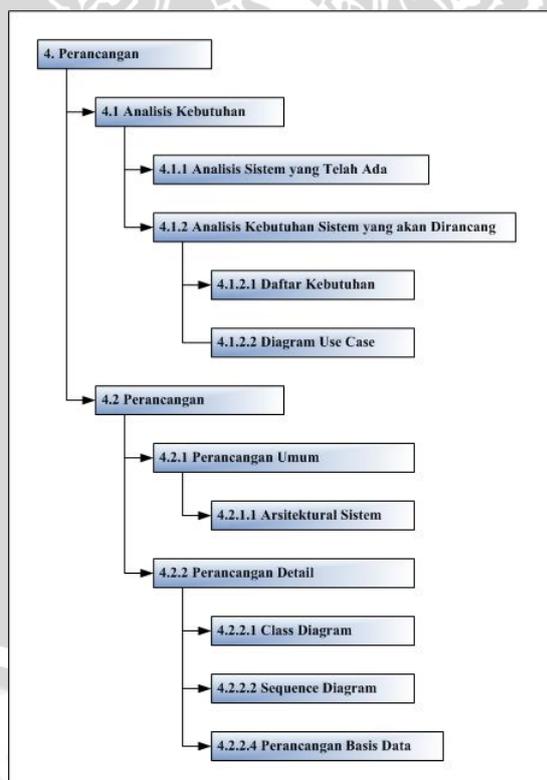


## BAB IV PERANCANGAN

Pada skripsi ini dirancang suatu aplikasi Sistem Manajemen Laboratorium Teknik Elektro Universitas Brawijaya dengan menggunakan pendekatan berorientasi obyek. Perancangan yang dilakukan meliputi dua tahap yaitu analisis berorientasi obyek dan desain berorientasi obyek. Pada tahap analisis berorientasi obyek akan dilakukan analisis terhadap sistem eksisting atau sistem yang telah ada dan analisis kebutuhan sistem yang akan dirancang dengan membuat daftar kebutuhan dan kemudian dimodelkan dengan *use case diagram* untuk menggambarkan kebutuhan tersebut. Tahap desain berorientasi obyek terdiri dari perancangan umum dan perancangan detail. Perancangan umum menggambarkan arsitektural sistem keseluruhan. Perancangan detail terdiri dari perancangan *class diagram* dan *sequence diagram* sebagai pemodelan perangkat lunak dan perancangan basis data.



Gambar 4.1 Diagram Pohon Perancangan

Sumber : [Perancangan]

#### 4.1. Analisis Berorientasi Obyek

Pengembangan sebuah perangkat lunak bertujuan untuk menghasilkan perangkat lunak yang dapat memenuhi kebutuhan *user*. Setiap pengembangan sebuah sistem perangkat lunak memerlukan adanya dokumentasi terhadap kebutuhan-kebutuhan *user* agar tujuan tersebut tercapai. Tahap analisis berorientasi obyek dilakukan dengan memodelkan kebutuhan *user* ke dalam *use case diagram*. *Use case diagram* bertujuan untuk menggambarkan kebutuhan-kebutuhan fungsional yang harus disediakan oleh sistem informasi agar dapat memecahkan permasalahan yang dihadapi oleh *user*.

##### 4.1.1. Analisis Sistem Eksisting

Prasyarat untuk bisa mengikuti praktikum adalah bagi yang akan mengambil praktikum :

1. Praktikum dasar, telah menempuh matakuliah yang bersangkutan dengan nilai tidak sama dengan K.
2. Praktikum kosentrasi, telah menempuh matakuliah yang bersangkutan dan lulus dengan nilai minimal D.
3. Praktikum bengkel elektro, telah menempuh minimal sama dengan 35 sks.

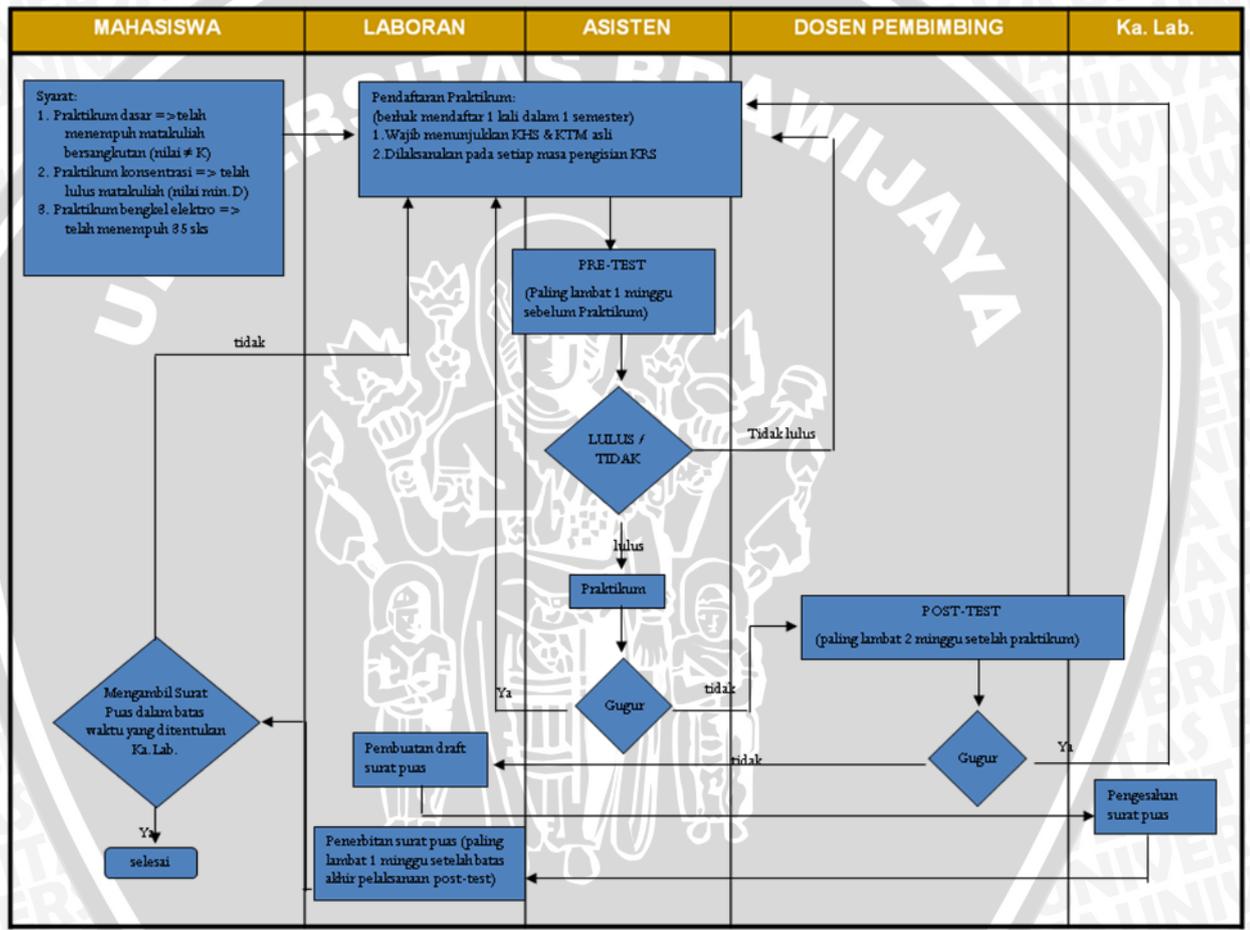
Selama ini proses pendaftaran praktikan di laboratorium Teknik Elektro Universitas Brawijaya menggunakan cara manual pada laboratorium yang bersangkutan. Secara garis besar adalah calon praktikan / mahasiswa harus datang langsung pada laboratorium yang ada, kemudian asisten mencatat sesuai kuota yang telah disediakan. Apabila calon praktikan mendaftar lebih dari 2 praktikum atau lebih, maka calon praktikan harus ke laboratorium tersebut satu – persatu. Untuk meng-*update* informasi tentang praktikum juga demikian, harus dilihat pada papan pengumuman yang tersebar di gedung Teknik Elektro.

Perancangan dan pembuatan perangkat lunak yang akan dilakukan meliputi perancangan sistem yang dimulai dengan pembentukan SOP (*Standard Operational Procedure*). SOP manual Sistem Informasi Manajemen Laboratorium Teknik Elektro Universitas Brawijaya ditunjukkan seperti pada tabel 4.1 dan

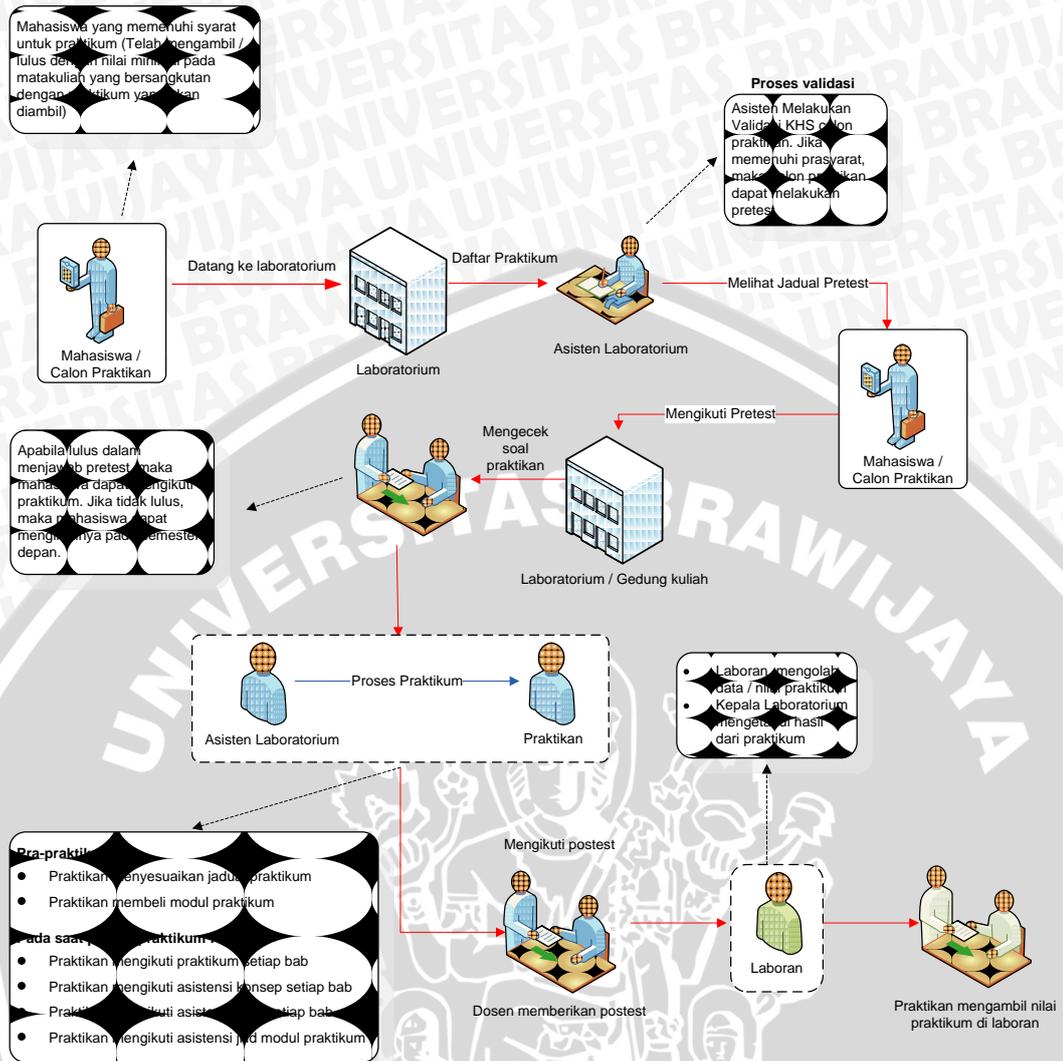
Gambar 4.2 berikut:

Prosedur Praktikum yang ada sekarang (exist) :

Tabel 4.1 SOP mahasiswa mengikuti praktikum



Sumber [Laboratorium Teknik Elektro Universitas Brawijaya]



Gambar 4.2 Blok Diagram Sistem Pengambilan Data (exist / saat ini)

Sumber : [Laboratorium Teknik Elektro Universitas Brawijaya]

Kelebihan dari sistem manual :

1. Kasus-kasus tertentu yang tidak dapat diselesaikan secara otomatis lewat sistem dapat lebih mudah diatasi.
2. Kegagalan pada satu bagian sistem tidak selalu berpengaruh pada keseluruhan sistem.

