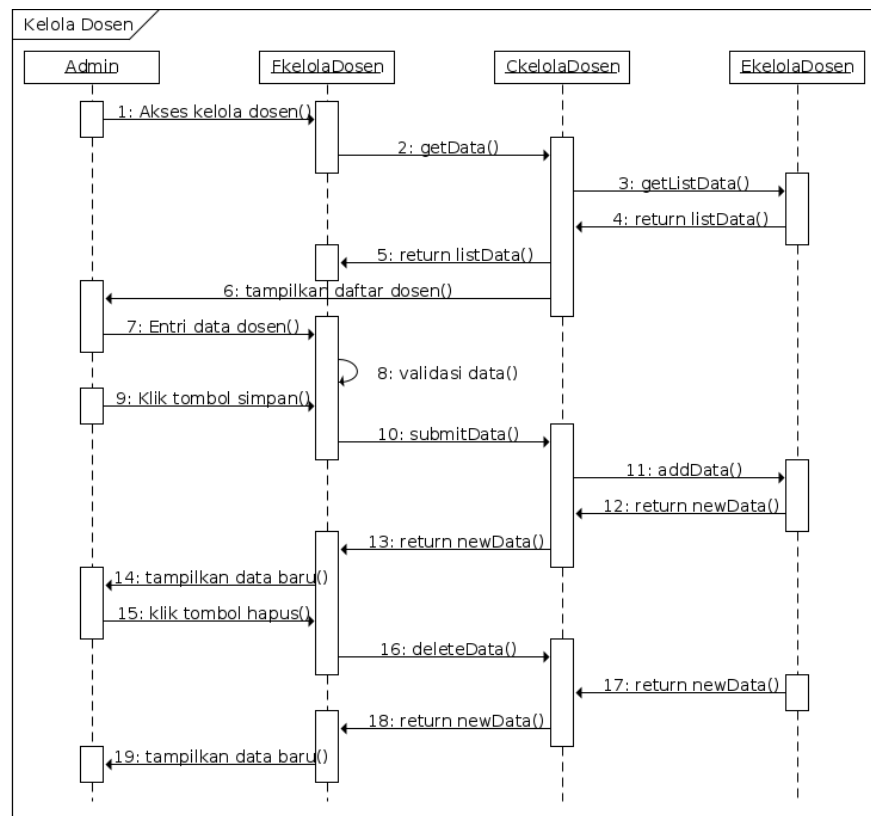
Sequence diagram ini menjelaskan tentang bagaimana aktifitas admin dalam mengelola mahasiswa. Nantinya mahasiswa ini akan mengikuti ujian yang dibuat oleh dosen. Adapun sequence diagram kelola mahasiswa dapat digambarkan seperti Gambar 3.8 :



Gambar 3.8 Sequence Diagram kelola mahasiswa

3.2.1.5.4 Sequence Diagram Kelola Dosen

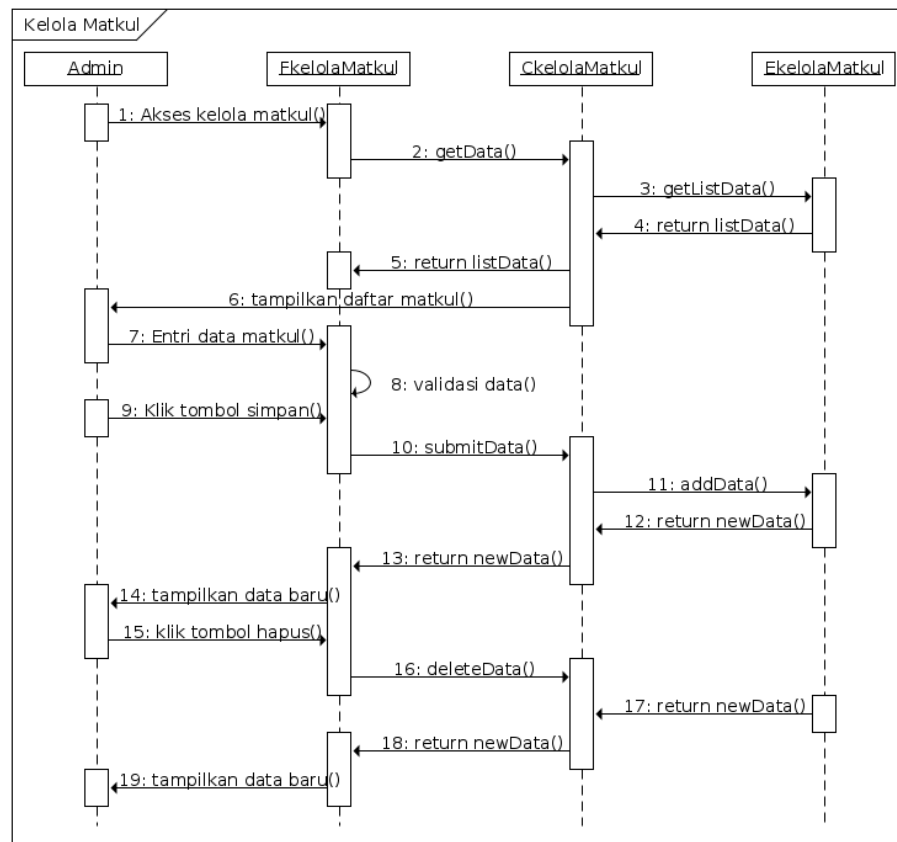
Diagram ini menggambarkan aktifitas admin dalam mengelola data dosen. Dosen yang sudah terdaftar dapat menggunakan aplikasi *CAT* ini. Setiap dosen baru ditambahkan, data dosen tersebut akan bertambah otomatis pada data user sehingga dosen dapat langsung masuk ke sistem. Adapun sequence diagram kelola dosen pada Gambar 3.9 :



Gambar 3.9 Sequence Diagram kelola dosen

3.2.1.5.5 Sequence Diagram Kelola Mata Kuliah

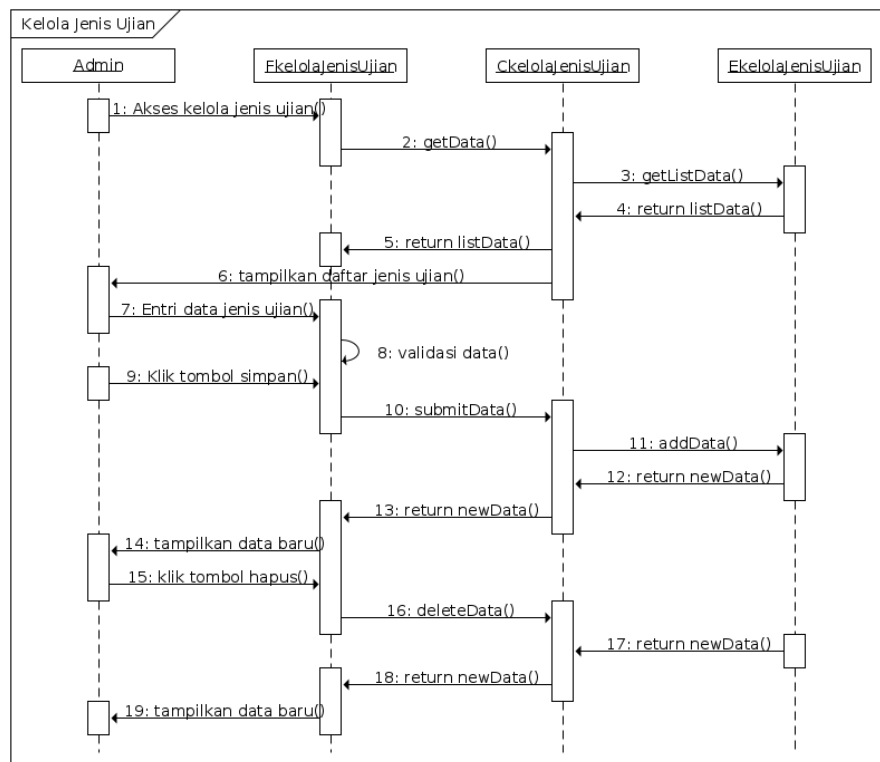
Diagram ini menggambarkan detail proses dari mengelola mata kuliah. Dimulai dari proses pengambilan daftar mata kuliah hingga proses penyimpanan mata kuliah. Adapun *Sequence diagram* kelola mata kuliah dapat dilihat pada Gambar 3.10 :