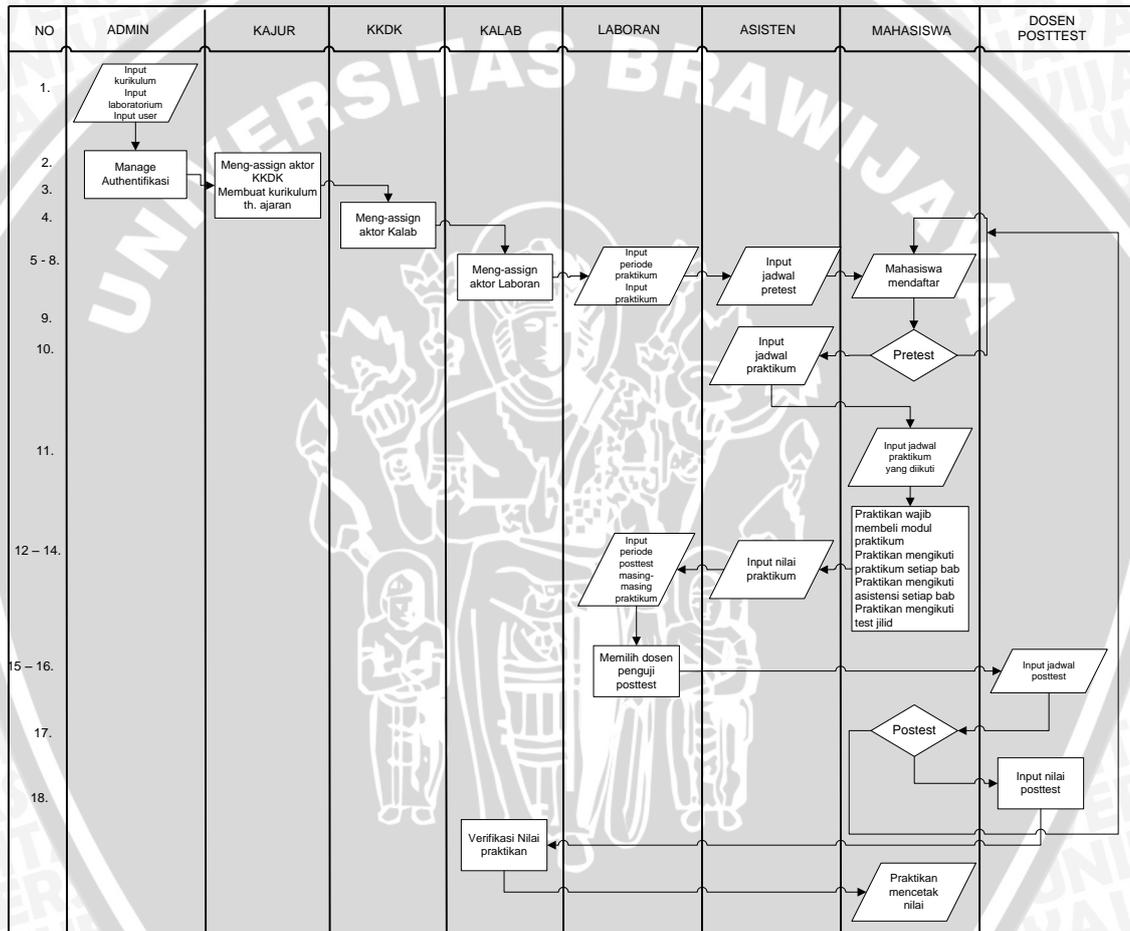
Kekurangan dari sistem penjualan manual adalah :

3. Integritas dan validitas data tidak dapat dijaga.
4. Keamanan data tidak terjamin.

5. Pengelolaan dan pelaporan data praktikum lebih rumit dan membutuhkan waktu yang lama.
6. Data tidak *up to date*.
7. Data hanya dapat diakses pada tempat dimana data disimpan.

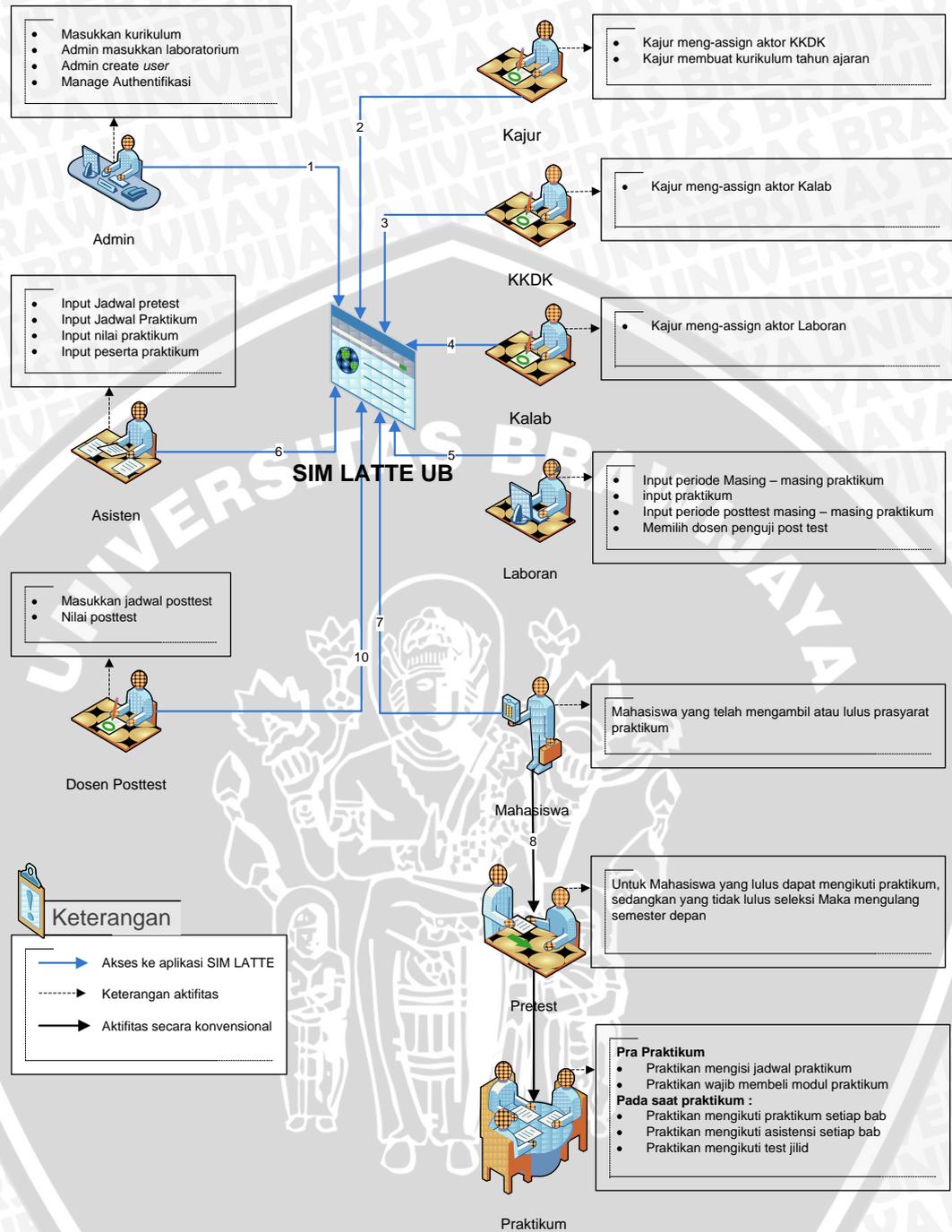
#### 4.1.2. Analisis Kebutuhan Sistem yang Baru

Tabel 4.2 SOP mahasiswa mengikuti praktikum



Sumber [perancangan]

Dalam perancangan sistem juga disertakan Blok diagram sistem dari Sistem Informasi yang akan dibuat. Arsitektur sistem Informasi Demografi dapat ditunjukkan pada gambar 4.3 berikut.



**Gambar 4.3** Blok Diagram Sistem Pengambilan Data (Sistem yang akan dibuat)

**Sumber :** [Laboratorium Teknik Elektro Universitas Brawijaya]

Pengembangan sebuah perangkat lunak bertujuan untuk menghasilkan perangkat lunak yang dapat memenuhi kebutuhan *user*. Setiap pengembangan sebuah sistem perangkat lunak memerlukan adanya dokumentasi terhadap kebutuhan-kebutuhan *user* agar tujuan tersebut tercapai. Tahap analisis



kebutuhan dilakukan dengan memodelkan kebutuhan ke dalam *use case diagram*. *Use case diagram* bertujuan untuk menggambarkan kebutuhan-kebutuhan fungsional yang harus disediakan oleh Sistem Informasi Manajemen Laboratorium agar dapat memecahkan permasalahan yang dihadapi oleh pengguna.

#### 4.1.2.1 Daftar Kebutuhan

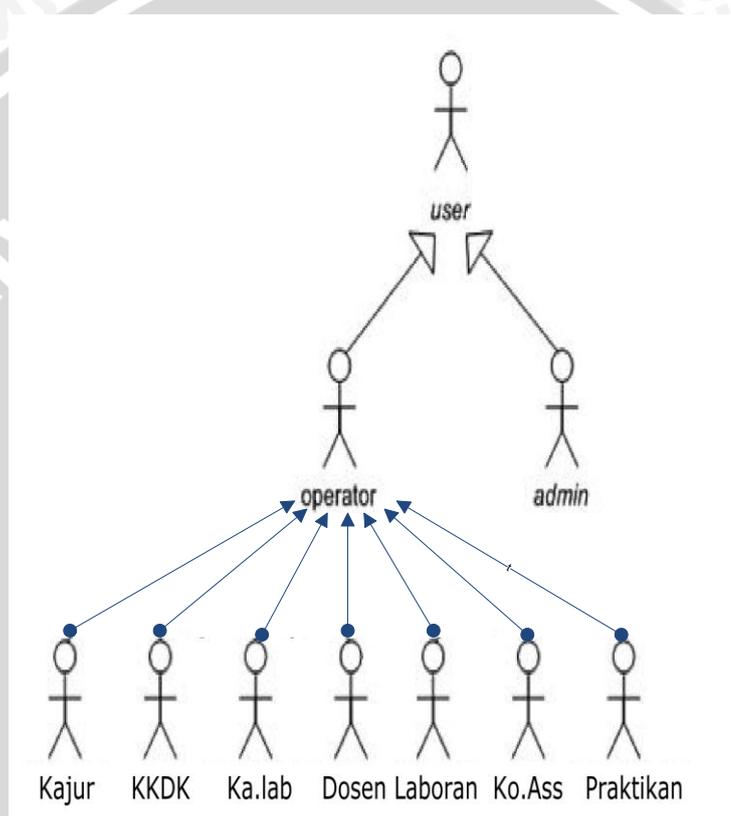
Daftar kebutuhan merupakan daftar yang menguraikan kebutuhan-kebutuhan *user* yang harus disediakan oleh perangkat lunak baik kebutuhan fungsional maupun non fungsional. Proses yang dilakukan sebelum menentukan daftar kebutuhan adalah mengidentifikasi aktor yang menggunakan Sistem Informasi Penjualan. Tabel berikut memperlihatkan seorang aktor beserta penjelasannya yang merupakan hasil dari proses identifikasi aktor.

**Tabel 4.3** Deskripsi Aktor

No.	Aktor	Keterangan
1.	<i>user</i>	Aktor yang menggunakan Sistem Informasi Manajemen Laboratorium yang belum melakukan login.
2.	Kalab	Aktor yang dapat melihat informasi data aktor mahasiswa sesuai laboratorium yang dikepalai.
3.	KKDK	Aktor yang dapat melihat informasi data aktor mahasiswa sesuai dengan paket yang bersangkutan.
4.	Kajur & Sekjur	Aktor yang dapat melihat informasi data aktor seluruh mahasiswa.
3.	Praktikan	Aktor yang memproses jadwal praktikan yang bersangkutan.
4.	Asisten	Aktor yang mengelola jadwal praktikum, <i>input</i> nilai praktikan.
5.	Laboran	Aktor yang mengelola data praktikan dan mengelola periode jadwal praktikum laboratorium dan jadwal <i>post-test</i> .
6.	Dosen	Aktor yang melakukan proses <i>input</i> jadwal <i>post-test</i> dan <i>input</i> nilai praktikan.
7.	admin	Aktor yang mengelola sistem termasuk manajemen <i>user</i> Sistem Informasi Manajemen Laboratorium Teknik Elektro.

Sumber : [Perancangan]

Aktor praktikan, asisten, dosen, laboran adalah operator dari Sistem Informasi Manajemen Laboratorium pada Teknik Elektro Universitas Brawijaya. Aktor admin adalah *administrator* Sistem Informasi Manajemen Laboratorium pada Teknik Elektro Universitas Brawijaya. Generalisasi dari para pengguna Sistem Informasi Manajemen Laboratorium ditunjukkan pada gambar 4.3.



Gambar 4.4 Generalisasi Aktor

Sumber : [Perancangan]

Setelah melakukan proses identifikasi aktor maka dilakukan proses penyusunan daftar kebutuhan fungsional sistem. Daftar kebutuhan fungsional sistem disertai dengan nama *use case* yang merepresentasikan fungsionalitas dari kebutuhan tersebut. Daftar kebutuhan fungsional tersebut ditunjukkan pada tabel berikut:

Tabel 4.4 Daftar Kebutuhan Fungsional

No.	Kebutuhan	Use Case
1.	Sistem harus mampu memberikan fasilitas untuk login, sehingga hanya pengguna tertentu yang telah terdaftar dapat masuk dan mengakses fasilitas tertentu (sebagai suatu aktor). Pengguna sistem yang tidak login hanya akan menjadi aktor <i>user</i> .	<i>Login</i>
2.	Sistem harus menyediakan fasilitas agar pengguna yang telah login dapat keluar dari sistem.	<i>Logout</i>
3.	Sistem harus menyediakan fasilitas untuk menampilkan data operator. Data yang dapat ditampilkan dari data operator adalah <i>username</i> , <i>user real name</i> dan level autentifikasi. Sistem dapat dijalankan oleh aktor administrator.	Melihat data operator
4.	Sistem harus menyediakan fasilitas untuk menambah <i>user</i> . Data yang harus dicatat dari penambahan <i>user</i> adalah <i>username</i> , <i>password</i> , <i>user real name</i> , dan level autentifikasi. Sistem dapat dijalankan oleh aktor administrator.	Menambah operator
5.	Sistem harus menyediakan fasilitas untuk mengedit data <i>user</i> . Data yang dapat diedit dari data <i>user</i> adalah <i>username</i> , <i>password</i> , <i>user real name</i> , dan level autentifikasi. Sistem dapat dijalankan oleh aktor administrator.	Mengedit data operator
6.	Sistem harus menyediakan fasilitas untuk menghapus <i>user</i> . Aktor yang dapat menghapus aktor lain adalah aktor admin. Sistem dapat dijalankan oleh aktor administrator.	Menghapus operator
7.	Sistem harus menyediakan fasilitas untuk mencari data <i>user</i> . Pencarian data dapat berdasarkan nama <i>user</i> dan <i>username</i> . Sistem dapat dijalankan oleh aktor administrator.	Mencari operator
8.	Sistem harus menyediakan fasilitas tampilan data periode masing-masing praktikum. Data yang ditampilkan adalah laboratorium yang akan melaksanakan praktikum, ubah periode, menu hapus periode. Sistem dapat dijalankan oleh aktor laboran.	Melihat periode praktikum
9.	Sistem harus menyediakan fasilitas tambah data periode masing-masing praktikum. Data periode yang dapat ditambahkan adalah periode praktikum pada laboratorium yang bersangkutan. Sistem dapat dijalankan oleh aktor laboran.	Menambah periode praktikum
10.	Sistem harus menyediakan fasilitas mengedit data periode masing-masing praktikum. Data periode yang dapat diedit adalah periode praktikum pada laboratorium yang bersangkutan. Sistem dapat dijalankan oleh aktor laboran.	Mengedit periode praktikum

11.	Sistem harus menyediakan fasilitas menghapus periode masing-masing praktikum. Data periode yang dapat dihapus adalah periode praktikum pada laboratorium yang bersangkutan. Sistem dapat dijalankan oleh aktor laboran.	Menghapus periode praktikum
12.	Sistem harus menyediakan fasilitas tampilan data periode posttest dari masing-masing praktikum yang bersangkutan. Data yang ditampilkan adalah periode posttest untuk dosen penguji praktikum, dosen penguji posttest, daftar peserta posttest (praktikan) yang akan ditest dosen yang bersangkutan. Sistem dapat dijalankan oleh aktor laboran.	Melihat periode <i>posttest</i>
13.	Sistem harus menyediakan fasilitas menambah data periode posttest masing-masing praktikum yang bersangkutan. Data yang ditambah adalah periode posttest untuk dosen penguji praktikum, dosen penguji posttest, daftar peserta posttest (praktikan) yang akan ditest dosen yang bersangkutan. Sistem dapat dijalankan oleh aktor laboran.	Menambah periode posttest
14.	Sistem harus menyediakan fasilitas mengedit data periode posttest masing-masing praktikum yang bersangkutan. Data yang diedit adalah periode posttest untuk dosen penguji praktikum, dosen penguji posttest, daftar peserta posttest (praktikan) yang akan ditest dosen yang bersangkutan. Sistem dapat dijalankan oleh aktor laboran.	Mengedit periode posttest
15.	Sistem harus menyediakan fasilitas menghapus data periode posttest dari masing-masing praktikum yang bersangkutan. Data yang dapat dihapus adalah periode posttest untuk dosen penguji praktikum, dosen penguji posttest, daftar peserta posttest (praktikan) yang akan ditest dosen yang bersangkutan. Sistem dapat dijalankan oleh aktor laboran.	Menghapus periode posttest
16.	Sistem harus menyediakan fasilitas melihat detil jadwal <i>posttest</i> dari praktikum yang bersangkutan. Data yang dapat dilihat adalah nama dosen penguji <i>posttest</i> , praktikan yang akan diuji, tanggal pengujian. Sistem dapat dijalankan oleh praktikan, asisten dan dosen.	Melihat jadwal <i>posttest</i>
17.	Sistem harus menyediakan fasilitas menambah jadwal <i>posttest</i> dari praktikum yang bersangkutan. Data yang dapat dilihat adalah nama dosen penguji <i>posttest</i> , praktikan yang akan diuji, tanggal pengujian. Sistem dapat dijalankan oleh dosen.	Menambah jadwal <i>posttest</i>
18.	Sistem harus menyediakan fasilitas untuk menampilkan data jadwal	Melihat jadwal praktikum

	praktikum. Data yang ditampilkan adalah laboratorium yang melaksanakan praktikum, periode masing-masing praktikum, kuota masing-masing periode praktikum, tanggal pelaksanaan praktikum, daftar peserta praktikum.	
19.	Sistem harus menyediakan fasilitas untuk menambah jadwal praktikum. Data yang dapat ditambahkan adalah laboratorium yang melaksanakan praktikum, periode masing-masing praktikum, kuota masing-masing periode praktikum, tanggal pelaksanaan praktikum, daftar peserta praktikum.	Menambah jadwal praktikum
20.	Sistem harus menyediakan fasilitas untuk mengubah jadwal praktikum. Data yang dapat diubah adalah laboratorium yang melaksanakan praktikum, periode masing-masing praktikum, kuota masing-masing periode praktikum, tanggal pelaksanaan praktikum, daftar peserta praktikum	Mengubah jadwal praktikum
21.	Sistem harus menyediakan fasilitas untuk menghapus jadwal praktikum. Data yang dapat dihapus adalah laboratorium yang melaksanakan praktikum, periode masing-masing praktikum, kuota masing-masing periode praktikum, tanggal pelaksanaan praktikum, daftar peserta praktikum.	Menghapus jadwal praktikum
22.	Sistem harus menyediakan fasilitas untuk menampilkan detil data nilai praktikum. Data yang dapat ditampilkan adalah nama praktikan, nim praktikan, angkatan praktikan, nama laboratorium, nama praktikum, nilai asisten, nilai dosen, nilai rata-rata (dari asisten dan dosen).	Melihat nilai praktikum
23.	Sistem harus menyediakan fasilitas untuk menambahkan detil data nilai praktikum. Data yang dapat ditampilkan adalah nama praktikan, nim praktikan, angkatan praktikan, nama laboratorium, nama praktikum, nilai asisten, nilai dosen, nilai rata-rata (dari asisten dan dosen).	Menambah nilai praktikum
24.	Sistem harus menyediakan fasilitas untuk mengedit detil data nilai praktikum. Data yang dapat ditampilkan adalah nama praktikan, nim praktikan, angkatan praktikan, nama laboratorium, nama praktikum, nilai asisten, nilai dosen, nilai rata-rata (dari asisten dan dosen).	Mengedit nilai praktikum