Gambar3.10 Sequence Diagram kelola mata kuliah

3.2.1.5.6 Sequence Diagram Kelola Jenis Ujian

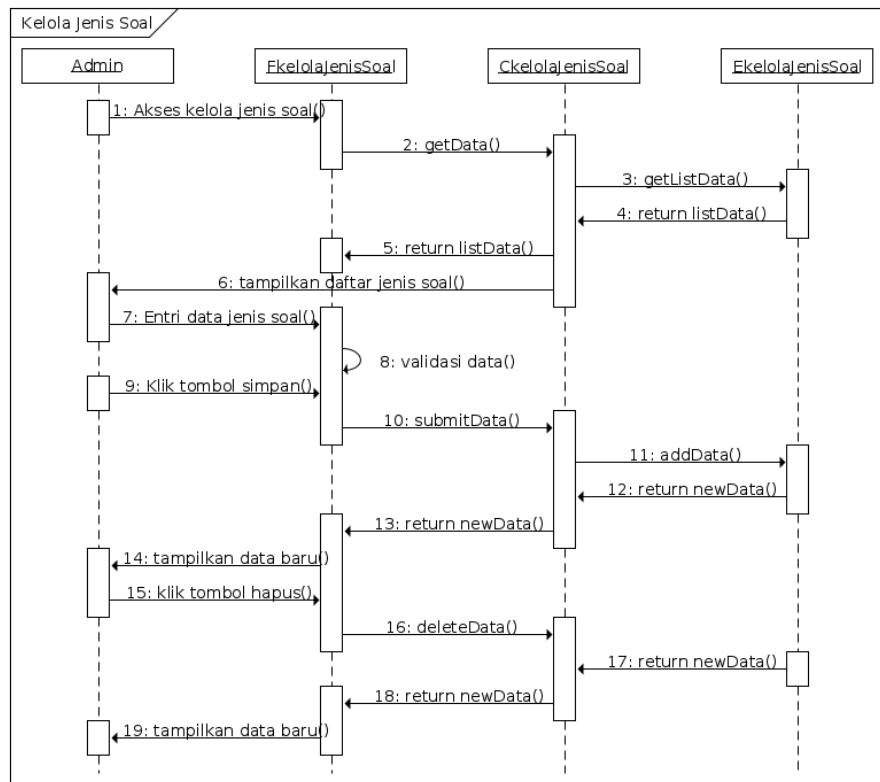
Diagram ini menggambarkan detail proses dari kegiatan mengelola jenis ujian. Dimulai dari proses pengambilan daftar jenis ujian yang sudah ditambahkan hingga proses penyimpanan jenis ujian baru ke basis data. Adapun *Sequence diagram* kelola jenis ujian dapat dilihat pada Gambar 3.11 :



Gambar 3.11 Sequence Diagram kelola jenis ujian

3.2.1.5.7 Sequence Diagram Kelola Tipe Soal

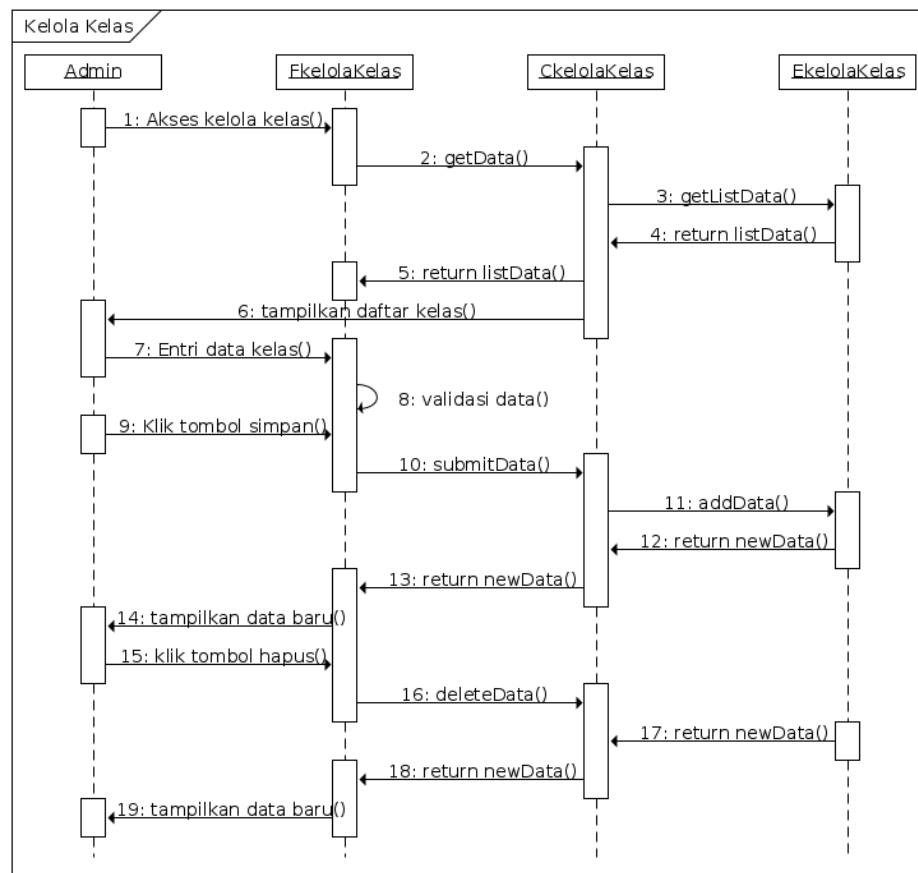
Diagram ini menggambarkan detail proses dari kegiatan mengelola tipe soal. Dimulai dari proses pengambilan daftar tipe soal yang sudah ditambahkan hingga proses penyimpanan tipe soal baru ke basis data. Adapun *Sequence diagram* kelola tipe soal dapat dilihat pada Gambar 3.12 :



Gambar 3.12 Sequence Diagram kelola tipe soal

3.2.1.5.8 Sequence Diagram Kelola Kelas

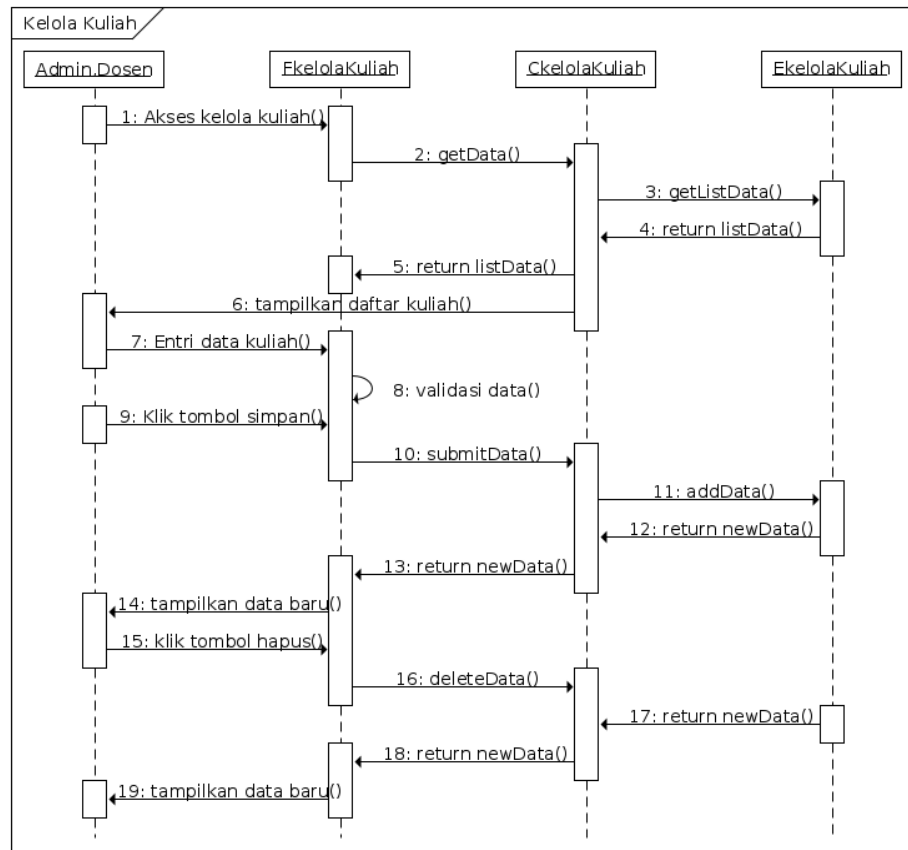
Diagram ini menggambarkan detail proses dari kegiatan mengelola kelas. Dimulai dari proses pengambilan daftar kelas yang sudah ditambahkan hingga proses penyimpanan kelas baru ke basis data. Adapun *Sequence diagram* kelola kelas dapat dilihat pada Gambar 3.13 :