Sumber : [Perancangan]

Keseluruhan kebutuhan fungsionalitas di atas dibagi menjadi 2 modul untuk lebih mempermudah pemahaman dan perancangan sistem. Kedua modul tersebut adalah sebagai berikut :

1. Modul administrasi sistem
2. Modul praktikum

Pembagian modul-modul dan *use case* yang termasuk dalam modul tersebut ditunjukkan dalam tabel berikut ini :

**Tabel 4.5** Modul dan *Use Case*

No.	Modul	<i>Use Case</i>
1.	Modul administrasi sistem	<ol style="list-style-type: none"> <li>1. <i>Login</i></li> <li>2. <i>Logout</i></li> <li>3. Melihat data operator</li> <li>4. Menambah operator</li> <li>5. Mengedit data operator</li> <li>6. Menghapus operator</li> <li>7. Mencari operator</li> </ol>
2.	Modul praktikum	<ol style="list-style-type: none"> <li>1. Melihat data periode praktikum</li> <li>2. Melihat detail periode praktikum</li> <li>3. Menambah periode praktikum</li> <li>4. Menghapus periode praktikum</li> <li>5. Melihat data periode <i>posttest</i></li> <li>6. Menambah data periode <i>posttest</i></li> <li>7. Mengedit data periode <i>posttest</i></li> <li>8. Menghapus data periode <i>posttest</i></li> <li>9. Melihat detail jadwal <i>posttest</i></li> <li>10. Menambah detail jadwal <i>posttest</i></li> <li>11. Melihat jadwal praktikum</li> <li>12. Menambah jadwal praktikum</li> <li>13. Mengedit jadwal praktikum</li> <li>14. Menghapus jadwal praktikum</li> <li>15. Melihat nilai praktikum</li> <li>16. Menambah nilai praktikum</li> <li>17. Mengedit nilai praktikum</li> </ol>

Sumber : [Perancangan]



Daftar kebutuhan non fungsional Sistem Informasi Penjualan ditunjukkan pada tabel berikut :

**Tabel 4.6** Daftar Kebutuhan Non Fungsional

No.	Kebutuhan
1.	Sistem informasi dikembangkan menggunakan bahasa PHP
2.	Sistem informasi harus dapat diakses menggunakan <i>web browser</i>

Sumber : [Perancangan]

#### 4.1.2.2 Use Case Diagram

*Use case diagram* adalah salah satu diagram dalam UML untuk memodelkan kebutuhan dari sistem. Masing-masing *use case diagram* menunjukkan sekumpulan *use case*, aktor dan hubungannya.

Pemodelan dalam *use case diagram* yang menggambarkan fungsionalitas yang disediakan oleh Sistem Informasi Penjualan dibagi menjadi tiga buah diagram yang bersesuaian dengan modul dalam Sistem Informasi Penjualan.

##### 4.1.2.2.1 Use Case Diagram untuk Modul Administrasi Sistem

Gambar berikut adalah *use case diagram* untuk modul administrasi sistem. Diagram *use case* ini melibatkan 7 aktor dan 14 *use case*. Semua *use case* yang termasuk dalam modul pendukung administrasi sistem dapat dilihat pada Tabel 4.3.

Ketujuh aktor yang terlibat dalam modul administrasi sistem adalah :

1. *user*
2. admin
3. Laboran
4. Asisten
5. Praktikan
6. KKDK
7. Kalab
8. Kajur

9. Dosen

Berikut adalah spesifikasi dari *use case* untuk modul administrasi sistem :



Gambar 4.5 spesifikasi *use case* untuk login

Sumber : [Perancangan]

Spesifikasi *Use Case* untuk Login

Tabel 4.7 Spesifikasi *Use Case* Login

<b>Nama use case</b>	Login	
<b>Aktor</b>	user	
<b>Deskripsi</b>	Sistem harus mampu memberikan fasilitas untuk login, sehingga hanya pengguna tertentu yang telah terdaftar dapat masuk dan mengakses fasilitas tertentu (sebagai suatu aktor). Pengguna sistem yang tidak login hanya akan menjadi aktor user.	
<b>Pra-kondisi</b>	Sistem telah berjalan dan menampilkan form untuk login.	
<b>Pasca-kondisi</b>	User telah login sebagai aktor operator atau aktor admin.	
<b>Aliran Utama</b>		
	<b>Aksi dari Aktor</b>	<b>Tanggapan dari Sistem</b>
	1. user memasukkan <i>username</i> dan <i>password</i> dan menekan tombol submit.	2. Sistem memvalidasi <i>username</i> dan <i>password</i> , dan aktor user telah login. Maka sistem akan menampilkan halaman <i>control panel</i> sesuai dengan level user yang menjalankan <i>use case</i> login.
<b>Aliran Alternatif 1 : Pasangan <i>username</i> dan <i>password</i> tidak ada dalam sistem</b> (user pada langkah nomer 1 aliran utama tidak memasukkan <i>username</i> atau <i>password</i> yang sesuai.)		
	<b>Aksi dari Aktor</b>	<b>Tanggapan dari Sistem</b>
		1. Menampilkan pesan bahwa pasangan <i>username</i> dan <i>password</i> tidak cocok.
		2. Kembali pada langkah nomer 1 aliran utama.

Sumber : [Perancangan]

## Spesifikasi Use Case untuk Logout

Tabel 4.8 Spesifikasi Use Case Logout

<b>Nama use case</b>	Logout	
<b>Aktor</b>	Operator atau admin	
<b>Deskripsi</b>	Sistem harus menyediakan fasilitas agar pengguna yang telah login dapat keluar dari sistem.	
<b>Pra-kondisi</b>	User telah login.	
<b>Pasca-kondisi</b>	User telah logout.	
<b>Aliran Utama</b>		
<b>Aksi dari Aktor</b>	<b>Tanggapan dari Sistem</b>	
1. Aktor memilih menu <i>logout</i> .	2. Sistem melakukan <i>logout</i> untuk aktor tersebut.	

Sumber : [Perancangan]



Gambar 4.6 Spesifikasi use case untuk aktor mahasiswa

Sumber : [Perancangan]

## Use case lihat informasi praktikum

Tabel 4.9 Spesifikasi use case lihat informasi praktikum

<b>Nama use case</b>	Lihat informasi	
<b>Aktor</b>	Semua aktor	
<b>Tujuan</b>	Untuk melihat informasi yang dikeluarkan laboratorium	
<b>Deskripsi</b>	Lihat informasi menampilkan informasi yang dikeluarkan oleh aktor admin, laboran, asisten, dosen. Dapat dilihat dengan/tanpa proses <i>login</i> .	
<b>Pra-kondisi</b>	Sistem telah dijalankan <i>use case login</i> .	
<b>Pasca-kondisi</b>	Sistem menampilkan informasi laboratorium.	
<b>Aliran utama</b>		
<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>	

1. Mahasiswa menjalankan sistem informasi laboratorium.	2. Sistem menampilkan informasi yang diterbitkan oleh aktor admin, laboran, asisten.
---	--

Sumber : [Perancangan]

### Use case daftar praktikum

Tabel 4.10 Spesifikasi use case daftar praktikum

<b>Nama use case</b>	Daftar praktikum	
<b>Aktor</b>	Mahasiswa	
<b>Tujuan</b>	Untuk melakukan pendaftaran praktikum.	
<b>Deskripsi</b>	Pendaftaran dilakukan dengan memberikan tanda centang pada periode praktikum yang ingin diikuti kemudian menekan tombol daftar.	
<b>Pra-kondisi</b>	Sistem telah dijalankan pada use case login dan menekan tombol navigasi daftar praktikum. Aktor Laboran telah memberikan input periode untuk masing-masing praktikum.	
<b>Pasca-kondisi</b>	Mahasiswa telah terdaftar sebagai calon praktikan.	
<b>Aliran utama</b>		
<b>Aksi dari aktor</b>		<b>Tanggapan dari sistem</b>
1. Mahasiswa memberikan tanda centang pada periode praktikum yang ingin diikuti.	2.	Sistem memberi tanda pada pada semua praktikum dengan periode yang sama.
3. Mahasiswa menekan tombol “daftar”.	4.	Sistem menyimpan data pendaftaran praktikum kedalam basis data dan menampilkan pesan bahwa pendaftaran telah dilakukan.
5. Mahasiswa menekan tombol “Batal”	6.	Sistem akan kembali pada use case lihat informasi praktikum.
<b>Aliran Alternatif 1 : Periode praktikum sama</b>		
Aktor mahasiswa memilih / mencentang waktu periode yang sama.)		
	1.	Sistem menampilkan pesan bahwa waktu dari periode praktikum tidak boleh sama.

Sumber : [Perancangan]

### Use case pembatalan praktikum

Tabel 4.11 Spesifikasi use case pembatalan praktikum

<b>Nama use case</b>	Pembatalan praktikum
<b>Aktor</b>	Mahasiswa



<b>Tujuan</b>	Untuk melakukan pembatalan praktikum yang telah didaftar.	
<b>Deskripsi</b>	Proses pembatalan dilakukan dengan menekan tombol hapus pada list praktikum yang telah didaftar.	
<b>Pra-kondisi</b>	Sistem dapat dijalankan apabila aktor telah <i>login</i> , aktor telah memasukkan data pemilihan periode sebelumnya.	
<b>Pasca-kondisi</b>	Praktikum telah dihapus.	
Aliran utama		
Aksi dari aktor	Tanggapan dari sistem	
1. Aktor mahasiswa menekan tombol hapus pada <i>list</i> praktikum yang telah didaftar.	2. Sistem menampilkan pesan untuk konfirmasi “apakah anda yakin menghapus periode tersebut?”.	
3. Aktor mahasiswa memilih tombol “ya” pada pesan konfirmasi penghapusan periode praktikum.	4. sistem melakukan proses penghapusan data praktikum pada basis data dan kembali ke halaman <i>control panel</i> .	

Sumber : [Perancangan]

### Use case edit biodata

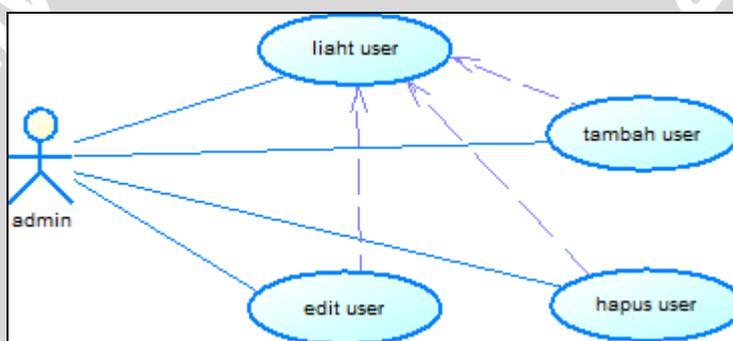
**Tabel 4.12** Spesifikasi *use case* edit biodata

<b>Nama use case</b>	Edit biodata	
<b>Aktor</b>	Semua aktor	
<b>Tujuan</b>	Untuk melakukan perubahan data mahasiswa.	
<b>Deskripsi</b>	Perubahan data aktor dilakukan dengan mengganti data user, data user meliputi: user id, password, nama, NIM/NIP, paket, alamat, telepon dan email.	
<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case</i> login dan aktor memilih tombol navigasi <i>edit</i> biodata.	
<b>Pasca-kondisi</b>	Data mahasiswa telah dirubah.	
Aliran utama		
Aksi dari aktor	Tanggapan dari sistem	
1. Aktor memilih menu edit user.	2. Sistem menampilkan halaman perubahan biodata mahasiswa. Halaman ini menampilkan 8 textfield yaitu: user id, password, nama, NIM/NIP, paket, alamat, telepon dan email serta tombol simpan dan hapus.	
3. Mahasiswa memperbaharui data user baru dan menekan tombol “simpan”.	4. Sistem memeriksa input yang dimasukan admin, apakah karakter yang dimasukan	



	sah, apakah data yang dimasukkan lengkap? Jika telah benar, maka selanjutnya sistem menyimpan data mahasiswa tersebut kedalam basis data.
<b>aliran alternative 1: data tidak lengkap</b>	
	5. Sistem menampilkan pesan kesalahan dan kembali ke langkah 2.
<b>aliran alternative 2: karakter yang dimasukkan tidak sah</b>	
	6. Sistem menampilkan pesan kesalahan dan kembali ke langkah 2.

Sumber : [Perancangan]



Gambar 4.7 Spesifikasi use case aktor admin.

Sumber : [Perancangan]

### Use case daftar user

Tabel 4.13 Spesifikasi use case lihat user

<b>Nama use case</b>	Lihat user
<b>Aktor</b>	Admin
<b>Tujuan</b>	Untuk melihat daftar user yang tercatat dalam basis data SIM laboratorium.
<b>Deskripsi</b>	Lihat user menampilkan daftar user yang tercatat dalam basis data dan dikelompokkan berdasarkan jenis aktor.
<b>Pra-kondisi</b>	Sistem telah dijalankan pada use case login.
<b>Pasca-kondisi</b>	Sistem menampilkan daftar user.
<b>Aliran utama</b>	
<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>
1. Aktor admin menjalankan sistem informasi	2. Sistem menampilkan daftar user yang



laboratorium.	tercatat didalam basis data SIM laboratorium dan dikelompokkan berdasarkan jenis aktor seperti admin, laboran, dosen, asisten, praktikan, Kalab, KKDK, Pimpinan.
---------------	--

Sumber : [Perancangan]

### Use case lihat profil user

**Tabel 4.14** Spesifikasi *use case* lihat user

<b>Nama use case</b>	Lihat aktor
<b>Aktor</b>	Admin
<b>Tujuan</b>	Untuk melihat profil <i>aktor</i> yang tercatat dalam basis data SIM laboratorium.
<b>Deskripsi</b>	Lihat <i>user</i> menampilkan profil <i>aktor</i> dan melihat privileginya yang tercatat dalam basis data.
<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case login</i> , atau memilih menu profil aktor
<b>Pasca-kondisi</b>	Sistem menampilkan profil <i>aktor</i>
<b>Aliran utama</b>	
<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>
3. Aktor admin menjalankan sistem informasi laboratorium.	4. sistem menampilkan daftar <i>user</i> yang tercatat didalam basis data SIM laboratorium.
5. Aktor admin memilih aktor	6. Sistem menampilkan halaman data aktor beserta tombol "edit user"

Sumber : [Perancangan]

### Use case tambah user

**Tabel 4.15** Spesifikasi *use case* tambah *user*

<b>Nama use case</b>	Tambah <i>user</i>
<b>Aktor</b>	Admin
<b>Tujuan</b>	Untuk melakukan penambahan <i>user</i> pada SIM Laboratorium.
<b>Deskripsi</b>	Pemanbahan user dilakukan dengan mengisikan data user baru, data user meliputi: user id, password, nama, NIM, paket, alamat, telepon dan email



<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case login</i> .
<b>Pasca-kondisi</b>	<i>User</i> baru telah terdaftar dalam sistem.
<b>Aliran utama</b>	
Aksi dari aktor	Tanggapan dari sistem
1. admin memilih menu tambah <i>user</i> .	2. sistem menampilkan halaman penambahan <i>user</i> . Halaman ini menampilkan 8 textfield yaitu: <i>user id</i> , password, nama, NIM, paket, alamat, telepon dan email serta tombol simpan dan hapus.
3. admin mengisikan data <i>user</i> baru dan menekan tombol “simpan”.	4. sistem memeriksa <i>input</i> yang dimasukan admin, apakah karakter yang dimasukan sah, apakah data yang dimasukan lengkap? Jika telah benar, maka selanjutnya sistem menyimpan data <i>user</i> tersebut kedalam basis data sistem.
<b>aliran alternative 1: data tidak lengkap</b>	
	5. sistem menampilkan pesan kesalahan dan kembali ke langkah 2.
<b>aliran alternative 2: karakter yang dimasukan tidak sah</b>	
	6. sistem menampilkan pesan kesalahan dan kembali ke langkah 2.

Sumber : [Perancangan]

### Use case edit user

**Tabel 4.16** Spesifikasi *use case* edit *user*

<b>Nama use case</b>	Edit <i>user</i>
<b>Aktor</b>	Admin
<b>Tujuan</b>	Untuk melakukan perubahan data <i>user</i> atau previlage-nya.
<b>Deskripsi</b>	Perubahan data <i>user</i> dilakukan dengan mengganti data <i>user</i> , data <i>user</i> yang dapat dirubah meliputi: password, paket, alamat, telepon, email dan previlage-nya.
<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case</i> lihat <i>user</i> .
<b>Pasca-kondisi</b>	Data <i>user</i> baru telah dirubah.
<b>Aliran utama</b>	
Aksi dari aktor	Tanggapan dari sistem
1. Aktor admin memilih menu “ <i>edit user</i> ”.	2. sistem menampilkan halaman perubahan data <i>user</i> . Halaman ini menampilkan 8



	textfield yaitu: user id, password, nama, NIM, paket, alamat, telepon, email, serta tombol simpan dan Batal.
3. Aktor admin memperbaharui data user baru dan menekan tombol simpan.	4. sistem memeriksa input yang dimasukan admin, apakah karakter yang dimasukan sah, apakah data yang dimasukan lengkap? Jika telah benar, maka selanjutnya sistem menyimpan data user tersebut kedalam basis data.
5. Aktor admin menekan tombol batal.	6. Sistem akan kembali ke awal dari halaman edit <i>user</i> .
<b>aliran alternative 1: data tidak lengkap</b>	
	7. sistem menampilkan pesan kesalahan dan kembali ke langkah 2.
<b>aliran alternative 2: karakter yang dimasukan tidak sah</b>	
	8. sistem menampilkan pesan kesalahan dan kembali ke langkah 2.

Sumber : [Perancangan]

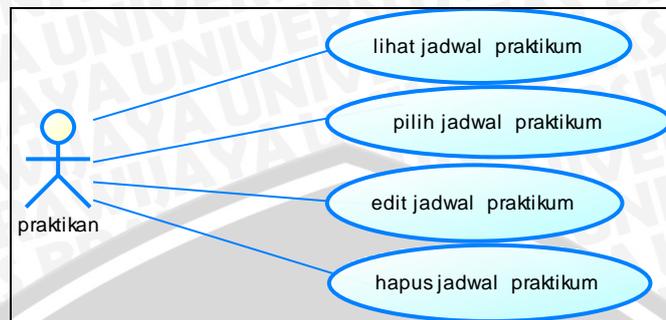
### Use case hapus user

Tabel 4.17 Spesifikasi *use case* hapus *user*

<b>Nama use case</b>	Hapus <i>user</i>
<b>Aktor</b>	Admin
<b>Tujuan</b>	Untuk melakukan penghapusan <i>user</i> pada SIM Laboratorium.
<b>Deskripsi</b>	Penghapusan dilakukan dengan menekan tombol hapus pada list <i>user</i> .
<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case</i> lihat <i>user</i> .
<b>Pasca-kondisi</b>	<i>User</i> telah dihapus dari sistem.
<b>Aliran utama</b>	
<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>
1. Aktor admin menekan tombol hapus pada list <i>user</i> .	2. sistem menampilkan pesan konfirmasi penghapusan “OK” atau “Batal”.
3. Aktor admin menekan tombol “OK” pada pesan yang ditampilkan sistem.	4. sistem menghapus <i>user</i> dari basis data sistem.
5. Aktor admin menekan tombol “Batal” pada pesan konfirmasi yang ditampilkan sistem.	6. Sistem membatalkan penghapusan <i>user</i> dari basis data sistem.

Sumber : [Perancangan]





Gambar 4.8 Spesifikasi *use case* untuk aktor praktikan.

Sumber : [Perancangan]

### Use case lihat jadwal praktikum

Tabel 4.18 Spesifikasi *use case* jadwal praktikum

<b>Nama use case</b>	Lihat jadwal praktikum
<b>Aktor</b>	praktikan
<b>Tujuan</b>	Untuk melihat jadwal praktikum yang telah diterbitkan oleh laboratorium.
<b>Deskripsi</b>	Lihat jadwal praktikum menampilkan jadwal praktikum yang dikeluarkan oleh pihak laboratorium yang bersangkutan..
<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case</i> login.
<b>Pasca-kondisi</b>	Sistem menampilkan jadwal praktikum.
<b>Aliran utama</b>	
<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>
1. praktikan memilih menu lihat jadwal praktikum di laboratorium yang bersangkutan.	2. sistem menampilkan jadwal praktikum yang telah dikeluarkan pihak laboratorium, jika jadwal belum diterbitkan maka sistem hanya menampilkan halaman kosong yang berisi pesan “Data belum tersedia”.

Sumber : [Perancangan]

### Use case pilih jadwal praktikum

Tabel 4.19 Spesifikasi *use case* pilih jadwal praktikum

<b>Nama use case</b>	pilih jadwal praktikum
----------------------	------------------------



<b>Aktor</b>	praktikan
<b>Tujuan</b>	Untuk memilih jadwal praktikum yang diambil saat itu.
<b>Deskripsi</b>	pilih jadwal praktikum menampilkan jadwal praktikum yang dikeluarkan oleh pihak laboratorium, halaman ini berupa table jadwal dengan checkbox dan tombol simpan.
<b>Pra-kondisi</b>	Praktikan lolos <i>pretest</i> kemudian Sistem telah dijalankan pada <i>use case</i> lihat jadwal praktikum.
<b>Pasca-kondisi</b>	jadwal praktikum telah disimpan dalam sistem.
<b>Aliran utama</b>	
<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>
1. praktikan memilih menu pilih jadwal.	2. sistem menampilkan tabel jadwal praktikum, halaman ini berupa tabel jadwal dengan check box dan menampilkan jumlah peserta ditiap waktu praktikum serta tombol “Simpan” dan “Batal”.
3. praktikan memilih jadwal dan menekan tombol “Simpan”.	4. sistem memeriksa apakah waktu yang dipilih sudah memenuhi kuota atau belum, jika belum maka sistem menyimpan jadwal praktikum.
5. Praktikan menekan tombol “Batal”.	6. Sistem akan kembali ke langkah 2.
<b>aliran alternative 1: kuota kelompok telah penuh</b>	
7. Praktikan mengisi jadwal pada kuota kelompok yang telah penuh.	8. sistem menampilkan pesan kesalahan, dan kembali ke langkah 2.
<b>aliran alternative 2: Jadwal tidak diisikan 1 atau lebih</b>	
9. Praktikan tidak mengisi jadwal masing-masing bab. Baik itu salah satu atau beberapa.	10. sistem menampilkan pesan “Jadwal setiap Bab harus diisi.”
<b>aliran alternative 1: Mengisi jadwal dalam waktu yang sama antar bab</b>	
11. Praktikan mengisi jadwal yang sama antar bab.	12. Sistem menampilkan pesan “Jadwal antar bab tidak boleh sama dalam satu waktu.”