Gambar 3.13 Sequence Diagram kelola kelas

3.2.1.5.9 Sequence Diagram Kelola Kuliah

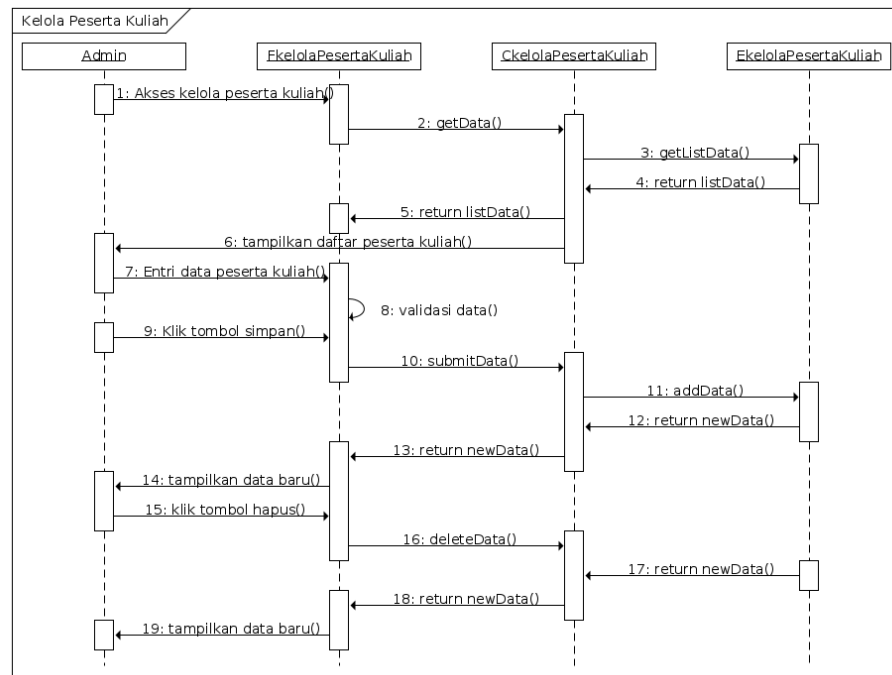
Diagram ini menggambarkan detail proses dari kegiatan mengelola kuliah. Dimulai dari proses pengambilan daftar kuliah yang sudah ditambahkan hingga proses penyimpanan kuliah baru ke basis data. Adapun *Sequence diagram* kelola kuliah dapat dilihat pada Gambar 3.14 :



Gambar 3.14 Sequence Diagram kelola kuliah

3.2.1.5.10 Sequence Diagram Kelola Peserta Kuliah

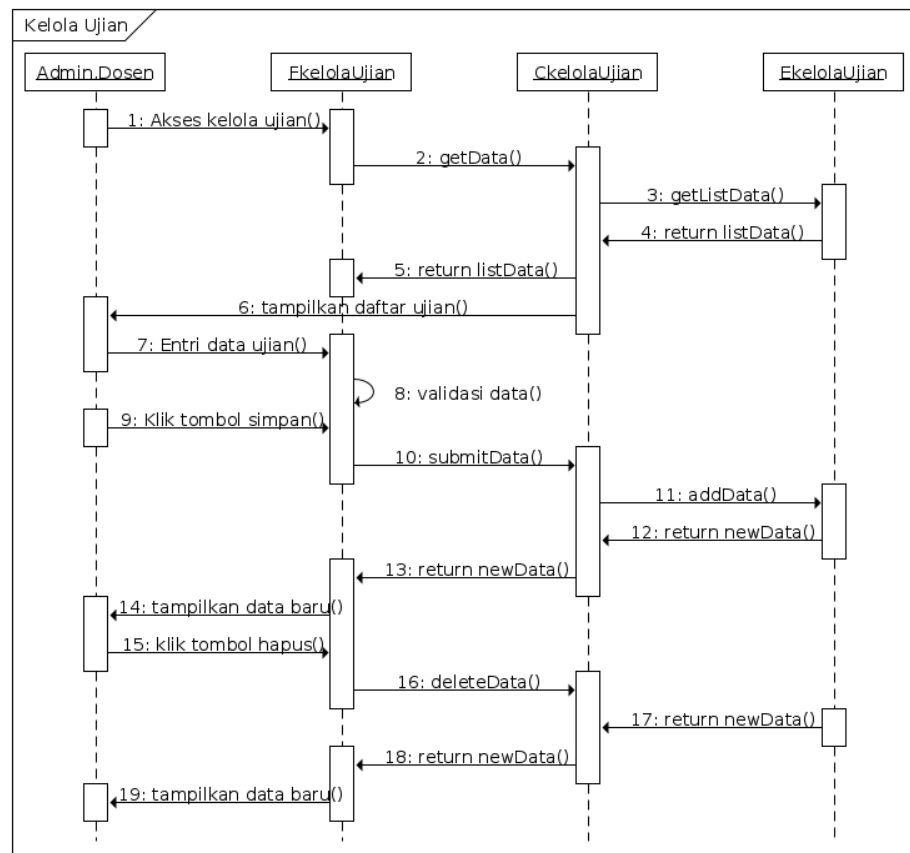
Diagram ini menggambarkan detail proses dari kegiatan mengelola jenis ujian. Dimulai dari proses pengambilan daftar jenis ujian yang sudah ditambahkan hingga proses penyimpanan jenis ujian baru ke basis data. Adapun *Sequence diagram* kelola jenis ujian dapat dilihat pada Gambar 3.15 :



Gambar 3.15 Sequence Diagram kelola peserta kuliah

3.2.1.5.11 Sequence Diagram Kelola Ujian

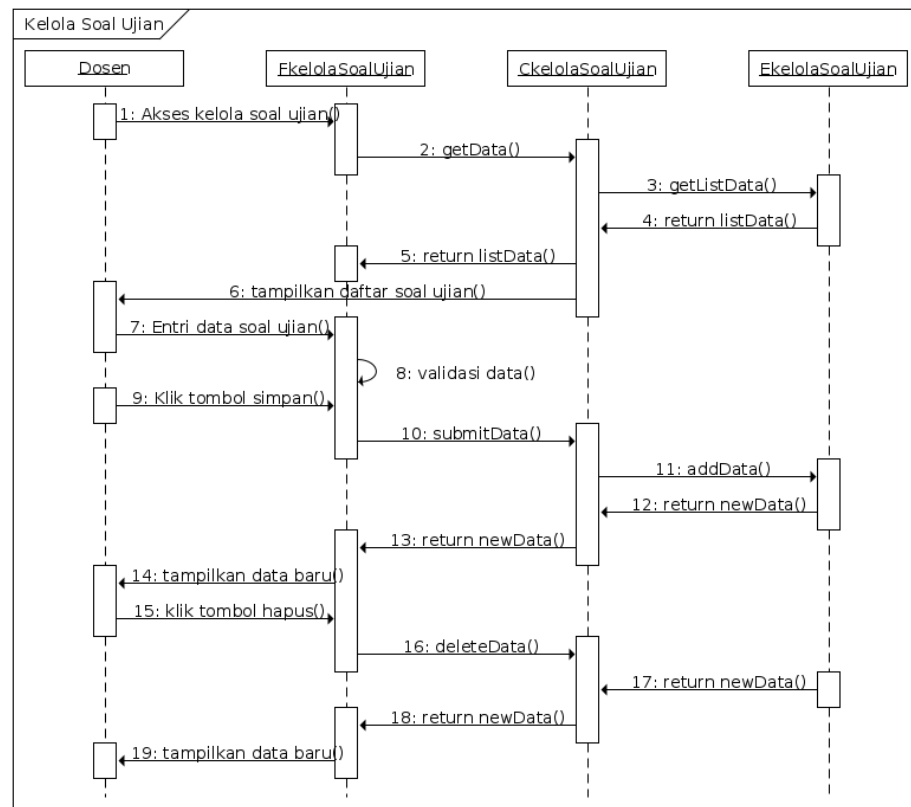
Diagram ini menggambarkan detail proses dari kegiatan mengelola ujian. Dimulai dari proses pengambilan daftar ujian yang sudah ditambahkan hingga proses penyimpanan ujian baru ke basis data. Adapun *Sequence diagram* kelola ujian dapat dilihat pada Gambar 3.16 :



Gambar 3.16 Sequence Diagram kelola ujian

3.2.1.5.12 Sequence Diagram Kelola Soal Ujian

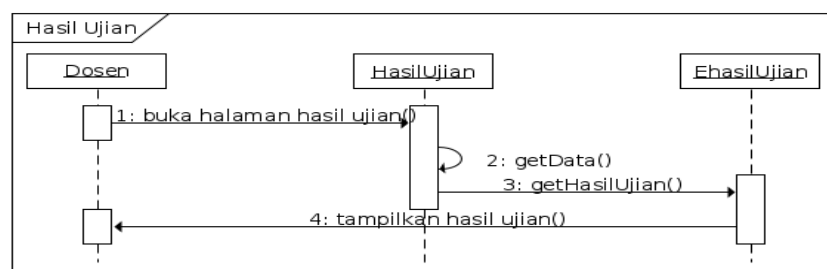
Diagram ini menggambarkan detail proses dari kegiatan mengelola soal ujian. Dimulai dari proses pengambilan daftar soal ujian yang sudah ditambahkan hingga proses penyimpanan soal ujian baru ke basis data. Adapun *Sequence diagram* kelola soal ujian dapat dilihat pada Gambar 3.17 :



Gambar 3.17 Sequence Diagram kelola soal ujian

3.2.1.5.13 Sequence Diagram Hasil Ujian

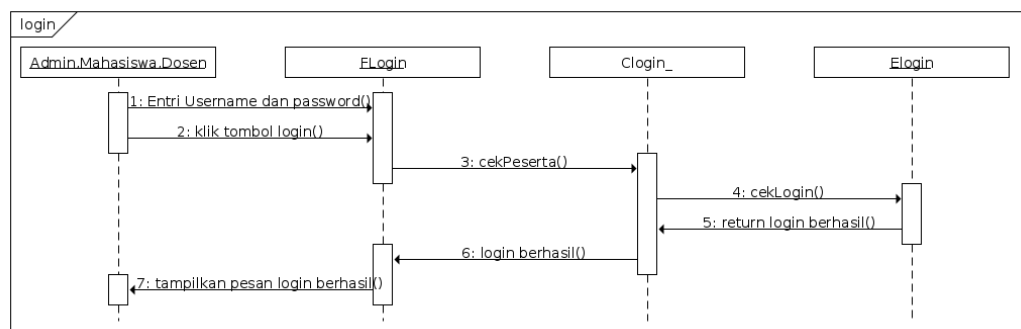
Diagram ini menggambarkan detail proses dari kegiatan mengelola hasil ujian. Dimulai dari proses pengambilan daftar hasil ujian yang sudah ditambahkan hingga proses penyimpanan hasil ujian baru ke basis data. Adapun *Sequence diagram* kelola hasil ujian dapat dilihat pada Gambar 3.18 :



Gambar 3.18 Sequence Diagram hasil ujian

3.2.1.5.14 Sequence Diagram Login

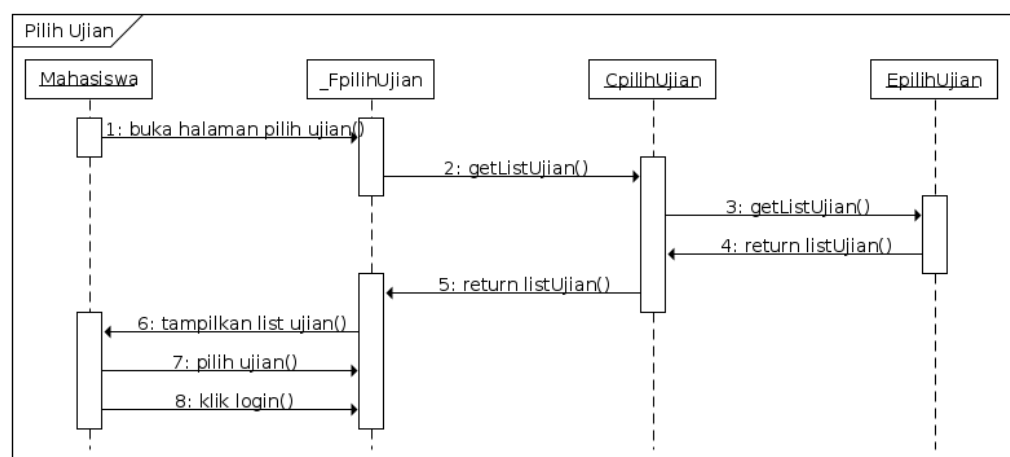
Diagram ini menggambarkan proses seorang pengguna melakukan *login* ke sistem. Dimulai dari mengetikkan *username* dan *password* hingga proses pengecekan hak akses pengguna. Adapun *sequence diagram login* dapat digambarkan seperti Gambar 3.19 :



Gambar 3.19 Sequence Diagram login

3.2.1.5.15 Sequence Diagram Pilih Ujian

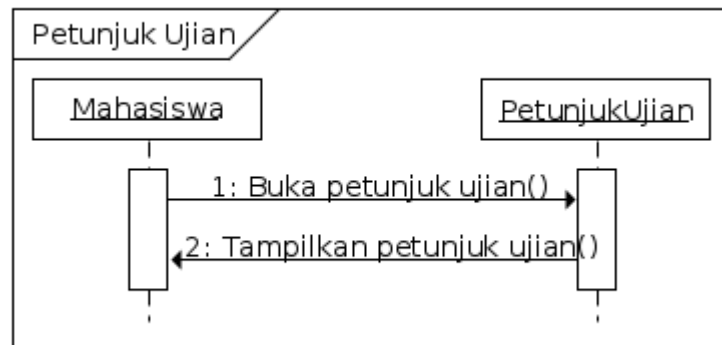
Diagram ini menggambarkan proses mahasiswa dalam memilih ujian yang ingin di ikuti. Adapun *sequence diagram pilih ujian* dapat digambarkan seperti Gambar 3.20 :