Sumber : [Perancangan]

### Use case edit jadwal praktikum

Tabel 4.20 Spesifikasi *use case* edit jadwal praktikum

<b>Nama use case</b>	edit jadawal praktikum
<b>Aktor</b>	praktikan



<b>Tujuan</b>	Untuk merubah jadwal praktikum yang telah dipilih.
<b>Deskripsi</b>	edit jadwal praktikum merupakan fasilitas untuk melakukan perubahan jadwal praktikum yang telah dipilih sebelumnya.
<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case</i> lihat jadwal praktikum dan jadwal telah diisi oleh praktikan yang bersangkutan.
<b>Pasca-kondisi</b>	jadwal praktikum telah dirubah.
<b>Aliran utama</b>	
<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>
1. praktikan memilih menu edit jadwal.	2. sistem menampilkan tabel jadwal praktikum, halaman ini berupa tabel jadwal dengan check box (bertanda centang jika waktunya telah dipilih) dan menampilkan jumlah peserta di tiap waktu praktikum serta tombol “Simpan” dan “Batal”.
3. praktikan memilih jadwal yang baru dan menekan tombol “Simpan”.	4. sistem memeriksa apakah waktu yang dipilih sudah memenuhi kuota atau belum, jika belum maka sistem menyimpan jadwal praktikum.
5. Praktikan menekan tombol “Batal”.	6. Sistem kembali ke langkah 2.
<b>aliran alternative 1: kuota kelompok telah penuh</b>	
7. Praktikan memilih kelompok bab yang kuotanya telah penuh.	8. sistem menampilkan pesan kesalahan,dan kembali ke langkah 2.

Sumber : [Perancangan]

### Use case hapus jadwal praktikum

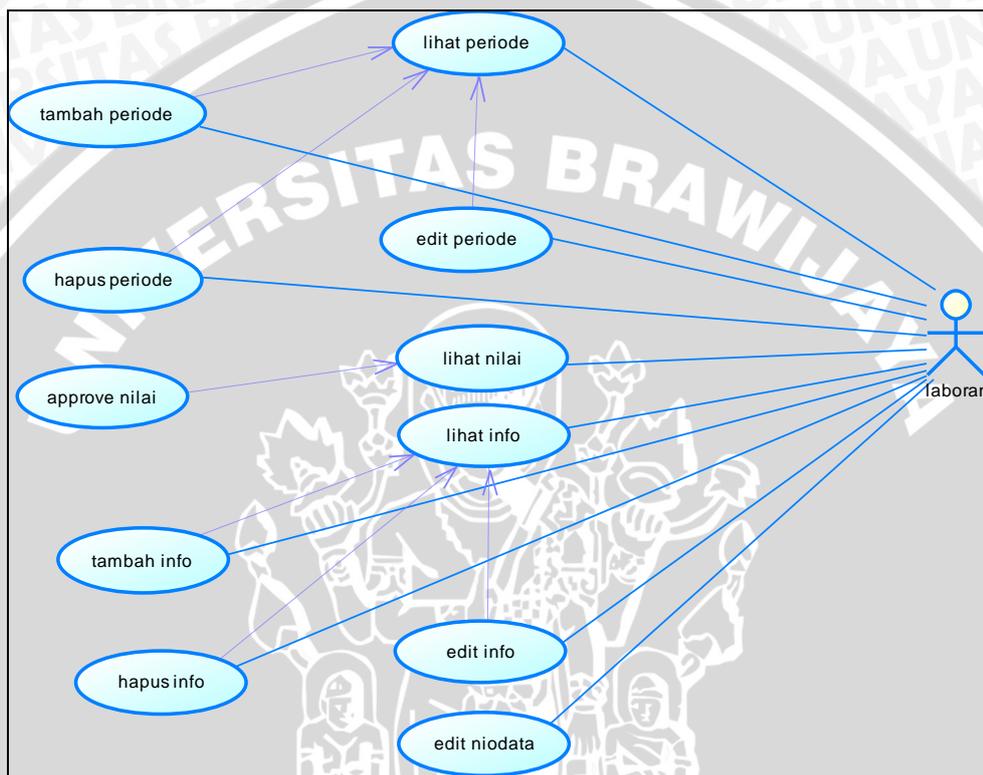
**Tabel 4.21** Spesifikasi *use case* hapus jadwal praktikum

<b>Nama use case</b>	Hapus jadawal praktikum
<b>Aktor</b>	praktikan
<b>Tujuan</b>	Untuk menghapus jadwal praktikum yang telah dipilih.
<b>Deskripsi</b>	hapus jadwal praktikum merupakan fasilitas untuk melakukan penghapusan jadwal praktikum yang telah dipilih sebelumnya.
<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case</i> lihat jadwal praktikum.
<b>Pasca-kondisi</b>	jadwal praktikum telah dihapus.
<b>Aliran utama</b>	
<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>



1. Praktikkan menekan tombol hapus pada list jadwal praktikum yang diikuti.	2. Sistem menampilkan pesan pengapusan jadwal, “OK” dan “Batal”.
3. Praktikkan menekan tombol “ok”.	4. Sistem menghapus jadwal praktikum.
5. Jika praktikkan memilih tombol “batal”.	6. Sistem akan kembali ke pra-kondisi.

Sumber : [Perancangan]



Gambar 4.9 Spesifikasi use case untuk aktor laboran

Sumber : [Perancangan]

### Use case edit biodata

Tabel 4.22 Spesifikasi use case edit biodata

<b>Nama use case</b>	Edit biodata
<b>Aktor</b>	Semua aktor
<b>Tujuan</b>	Untuk melakukan perubahan data aktor.
<b>Deskripsi</b>	Perubahan data laboran dilakukan dengan mengganti data user, data user meliputi: user id, password, nama, NIP, alamat, telepon dan email.
<b>Pra-kondisi</b>	Sistem telah dijalankan pada use case login.
<b>Pasca-kondisi</b>	Data aktor telah dirubah.
<b>Aliran utama</b>	



Aksi dari aktor	Tanggapan dari sistem
1. Aktor memilih menu edit user.	2. sistem menampilkan halaman perubahan biodata laboran. Halaman ini menampilkan 7 textfield yaitu: user id, password, nama, NIP, alamat, telepon dan email serta tombol simpan dan hapus.
3. Aktor memperbaharui data user baru dan menekan tombol “Simpan”.	4. sistem memeriksa input yang dimasukan admin, apakah karakter yang dimasukan sah, apakah data yang dimasukan lengkap? Jika telah benar, maka selanjutnya sistem mnyimpan data laboran tersebut kedalam basis data.
5. Aktor memilih tombol “Batal”	6. Sistem akan kembali ke langkah 2.
<b>aliran alternative 1: data tidak lengkap</b>	
	7. sistem menampilkan pesan kesalahan dan kembali ke langkah 2.
<b>aliran alternative 2: karakter yang dimasukan tidak sah</b>	
	8. sistem menampilkan pesan kesalahan dan kembali ke langkah 2.

Sumber : [Perancangan]

### Use case lihat periode

Tabel 4.23 Spesifikasi use case lihat periode

<b>Nama use case</b>	Lihat priode
<b>Aktor</b>	laboran
<b>Tujuan</b>	Untuk melihat periode praktikum yang telah diterbitkan oleh laboratorium.
<b>Deskripsi</b>	Lihat periode praktikum menampilkan periode praktikum yang dikeluarkan oleh pihak laboratorium yang bersangkutan dan kuota peserta.
<b>Pra-kondisi</b>	Sistem telah dijalankan pada use case login.
<b>Pasca-kondisi</b>	Sistem menampilkan periode praktikum.
<b>Aliran utama</b>	
Aksi dari aktor	Tanggapan dari sistem
1. laboran memilih menu lihat periode praktikum.	2. sistem menampilkan daftar periode praktikum yang telah dikeluarkan pihak laboratorium beserta kuota peserta, menu



	isi kuota dan ubah kuota, jika jadwal belum diterbitkan makan sistem hanya menampilkan halaman kosong.
--	--

Sumber : [Perancangan]

### Use case tambah periode

Tabel 4.24 Spesifikasi use case tambah periode

<b>Nama use case</b>	tambah priode	
<b>Aktor</b>	laboran	
<b>Tujuan</b>	Untuk menambah periode praktikum.	
<b>Deskripsi</b>	Penambahan periode dilakukan dengan mengisikan nomer periode dan waktu.	
<b>Pra-kondisi</b>	Sistem telah dijalankan pada use case login.	
<b>Pasca-kondisi</b>	Periode baru telah ditambahkan.	
<b>Aliran utama</b>		
	<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>
	1. Laboran memilih menu tambah periode praktikum.	2. sistem menampilkan halaman penambahan periode, halaman ini menampilkan form dengan 3 buah textfield yaitu: nama praktikum, nomer periode dan waktu serta tombol simpan dan hapus.
	3. Laboran mengisikan data periode dan menekan tombol simpan.	4. sistem memeriksa apakah data yang diisikan telah lengkap dan menggunakan karakter yang diijinkan? Jika benar maka sistem menyimpan periode baru tersebut.
	5. Laboran memilih tombol “SIMPAN.”	6. Sistem akan menampilkan seperti pada langkah use case hapus periode.
<b>aliran alternative 1: data periode tidak lengkap</b>		
		7. sistem menampilkan pesan kesalahan dan kembali ke langkah 2.
<b>aliran alternative 2: menggunakan karakter ilegal</b>		
		8. sistem menampilkan pesan kesalahan dan kembali ke langkah 2.

Sumber : [Perancangan]



## Use case edit periode

**Tabel 4.25** Spesifikasi *use case* edit periode

<b>Nama use case</b>	Edit priode	
<b>Aktor</b>	Laboran	
<b>Tujuan</b>	Untuk merubah periode praktikum.	
<b>Deskripsi</b>	Perubahan periode dilakukan dengan merubah nomer periode dan waktu.	
<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case</i> lihat periode.	
<b>Pasca-kondisi</b>	Periode telah dirubah	
<b>Aliran utama</b>		
	<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>
	1. Laboran memilih menu ubah periode praktikum.	2. sistem menampilkan halaman perubahan periode, halaman ini menampilkan form dengan 3 buah textfield dan nilainya yaitu: nama praktikum, nomer periode dan waktu serta tombol simpan dan hapus.
	3. Laboran memperbaharui data periode dan menekan tombol simpan.	4. sistem memeriksa apakah data yang diisikan telah lengkap dan menggunakan karakter yang diijinkan? Jika benar maka sistem menyimpan perubahan periode tersebut.
<b>aliran alternative 1: data periode tidak lengkap</b>		
		5. sistem menampilkan pesan kesalahan dan kembali ke langkah 2.
<b>aliran alternative 2: menggunakan karakter ilegal</b>		
		6. sistem menampilkan pesan kesalahan dan kembali ke langkah 2.