Gambar 3.20 Sequence Diagram pilih ujian

3.2.1.5.16 Sequence Diagram Petunjuk Ujian

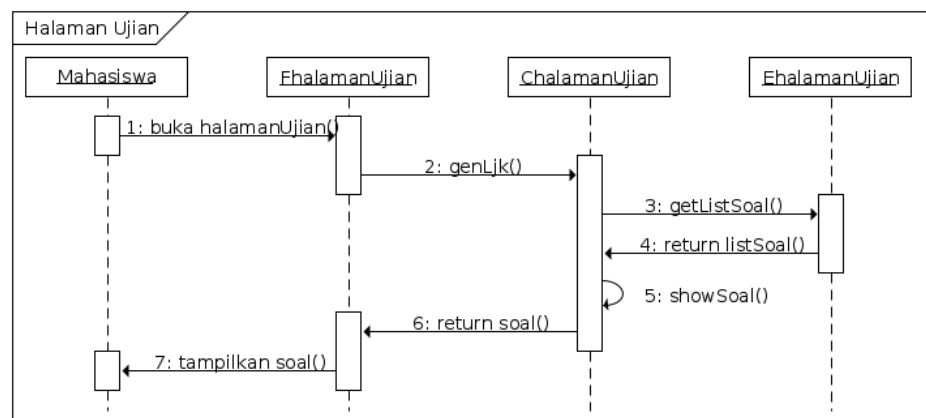
Diagram ini menggambarkan proses seorang mahasiswa dalam mengakses halaman petunjuk ujian. Adapun *sequence diagram* petunjuk ujian dapat digambarkan seperti Gambar 3.21 :



Gambar 3.21 Sequence Diagram petunjuk ujian

3.2.1.5.17 Sequence Diagram Halaman Ujian

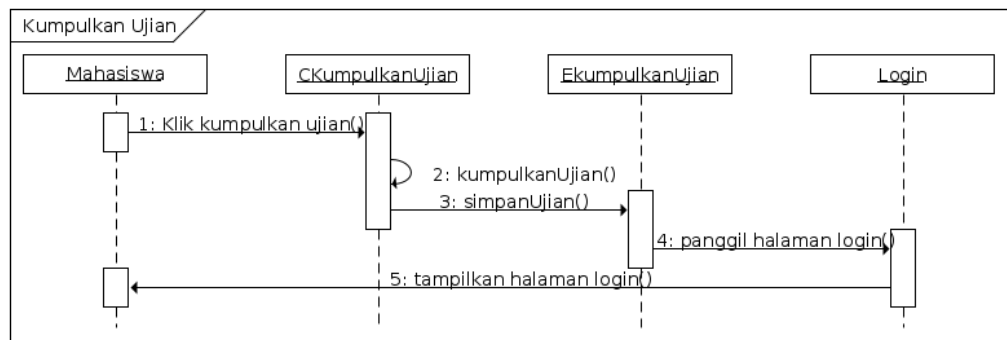
Diagram ini menggambarkan proses seorang mahasiswa dalam mengerjakan ujian pada aplikasi yang akan dikembangkan. Adapun *sequence diagram* halaman ujian dapat digambarkan seperti Gambar 3.22 :



Gambar 3.22 Sequence Diagram halaman ujian

3.2.1.5.18 Sequence Diagram Kumpulkan Ujian

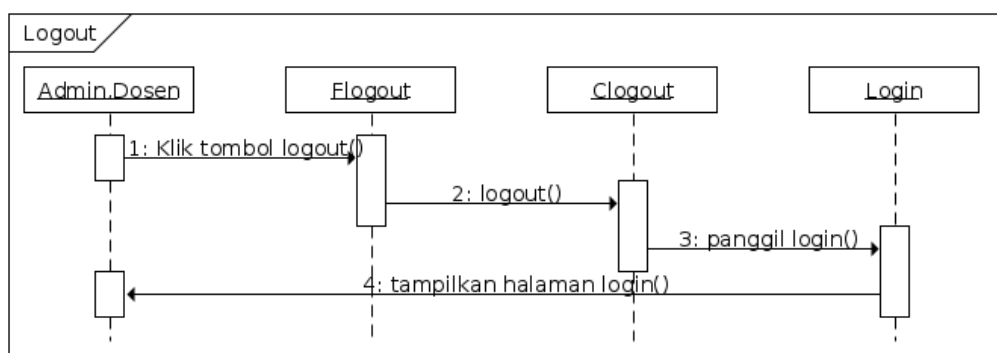
Diagram ini menggambarkan proses seorang mahasiswa pada saat mengumpulkan ujian. Adapun *sequence diagram* kumpulkan ujian dapat digambarkan seperti Gambar 3.23 :



Gambar 3.23 Sequence Diagram kumpulkan ujian

3.2.1.5.19 Sequence Diagram Logout

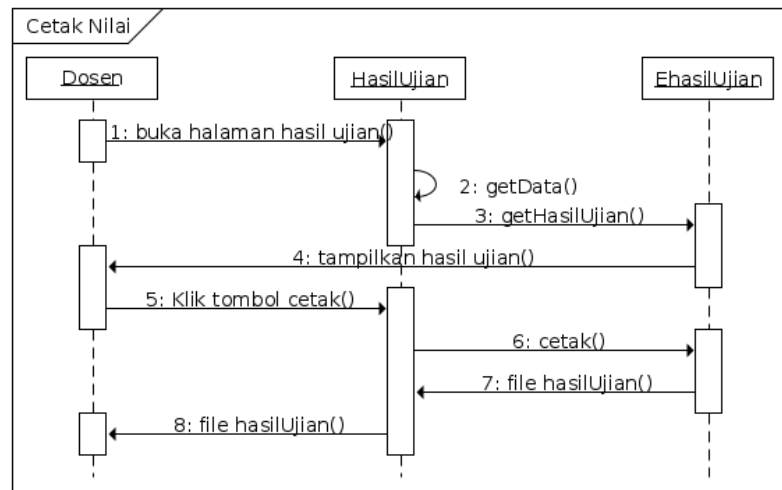
Diagram ini menggambarkan proses pengguna dalam melakukan *logout* atau keluar dari sistem. Adapun *sequence diagram* *logout* dapat digambarkan seperti Gambar 3.24 :



Gambar 3.24 Sequence Diagram logout

3.2.1.5.20 Sequence Diagram Cetak Nilai

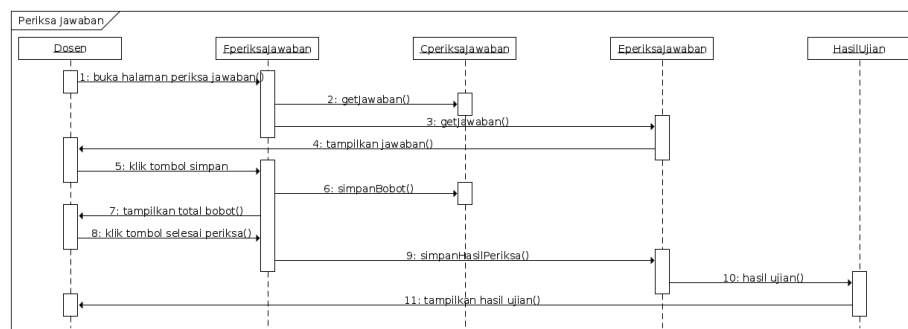
Diagram ini menggambarkan proses seorang dosen dalam mengakses nilai ujian mahasiswa serta mencetaknya ke dalam bentuk *file*. Adapun *sequence diagram* cetak nilai dapat digambarkan seperti Gambar 3.25 :



Gambar 3.25 Sequence Diagram cetak nilai

3.2.1.5.21 Sequence Diagram Periksa Jawaban

Diagram ini menunjukkan proses seorang dosen dalam melakukan pemeriksaan jawaban peserta ujian dengan tipe soal esai. Adapun *sequence diagram* periksa ujian dapat digambarkan seperti Gambar 3.21 :



Gambar 3.26 Sequence Diagram periksa jawaban

3.2.2 Desain Terinci

Desain terinci merupakan pengembangan lebih lanjut dari desain global.

Desain terinci terbagi atas 3 macam yaitu desain *output*, *input* dan *file*.

3.2.2.1 Desain Output

Desain *output* merupakan desain atau tampilan dari hasil proses sistem.

Output dapat ditampilkan pada monitor atau dicetak kedalam bentuk kertas.

Desain *output* dari aplikasi yang akan dikembangkan dapat dijelaskan sebagai berikut :

1. Daftar Mahasiswa

Halaman ini digunakan untuk melihat daftar mahasiswa secara keseluruhan. Adapun tampilan dari halaman mahasiswa dapat digambarkan seperti Gambar 3.27 :

Header				
Username	Daftar Mahasiswa			
	Tambah Data		Refresh	
	Pencarian ...		Reset	
Menu	No	NOBP	Nama Mahasiswa	Aksi
	99	Varchar(15)	Varchar(100)	
	99	Varchar(15)	Varchar(100)	
Sebelumnya 99 Selanjutnya Banyak Data Perhalaman 99				

Gambar 3.27 Halaman Daftar Mahasiswa

2. Daftar Dosen

Halaman ini berisi daftar dosen secara keseluruhan. Adapun tampilan dari halaman dosen dapat digambarkan seperti Gambar 3.28 :

Header															
Username	Daftar Dosen														
	<input type="button" value="Tambah Data"/> <input type="button" value="Refresh"/>		<input type="button" value="Pencarian ..."/> <input type="button" value="Reset"/>												
	<table border="1"> <thead> <tr> <th>No</th> <th>NIDN</th> <th>Nama Dosen</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td>99</td> <td>Varchar(15)</td> <td>Varchar(100)</td> <td></td> </tr> <tr> <td>99</td> <td>Varchar(15)</td> <td>Varchar(100)</td> <td></td> </tr> </tbody> </table>				No	NIDN	Nama Dosen	Aksi	99	Varchar(15)	Varchar(100)		99	Varchar(15)	Varchar(100)
No	NIDN	Nama Dosen	Aksi												
99	Varchar(15)	Varchar(100)													
99	Varchar(15)	Varchar(100)													
Menu	<input type="button" value="Sebelumnya"/> <input type="button" value="99"/> <input type="button" value="Selanjutnya"/>		Banyak Data Perhalaman <input type="button" value="99"/>												

Gambar 3.28 Halaman Daftar Dosen

3. Daftar Mata Kuliah

Halaman ini berisi daftar mata kuliah secara keseluruhan. Adapun tampilan dari halaman mata kuliah dapat digambarkan seperti Gambar 3.29 :

Header																						
Username	Daftar Matkul																					
	<input type="button" value="Tambah Data"/> <input type="button" value="Refresh"/>		<input type="button" value="Pencarian ..."/> <input type="button" value="Reset"/>																			
	<table border="1"> <thead> <tr> <th>No</th> <th>Kode Matkul</th> <th>Nama Matkul</th> <th>Semester</th> <th>SKS</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td>99</td> <td>Varchar(15)</td> <td>Varchar(100)</td> <td>Int(2)</td> <td>Int(2)</td> <td></td> </tr> <tr> <td>99</td> <td>Varchar(15)</td> <td>Varchar(100)</td> <td>Int(2)</td> <td>Int(2)</td> <td></td> </tr> </tbody> </table>					No	Kode Matkul	Nama Matkul	Semester	SKS	Aksi	99	Varchar(15)	Varchar(100)	Int(2)	Int(2)		99	Varchar(15)	Varchar(100)	Int(2)	Int(2)
No	Kode Matkul	Nama Matkul	Semester	SKS	Aksi																	
99	Varchar(15)	Varchar(100)	Int(2)	Int(2)																		
99	Varchar(15)	Varchar(100)	Int(2)	Int(2)																		
Menu	<input type="button" value="Sebelumnya"/> <input type="button" value="99"/> <input type="button" value="Selanjutnya"/>			Banyak Data Perhalaman <input type="button" value="99"/>																		

Gambar 3.29 Halaman Daftar Mata Kuliah

4. Daftar Kelas

Halaman ini berisi daftar kelas secara keseluruhan. Adapun tampilan dari halaman kelas dapat digambarkan seperti Gambar 3.30 :

Header											
Username	Daftar Kelas										
	<input type="button" value="Tambah Data"/> <input type="button" value="Refresh"/> <input type="button" value="Pencarian ..."/> <input type="button" value="Reset"/>										
	<table border="1"> <thead> <tr> <th>No</th> <th>Nama Kelas</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td>99</td> <td>Varchar(10)</td> <td></td> </tr> <tr> <td>99</td> <td>Varchar(10)</td> <td></td> </tr> </tbody> </table>			No	Nama Kelas	Aksi	99	Varchar(10)		99	Varchar(10)
No	Nama Kelas	Aksi									
99	Varchar(10)										
99	Varchar(10)										
Menu	<input type="button" value="Sebelumnya"/> <input type="button" value="99"/> <input type="button" value="Selanjutnya"/>		Banyak Data Perhalaman <input type="button" value="99"/>								

Gambar 3.30 Halaman Daftar Kelas

5. Daftar Tipe Soal

Halaman ini berisi daftar tipe soal secara keseluruhan. Adapun tampilan dari halaman tipe soal dapat digambarkan seperti Gambar 3.31 :

Header											
Username	Daftar Tipe Soal										
	<input type="button" value="Tambah Data"/> <input type="button" value="Refresh"/> <input type="button" value="Pencarian ..."/> <input type="button" value="Reset"/>										
	<table border="1"> <thead> <tr> <th>No</th> <th>Nama Tipe Soal</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td>99</td> <td>Varchar(20)</td> <td></td> </tr> <tr> <td>99</td> <td>Varchar(20)</td> <td></td> </tr> </tbody> </table>			No	Nama Tipe Soal	Aksi	99	Varchar(20)		99	Varchar(20)
No	Nama Tipe Soal	Aksi									
99	Varchar(20)										
99	Varchar(20)										
Menu	<input type="button" value="Sebelumnya"/> <input type="button" value="99"/> <input type="button" value="Selanjutnya"/>		Banyak Data Perhalaman <input type="button" value="99"/>								

Gambar 3.31 Halaman Daftar Tipe Soal

6. Daftar Jenis Ujian

Halaman ini berisi daftar jenis ujian secara keseluruhan. Adapun tampilan dari halaman jenis ujian dapat digambarkan seperti Gambar 3.32 :

Header											
Username	Daftar Jenis Ujian										
	<div>Tambah Data Refresh Pencarian ... Reset</div>										
	<table border="1"> <thead> <tr> <th>No</th> <th>Nama Jenis Ujian</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td>99</td> <td>Varchar(10)</td> <td></td> </tr> <tr> <td>99</td> <td>Varchar(10)</td> <td></td> </tr> </tbody> </table>			No	Nama Jenis Ujian	Aksi	99	Varchar(10)		99	Varchar(10)
No	Nama Jenis Ujian	Aksi									
99	Varchar(10)										
99	Varchar(10)										
Menu	<div>Sebelumnya 99 Selanjutnya</div>										
	<div>Banyak Data Perhalaman 99</div>										

Gambar 3.32 Halaman Daftar Jenis Ujian

7. Daftar User

Halaman ini berisi daftar user secara keseluruhan. Adapun tampilan dari halaman user dapat digambarkan seperti Gambar 3.33 :

Header														
Username	Daftar User													
	<div>Tambah Data Refresh Pencarian ... Reset</div>													
	<table border="1"> <thead> <tr> <th>No</th> <th>Username</th> <th>Jenis User</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td>99</td> <td>Varchar(30)</td> <td>Varchar(10)</td> <td></td> </tr> <tr> <td>99</td> <td>Varchar(30)</td> <td>Varchar(10)</td> <td></td> </tr> </tbody> </table>			No	Username	Jenis User	Aksi	99	Varchar(30)	Varchar(10)		99	Varchar(30)	Varchar(10)
No	Username	Jenis User	Aksi											
99	Varchar(30)	Varchar(10)												
99	Varchar(30)	Varchar(10)												
Menu	<div>Sebelumnya 99 Selanjutnya</div>													
	<div>Banyak Data Perhalaman 99</div>													