Sumber : [Perancangan]

## Use case hapus periode

**Tabel 4.26** Spesifikasi *use case* hapus periode

<b>Nama use case</b>	Hapus priode
<b>Aktor</b>	Laboran
<b>Tujuan</b>	Untuk menghapus periode praktikum.
<b>Deskripsi</b>	Penghapusan periode dilakukan dengan menekan tombol hapus pada list periode.

<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case</i> lihat periode.	
<b>Pasca-kondisi</b>	Periode telah dihapus.	
<b>Aliran utama</b>		
<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>	
1. Laboran memilih menekan tombol hapus pada daftar periode.	2. Sistem menampilkan pesan penghapusan periode(pesan).	
3. Laboran menekan tombol “HAPUS” pada pesan.	4. Sistem menghapus periode tersebut dari basis data sistem.	
5. Laboran menekan tombol “Batal.”	6. Sistem menampilkan halaman saat pra-kondisi.	

Sumber : [Perancangan]

### Use case ubah kuota praktikum

**Tabel 4.27** Spesifikasi *use case* hapus periode

<b>Nama use case</b>	Ubah kuota praktikum	
<b>Aktor</b>	Laboran	
<b>Tujuan</b>	Untuk mengubah kuota peserta praktikum.	
<b>Deskripsi</b>	Pengubahan kuota peserta praktikum dengan mengisikan jumlah peserta yang akan diterima pada periode praktikum tersebut	
<b>Pra-kondisi</b>	Sistem dapat dijalankan apabila aktor telah <i>login</i> , atau memilih menu ubah kuota pada <i>use case</i> lihat periode.	
<b>Pasca-kondisi</b>	Sistem menyimpan kuota praktikum pada periode tersebut.	
<b>Aliran utama</b>		
<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>	
1. Laboran menekan tombol ubah kuota pada daftar periode.	2. Sistem menampilkan field untuk pengubahan kuota peserta praktikum pada periode tersebut.	
3. Laboran mengisi jumlah kuota peserta praktikum lalu menekan tombol “simpan”.	4. Sistem menyimpan jumlah kuota peserta praktikum pada periode tersebut. Dan kembali ke <i>use case</i> lihat periode.	
<b>aliran alternative 1: menggunakan karakter ilegal</b>		
	5. Sistem menampilkan pesan kesalahan dan kembali ke langkah 2.	

Sumber : [Perancangan]



## Use case lihat informasi

**Tabel 4.28** Spesifikasi *use case* lihat informasi

<b>Nama use case</b>	Lihat informasi
<b>Aktor</b>	laboran
<b>Tujuan</b>	Untuk melihat informasi yang telah diterbitkan oleh laboratorium.
<b>Deskripsi</b>	Lihat informasi menampilkan informasi yang dikeluarkan oleh pihak laboratorium yang bersangkutan.
<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case login</i> .
<b>Pasca-kondisi</b>	Sistem menampilkan informasi yang diterbitkan laboratorium.
Aliran utama	
Aksi dari aktor	Tanggapan dari sistem
1. Laboran memilih menu lihat informasi.	2. Sistem menampilkan informasi yang telah dikeluarkan pihak laboratorium, jika jadwal belum diterbitkan maka sistem hanya menampilkan halaman kosong.

Sumber : [Perancangan]

## Use case tambah informasi

**Tabel 4.29** Spesifikasi *use case* tambah informasi

<b>Nama use case</b>	Tambah informasi
<b>Aktor</b>	Laboran
<b>Tujuan</b>	Untuk menambah informasi.
<b>Deskripsi</b>	Penambahan informasi dilakukan dengan judul dan informasi.
<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case login</i> .
<b>Pasca-kondisi</b>	Informasi baru telah ditambahkan.
Aliran utama	
Aksi dari aktor	Tanggapan dari sistem
1. Laboran memilih menu tambah informasi.	2. Sistem menampilkan halaman penambahan informasi, halaman ini menampilkan form dengan 2 buah textfield yaitu: judul dan informasi serta tombol simpan dan hapus.
3. Laboran mengisikan informasi dan menekan tombol simpan.	4. Sistem memeriksa apakah data yang diisikan telah lengkap dan menggunakan karakter yang diijinkan? Jika benar maka sistem menyimpan informasi baru tersebut.
5. Laboran menekan tombol "Batal"	6. Sistem akan menampilkan langkah pra-

	kondisi.
<b>aliran alternative 1: data periode tidak lengkap</b>	
	7. Sistem menampilkan pesan kesalahan dan kembali ke langkah 2.
<b>aliran alternative 2: menggunakan karakter ilegal</b>	
	8. Sistem menampilkan pesan kesalahan dan kembali ke langkah 2.

Sumber : [Perancangan]

### Use case edit informasi

**Tabel 4.30** Spesifikasi *use case* edit informasi

<b>Nama use case</b>	Edit informasi	
<b>Aktor</b>	Laboran	
<b>Tujuan</b>	Untuk merubah informasi yang telah diterbitkan.	
<b>Deskripsi</b>	Perubahan informasi dilakukan dengan merubah judul dan informasi.	
<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case</i> lihat informasi.	
<b>Pasca-kondisi</b>	Informasi telah dirubah	
<b>Aliran utama</b>		
<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>	
1. Laboran memilih menu ubah informasi.	2. Sistem menampilkan halaman perubahan informasi, halaman ini menampilkan form dengan 2 buah textfield dan nilainya yaitu: judul dan informasi dan waktu serta tombol simpan dan hapus.	
3. Laboran memperbaharui informasi dan menekan tombol simpan.	4. Sistem memeriksa apakah data yang diisikan telah lengkap dan menggunakan karakter yang diijinkan? Jika benar maka sistem menyimpan perubahan periode tersebut.	
5. Laboran menekan tombol "Batal"	6. Sistem akan menampilkan use case pra-kondisi.	
<b>aliran alternative 1: data periode tidak lengkap</b>		
	7. Sistem menampilkan pesan kesalahan dan kembali ke langkah 2.	
<b>aliran alternative 2: menggunakan karakter ilegal</b>		
	8. Sistem menampilkan pesan kesalahan dan kembali ke langkah 2.	

Sumber : [Perancangan]



## Use case hapus informasi

Tabel 4.31 Spesifikasi *use case* hapus informasi

<b>Nama use case</b>	Hapus informasi
<b>Aktor</b>	Laboran
<b>Tujuan</b>	Untuk menghapus informasi.
<b>Deskripsi</b>	Penghapusan periode dilakukan dengan menekan tombol hapus pada list informasi.
<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case</i> lihat informasi.
<b>Pasca-kondisi</b>	informasi telah dihapus.
<b>Aliran utama</b>	
<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>
1. Laboran memilih menekan tombol hapus pada daftar informasi.	2. Sistem menampilkan pesan penghapusan informasi (pesan).
3. Laboran menekan tombol “ok” pada pesan.	4. Sistem menghapus informasi tersebut dari basis data sistem.
5. Laboran menekan tombol “Batal.”	6. Sistem akan menampilkan langkah pra-kondisi.

Sumber : [Perancangan]

## Use case lihat nilai

Tabel 4.32 Spesifikasi *use case* lihat nilai

<b>Nama use case</b>	Lihat nilai
<b>Aktor</b>	laboran
<b>Tujuan</b>	Untuk melihat nilai akhir praktikum yang dihasilkan sistem.
<b>Deskripsi</b>	Lihat nilai menampilkan nilai hasil perhitungan sistem, perhitungan dilakukan sistem dengan mengalikan nilai dari asisten dan dosen dengan prosentase 40% -60%.
<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case login</i> .
<b>Pasca-kondisi</b>	Sistem menampilkan nilai akhir praktikum.
<b>Aliran utama</b>	
<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>
1. Laboran memilih menu lihat nilai.	2. Sistem menampilkan nilai hasil perhitungan sistem, perhitungan dilakukan sistem dengan mengalikan nilai dari asisten dan dosen dengan prosentase 40%-60%.

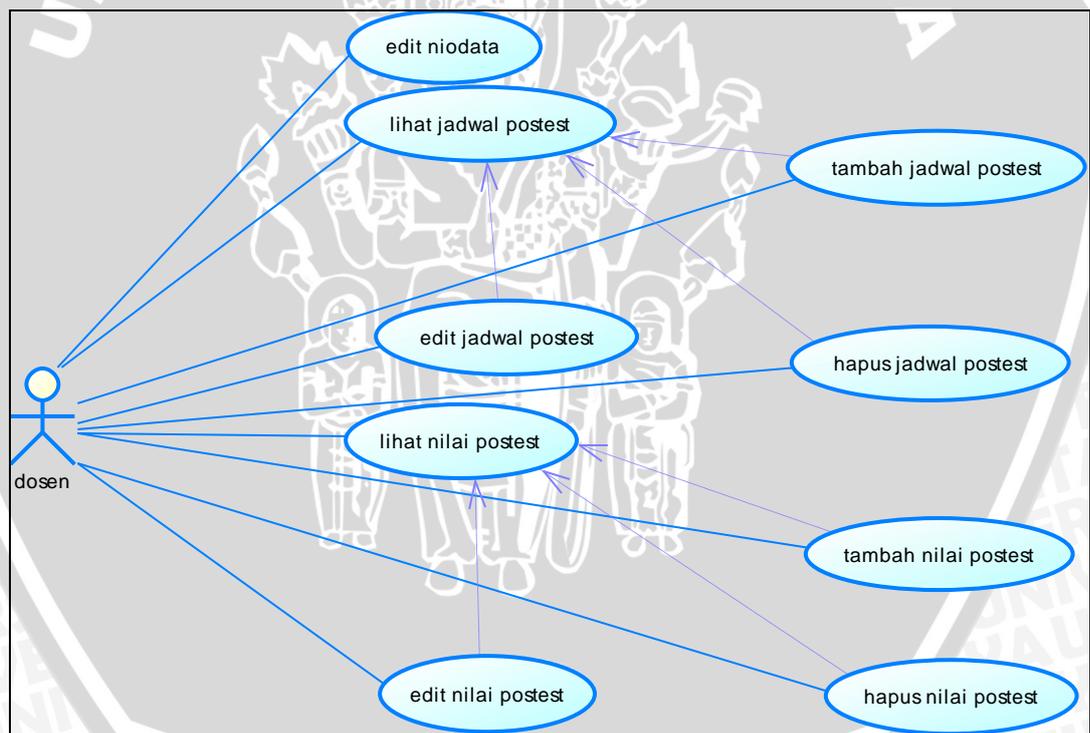
Sumber : [Perancangan]

## Use case approve nilai

Tabel 4.33 Spesifikasi *use case* approve nilai

<b>Nama use case</b>	Lihat nilai
<b>Aktor</b>	Laboran
<b>Tujuan</b>	Untuk menerbitkan nilai akhir.
<b>Deskripsi</b>	Approve nilai digunakan untuk menerbitkan nilai akhir.
<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case</i> lihat nilai.
<b>Pasca-kondisi</b>	Nilai akhir telah diterbitkan.
<b>Aliran utama</b>	
<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>
1. Laboran memilih menu approve nilai.	2. Sistem menerbitkan nilai akhir ke halaman informasi.