Gambar 3.33 Halaman Daftar User

8. Daftar Kuliah

Halaman ini berisi daftar kuliah yang telah ditambahkan oleh dosen. Kuliah ini nantinya akan didaftarkan pada ujian. Adapun tampilan dari halaman kuliah dapat digambarkan seperti Gambar 3.34 :

Header					
Username	Daftar Kuliah				
	<input type="button" value="Tambah Data"/> <input type="button" value="Refresh"/>		<input type="button" value="Pencarian ..."/> <input type="button" value="Reset"/>		
	No	Mata Kuliah	Kelas	Tahun Akademik	Status
	99	Varchar(100)	Varchar(10)	Varchar(5)	Varchar(10)
Menu					
	99	Varchar(100)	Varchar(10)	Varchar(5)	Varchar(10)
<input type="button" value="Sebelumnya"/> <input type="button" value="99"/> <input type="button" value="Selanjutnya"/>		Banyak Data Perhalaman <input type="button" value="99"/>			

Gambar 3.34 Halaman Daftar Kuliah

9. Daftar Peserta Kuliah

Halaman ini berisi daftar peserta kuliah yang telah ditambahkan oleh dosen. Peserta kuliah ini nantinya akan mengikuti ujian yang telah dibuat dosen. Adapun tampilan dari halaman peserta kuliah dapat digambarkan seperti Gambar 3.35 :

Header					
Username	Informasi Kuliah				
	Nama Dosen		Varchar(100)		
	Mata Kuliah		Varchar(100)		
	Kelas		Varchar(10)		
Menu	Tahun Akademik		Varchar(5)		
	Daftar Peserta Kuliah				
	<input type="button" value="Tambah Data"/> <input type="button" value="Refresh"/>		<input type="button" value="Pencarian ..."/> <input type="button" value="Reset"/>		
	No	Mata Kuliah	Kelas	Tahun Akademik	Status
99	Varchar(100)	Varchar(10)	Varchar(5)	Varchar(10)	
<input type="button" value="Sebelumnya"/> <input type="button" value="99"/> <input type="button" value="Selanjutnya"/>		Banyak Data Perhalaman <input type="button" value="99"/>			

Gambar 3.35 Halaman Daftar Peserta Kuliah

10. Daftar Ujian

Halaman ini berisi daftar ujian yang telah ditambahkan oleh dosen. Peserta kuliah nantinya akan mengikuti ujian yang telah dibuat dosen. Adapun tampilan dari halaman daftar ujian dapat digambarkan seperti Gambar 3.36 :

Header						
Username	Daftar Ujian					
	Tambah Data		Refresh		Pencarian ...	
					Reset	
Menu	No	Matkul	Jenis Ujian	Waktu	Status	TA
	99	Varchar(100)	Varchar(10)	Date	Varchar(15)	Varchar(5)
	99	Varchar(100)	Varchar(10)	Date	Varchar(15)	Varchar(5)
		Sebelumnya		99	Selanjutnya	
				Banyak Data Perhalaman		99

Gambar 3.36 Halaman Daftar Ujian

11. Daftar Soal Ujian

Halaman ini berisi daftar soal ujian yang telah ditambahkan oleh dosen. Soal ini nantinya akan keluar pada saat ujian secara acak. Adapun tampilan dari halaman soal ujian dapat digambarkan seperti Gambar 3.37 :

Header							
Username	Informasi Ujian						
	Mata Kuliah	Varchar(100)	Dosen	Varchar(100)			
	Kelas	Varchar(10)	Tahun Akademik	Varchar(5)			
	Hari	Varchar(15)	Waktu	Date			
Menu	Jenis Ujian	Varchar(5)	Jenis Soal	Varchar(5)			
	Soal Ujian		Hasil Ujian				
	Tambah Data		Refresh		Pencarian ...		Reset
	No	Isi Soal	Bobot	Aksi			
	99	Text	Int(5)				
	99	Text	Int(5)				
		Sebelumnya		99	Selanjutnya		Banyak Data Perhalaman
							99

Gambar 3.37 Halaman Daftar Soal Ujian

12. Daftar Hasil Ujian

Halaman ini berisi daftar hasil ujian peserta. Status ujian beserta nilai ujian mahasiswa akan ditampilkan di halaman ini. Adapun tampilan dari halaman daftar hasil ujian dapat digambarkan seperti Gambar 3.38 :

Header						
Username	Informasi Ujian					
Menu	Mata Kuliah	Varchar(100)	Dosen	Varchar(100)		
	Kelas	Varchar(10)	Tahun Akademik	Varchar(5)		
	Hari	Varchar(15)	Waktu	Date		
	Jenis Ujian	Varchar(5)	Jenis Soal	Varchar(5)		
Soal Ujian		Hasil Ujian				
Tambah Data		Refresh		Pencarian ...		Reset
No	NOBP	Nama	Kelas	Status	Nilai	Aksi
99	Varchar(15)	Varchar(100)	Varchar(5)	Varchar(15)	Int(3)	
99	Varchar(15)	Varchar(100)	Varchar(5)	Varchar(15)	Int(3)	
Sebelumnya		99	Selanjutnya		Banyak Data Perhalaman 99	

Gambar 3.38 Halaman Daftar Hasil Ujian

13. Laporan Hasil Ujian

Laporan isi berisi daftar mahasiswa serta nilai ujian yang didapatkannya. Pada laporan terdapat ketentuan, jika mahasiswa belum melaksanakan ujian maka nilainya akan menjadi 0 dan statusnya menjadi “Belum Ujian”. Sedangkan jika mahasiswa sudah ujian dan jawaban mahasiswa sudah diperiksa (tipe soal essai) maka status akan berubah menjadi “Sudah Ujian” dan pada kolom nilai akan keluar nilai sesuai dengan hasil ujian mahasiswa tersebut. Adapun tampilan dari laporan hasil ujian dapat dilihat pada Gambar 3.39 :

LAPORAN HASIL UJIAN				
Mata Kuliah	Varchar(100)			
Dosen	Varchar(100)			
Kelas	Varchar(30)			
Tahun Akademik	Varchar(5)			
Hari	Date			
Jam	Time			
Jenis Ujian	Varchar(10)			
Jenis Soal	Varchar(10)			

NOBP	Nama	Kelas	Status	Nilai
Varchar(15)	Varchar(100)	Varchar(10)	Varchar(30)	Int(3)
Varchar(15)	Varchar(100)	Varchar(10)	Varchar(30)	Int(3)

Gambar 3.39 Laporan Hasil Ujian

3.2.2.2 Desain *Input*

Dalam setiap sistem, pasti ada masukkan atau *input* yang nantinya akan diproses sistem. Dari proses *input* tadi, didapat hasil *output* seperti yang telah dijelaskan sebelumnya. Berikut adalah desain *input* yang akan dikembangkan :

1. Form Mahasiswa

Form ini digunakan untuk menambahkan mahasiswa baru kedalam sistem. Pengguna harus memasukkan NOBP dan nama kedalam form yang telah disediakan. Adapun desain dari form mahasiswa dapat digambarkan seperti Gambar 3.39 :

Header	
Username	Tambah Data NOBP <input type="text" value="Varchar(15)"/> Nama <input type="text" value="Varchar(100)"/> <input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/>
Menu	

Gambar 3.40 Form Mahasiswa

2. Form Dosen

Form ini digunakan untuk menambahkan dosen baru kedalam sistem. Pengguna harus memasukkan NIDN dan nama kedalam form yang telah disediakan. Adapun desain dari form dosen dapat digambarkan seperti Gambar 3.40 :

Header	
Username	Tambah Data NIDN <input type="text" value="Varchar(15)"/> Nama <input type="text" value="Varchar(100)"/> <input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/>
Menu	

Gambar 3.41 Form Mahasiswa

3. Form Matkul

Form ini digunakan untuk menambahkan mata kuliah baru kedalam sistem. Pengguna harus memasukkan kode mata kuliah, nama mata

kuliah, jumlah SKS dan semester kedalam form yang telah disediakan. Adapun desain dari form matkul dapat digambarkan seperti Gambar 3.41 :

Header	
Username	Tambah Data Kode Matkul <input type="text" value="Varchar(15)"/> Nama Matkul <input type="text" value="Varchar(100)"/> SKS <input type="text" value="Int(3)"/> Semester <input type="text" value="Int(3)"/> <input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/>
Menu	

Gambar 3.42 Form Matkul

4. Form Kelas

Form ini digunakan untuk menambahkan nama kelas baru. Adapun desain dari form kelas dapat digambarkan seperti Gambar 3.42 :

Header	
Username	Tambah Data Nama Kelas <input type="text" value="Varchar(10)"/> <input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/>
Menu	

Gambar 3.43 Form Kelas

5. Form Tipe Soal

Form ini digunakan untuk menambahkan tipe soal baru jika diperlukan. Adapun desain dari form tipe soal dapat digambarkan seperti Gambar 3.43 :

Header	
Username	Tambah Data Nama Tipe Soal <input type="text" value="Varchar(20)"/> <input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/>
Menu	

Gambar 3.44 Form Tipe Soal

6. Form Jenis Ujian

Form ini digunakan untuk menambahkan jenis ujian baru jika diperlukan. Adapun desain dari form jenis ujian dapat digambarkan seperti Gambar 3.44 :

Header	
Username	Tambah Data Nama Jenis Ujian <input type="text" value="Varchar(10)"/> <input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/>
Menu	

Gambar 3.45 Form Jenis Ujian

7. Form User

Form ini digunakan untuk menambahkan *user* baru. *User* baru dapat berupa admin baru, dosen baru atau mahasiswa baru. Adapun desain dari form user dapat digambarkan seperti Gambar 3.46 :

Header	
Username	Tambah Data Username <input type="text" value="Varchar(30)"/> Password <input type="text" value="Varchar(12)"/> Jenis User <input type="text" value="Varchar(10)"/>
Menu	<input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/>

Gambar 3.46 Form User

8. Form Kuliah


Form ini digunakan oleh dosen untuk menambahkan kuliah yang diampu. Kuliah yang sudah ditambahkan nantinya akan muncul di form tambah ujian. Pada form ini, dosen diminta untuk memilih mata kuliah, memilih kelas-kelas yang diampu serta mengentri tahun akademik. Adapun desain dari form kuliah dapat dilihat pada Gambar 3.47 :

Header	
Username	Tambah Data Pilih Mata Kuliah <input type="text" value="Varchar(100)"/>
Menu	Tambahkan Kelas <input type="text" value="Varchar(12)"/> Tahun Akademik <input type="text" value="Varchar(5)"/>
	<input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/>

Gambar 3.47 Form Kuliah

9. Form Peserta Kuliah

Form ini digunakan oleh dosen untuk menambah peserta kuliah. Dosen diminta untuk memilih salah satu kelas, kemudian memilih mahasiswa yang ingin ditambahkan. Adapun desain form peserta kuliah dapat dilihat pada Gambar 3.48 :

Header	
Username	Tambah Data Pilih Kelas <input type="text" value="Varchar(10)"/>  Tambahkan Mahasiswa <input type="text" value="Varchar(100)"/> <input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/>
Menu	

Gambar 3.48 Form Peserta Kuliah

10. Form Ujian

Form ini digunakan oleh dosen untuk menambahkan ujian baru. Dosen diminta untuk memilih kuliah yang ingin diujikan, jenis soal ujian, jenis ujian, jadwal ujian serta deskripsi ujian jika perlu. Adapun desain form ujian dapat dilihat pada Gambar 3.49 :

Header	
Username	Tambah Data Pilih Kuliah <input type="text" value="Varchar(100)"/>
Menu	Pilih Jenis Ujian <input type="text" value="Varchar(10)"/>
	Pilih Jenis Soal <input type="text" value="Varchar(15)"/>
	Hari Ujian <input type="text" value="Date"/>
	Jam Mulai <input type="text" value="Time"/>
	Jam Selesai <input type="text" value="Time"/>
	Deskripsi <input type="text" value="Varchar(100)"/>
<input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/>	

Gambar 3.49 Form Ujian

11. Form Soal Ujian

Form ini digunakan oleh dosen untuk menambahkan soal ujian baru. Form soal ujian memiliki 2 tampilan tergantung jenis soal ujian yang dipilih. Adapun desain form soal ujian dengan jenis soal ujian essai dapat dilihat pada Gambar 3.50 :

Header	
Username	Tambah Data Isi Soal <input type="text" value="Text"/>
Menu	Bobot Nilai <input type="text" value="Integer(3)"/>
	<input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/>

Gambar 3.50 Form Soal Ujian Essai

Pada jenis soal ujian essai, dosen hanya diminta untuk memasukkan isi soal serta bobot soal. Namun pada soal ujian pilihan ganda, dosen diminta untuk memasukkan pilihan ganda yang disediakan dan bobot nilai untuk tipe soal pilihan ganda otomatis diatur menjadi 1 (satu). Adapun desain form soal ujian dengan tipe soal pilihan ganda dapat dilihat pada Gambar 3.51 :

Header	
Username	Tambah Data Isi Soal <input type="text" value="Text"/> <input type="button" value="A"/> <input type="text" value="Varchar(200)"/> <input type="button" value="B"/> <input type="text" value="Varchar(200)"/> <input type="button" value="C"/> <input type="text" value="Varchar(200)"/> Jawaban <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/>
Menu	

Gambar 3.51 Form Soal Ujian Pilihan Ganda

12. Form Halaman Ujian

Form ini digunakan oleh mahasiswa untuk menjawab soal ujian. Form halaman ujian memiliki 2 desain untuk tipe soal essai dan untuk tipe soal pilihan ganda. Adapun desain form halaman ujian dengan jenis soal ujian essai dapat dilihat pada Gambar 3.52 :

Header	
Pilih Soal <div>99 99 99</div>	<div>Total Soal : 99 Sisa Waktu : Date</div> <div>Selesai : 99 Belum Selesai : 99</div> <p><i>Jawablah soal dibawah dengan benar!</i></p> <p>Isi Soal</p> <p>Text</p> <p>Jawaban Anda</p> <div>Text</div> <div>Simpan Jawaban</div> <div><- -></div>
Informasi Ujian Varchar(100) Varchar(100) Date Time <div>Kumpulkan Ujian</div>	

Gambar 3.52 Form Halaman Ujian Essai

Pada jenis soal ujian essai, mahasiswa diminta untuk mengetikkan jawabannya secara lengkap pada form yang telah disediakan. Dan jika sudah selesai, peserta wajib mengklik tombol “Simpan Jawaban” agar jawaban tersimpan. Pada form halaman ujian dengan tipe soal pilihan ganda, mahasiswa diminta untuk memilih jawaban yang dianggap benar. Dan jawaban peserta akan tersimpan otomatis. Adapun desain form halaman ujian dengan jenis soal ujian essai dapat dilihat pada Gambar 3.53 :

Header	
<p>Pilih Soal</p> <div>99 99 99</div>	<div> <div>Total Soal : 99</div> <div>Sisa Waktu : Date</div> <div>Selesai : 99</div> <div>Belum Selesai : 99</div> </div> <p><i>Jawablah soal dibawah dengan benar!</i></p> <p>Isi Soal</p> <p>Text</p> <p><input type="radio"/> Text</p> <p><input type="radio"/> Text</p> <p><input type="radio"/> Text</p> <div> <div><-</div> <div>-></div> </div>
<p>Informasi Ujian</p> <p>Varchar(100)</p> <p>Varchar(100)</p> <p>Date</p> <p>Time</p> <div>Kumpulkan Ujian</div>	

Gambar 3.53 Form Halaman Ujian Pilihan Ganda