Sumber : [Perancangan]



Gambar 4.10 Spesifikasi *use case* untuk aktor dosen

Sumber : [Perancangan]

## Use case edit biodata

**Tabel 4.34** Spesifikasi *use case* edit biodata

<b>Nama use case</b>	Edit biodata	
<b>Aktor</b>	dosen	
<b>Tujuan</b>	Untuk melakukan perubahan data dosen.	
<b>Deskripsi</b>	Perubahan data laboran dilakukan dengan mengganti data user, data user meliputi: user id, password, nama, NIP, alamat, telepon dan email.	
<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case</i> login.	
<b>Pasca-kondisi</b>	Data dosen telah dirubah.	
<b>Aliran utama</b>		
	<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>
	1. Dosen memilih menu edit user.	2. Sistem menampilkan halaman perubahan biodata dosen. Halaman ini menampilkan 7 textfield yaitu: user id, password, nama, NIP, alamat, telepon dan email serta tombol simpan dan hapus.
	3. dosen memperbaharui data user baru dan menekan tombol simpan.	4. sistem memeriksa input yang dimasukan admin, apakah karakter yang dimasukan sah, apakah data yang dimasukan lengkap? Jika telah benar, maka selanjutnya sistem menyimpan data dosen tersebut kedalam basis data.
	5. Aktor dosen menekan tombol "Batal"	6. Kembali ke langkah 2.
<b>aliran alternative 1: data tidak lengkap</b>		
		7. sistem menampilkan pesan kesalahan dan kembali ke langkah 2.
<b>aliran alternative 2: karakter yang dimasukan tidak sah</b>		
		8. sistem menampilkan pesan kesalahan dan kembali ke langkah 2.

Sumber : [Perancangan]

## Use case lihat jadwal posttest

**Tabel 4.35** Spesifikasi *use case* lihat jadwal posttest

<b>Nama use case</b>	Lihat jadwal posttest
<b>Aktor</b>	dosen
<b>Tujuan</b>	Untuk melihat jadwal posttest yang telah diterbitkan oleh dosen.

<b>Deskripsi</b>	Lihat jadwal posttest menampilkan jadwal posttest yang dikeluarkan oleh pihak dosen yang bersangkutan.
<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case login</i> .
<b>Pasca-kondisi</b>	Sistem menampilkan jadwal posttest yang diterbitkan dosen.
Aliran utama	
Aksi dari aktor	Tanggapan dari sistem
1. Dosen memilih menu lihat jadwal posttest.	2. Sistem menampilkan jadwal posttest yang telah dikeluarkan dosen, jika jadwal belum diterbitkan maka sistem hanya menampilkan halaman kosong.

Sumber : [Perancangan]

### Use case tambah jadwal posttest

Tabel 4.36 Spesifikasi *use case* tambah jadwal posttest

<b>Nama use case</b>	tambah jadwal posttest
<b>Aktor</b>	dosen
<b>Tujuan</b>	Untuk menambah jadwal posttest.
<b>Deskripsi</b>	Penambahan jadwal posttest dilakukan dengan waktu dan peserta.
<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case login</i> .
<b>Pasca-kondisi</b>	jadwal posttest baru telah ditambahkan.
Aliran utama	
Aksi dari aktor	Tanggapan dari sistem
1. Dosen memilih menu tambah jadwal posttest.	2. Sistem menampilkan halaman penambahan jadwal posttest, halaman ini menampilkan form dengan beberapa buah textfield yaitu: waktu dan peserta posttest serta tombol simpan dan hapus.
3. Dosen mengisikan jadwal dan menekan tombol simpan.	4. Sistem memeriksa apakah data yang diisikan telah lengkap dan menggunakan karakter yang diijinkan? Jika benar maka sistem menyimpan jadwal posttest baru tersebut.
5. Dosen menekan tombol "Batal"	6. Sistem kembali ke langkah 2.
aliran alternative 1: data periode tidak lengkap	
	7. Sistem menampilkan pesan kesalahan dan kembali ke langkah 2.
aliran alternative 2: menggunakan karakter ilegal	



	8. Sistem menampilkan pesan kesalahan dan kembali ke langkah 2.
--	---

Sumber : [Perancangan]

### Use case edit jadwal postest

Tabel 4.37 Spesifikasi use case edit jadwal posttest

<b>Nama use case</b>	edit jadwal postest	
<b>Aktor</b>	dosen	
<b>Tujuan</b>	Untuk merubah jadwal postest yang telah diterbitkan.	
<b>Deskripsi</b>	Perubahan jadwal postest dilakukan dengan merubah waktu dan peserta.	
<b>Pra-kondisi</b>	Sistem telah dijalankan pada use case lihat jadwal postest.	
<b>Pasca-kondisi</b>	jadwal postest telah dirubah	
<b>Aliran utama</b>		
<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>	
1. Dosen memilih menu rubah jadwal postest.	2. Sistem menampilkan halaman perubahan jadwal postest, halaman ini menampilkan form dengan beberapa buah textfield dan nilainya yaitu: waktu dan peserta postest serta tombol simpan dan hapus.	
3. Dosen memperbaharui jadwal dan menekan tombol simpan.	4. Sistem memeriksa apakah data yang diisikan telah lengkap dan menggunakan karakter yang diijinkan? Jika benar maka sistem menyimpan perubahan beberapa buah textfield yaitu: waktu dan peserta postest tersebut.	
5. Dosen menekan tombol “batal.”	6. Sistem akan kembali ke langkah 2.	
<b>aliran alternative 1: data periode tidak lengkap</b>		
	7. Sistem menampilkan pesan kesalahan dan kembali ke langkah 2.	
<b>aliran alternative 2: menggunakan karakter ilegal</b>		
	8. Sistem menampilkan pesan kesalahan dan kembali ke langkah 2.	

Sumber : [Perancangan]



## Use case hapus jadwal posttest

Tabel 4.38 Spesifikasi *use case* hapus jadwal posttest

<b>Nama use case</b>	hapus jadwal posttest
<b>Aktor</b>	dosen
<b>Tujuan</b>	Untuk menghapus jadwal posttest.
<b>Deskripsi</b>	penghapusan jadwal posttest dilakukan dengan menekan tombol hapus pada list jadwal posttest.
<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case</i> lihat jadwal posttest.
<b>Pasca-kondisi</b>	jadwal posttest telah dihapus.
<b>Aliran utama</b>	
<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>
1. Dosen memilih menekan tombol hapus pada daftar jadwal posttest.	2. Sistem menampilkan pesan penghapusan jadwal posttest (pesan).
3. Dosen menekan tombol “ok” pada pesan.	4. Sistem menghapus informasi tersebut dari basis data sistem.
5. Dosen menekan tombol “batal” pada pesan.	6. Sistem akan kembali ke <i>use case</i> lihat jadwal.

Sumber : [Perancangan]

## Use case lihat nilai posttest

Tabel 4.39 Spesifikasi *use case* lihat nilai posttest

<b>Nama use case</b>	Lihat nilai posttest
<b>Aktor</b>	dosen
<b>Tujuan</b>	Untuk melihat nilai posttest yang telah diterbitkan oleh dosen.
<b>Deskripsi</b>	Lihat nilai posttest menampilkan nilai posttest yang dikeluarkan oleh pihak dosen yang bersangkutan.
<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case</i> login.
<b>Pasca-kondisi</b>	Sistem menampilkan nilai posttest yang diterbitkan dosen.
<b>Aliran utama</b>	
<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>
1. Dosen memilih menu lihat nilai posttest.	2. Sistem menampilkan nilai posttest yang telah dikeluarkan dosen, jika jadwal belum diterbitkan maka sistem hanya menampilkan halaman kosong.

Sumber : [Perancangan]

## Use case tambah nilai posttest

**Tabel 4.40** Spesifikasi *use case* tambah nilai posttest

<b>Nama use case</b>	Tambah nilai posttest	
<b>Aktor</b>	dosen	
<b>Tujuan</b>	Untuk menambah nilai posttest.	
<b>Deskripsi</b>	Penambahan jadwal posttest dilakukan dengan nilai dan peserta.	
<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case</i> login.	
<b>Pasca-kondisi</b>	nilai posttest baru telah ditambahkan.	
<b>Aliran utama</b>		
	<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>
	1. Dosen memilih menu tambah nilai posttest.	2. Sistem menampilkan halaman penambahan nilai posttest, halaman ini menampilkan form dengan beberapa buah textfield yaitu: nilai dan peserta posttest serta tombol simpan dan hapus.
	3. Dosen mengisikan nilai dan menekan tombol simpan.	4. Sistem memeriksa apakah data yang diisikan telah lengkap dan menggunakan karakter yang diijinkan? Jika benar maka sistem menyimpan nilai posttest baru tersebut.
	5. Dosen menekan tombol “batal”	6. Sistem akan kembali ke langkah 2.
<b>aliran alternative 1: data periode tidak lengkap</b>		
		7. Sistem menampilkan pesan kesalahan dan kembali ke langkah 2.
<b>aliran alternative 2: menggunakan karakter ilegal</b>		
		8. Sistem menampilkan pesan kesalahan dan kembali ke langkah 2.

Sumber : [Perancangan]

## Use case edit nilai posttest

**Tabel 4.41** Spesifikasi *use case* edit nilai posttest

<b>Nama use case</b>	Edit nilai posttest
<b>Aktor</b>	Dosen
<b>Tujuan</b>	Untuk merubah nilai posttest yang telah diterbitkan.
<b>Deskripsi</b>	Perubahan nilai posttest dilakukan dengan merubah nilai dan peserta.
<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case</i> lihat jadwal posttest.

<b>Pasca-kondisi</b>	jadwal posttest telah dirubah
<b>Aliran utama</b>	
<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>
1. Dosen memilih menu rubah nilai posttest.	2. Sistem menampilkan halaman perubahan nilai posttest, halaman ini menampilkan form dengan beberapa buah textfield dan nilainya yaitu: nilai dan peserta posttest serta tombol simpan dan hapus.
3. Dosen memperbaharui nilai dan menekan tombol simpan.	4. Sistem memeriksa apakah data yang diisikan telah lengkap dan menggunakan karakter yang diijinkan? Jika benar maka sistem menyimpan perubahan beberapa buah textfield yaitu: nilai dan peserta posttest tersebut.
5. Dosen menekan tombol “Batal.”	6. Sistem akan kembali ke langkah 2.
<b>aliran alternative 1: data periode tidak lengkap</b>	
	7. Sistem menampilkan pesan kesalahan dan kembali ke langkah 2.
<b>aliran alternative 2: menggunakan karakter ilegal</b>	
	8. Sistem menampilkan pesan kesalahan dan kembali ke langkah 2.

Sumber : [Perancangan]

### Use case hapus nilai posttest

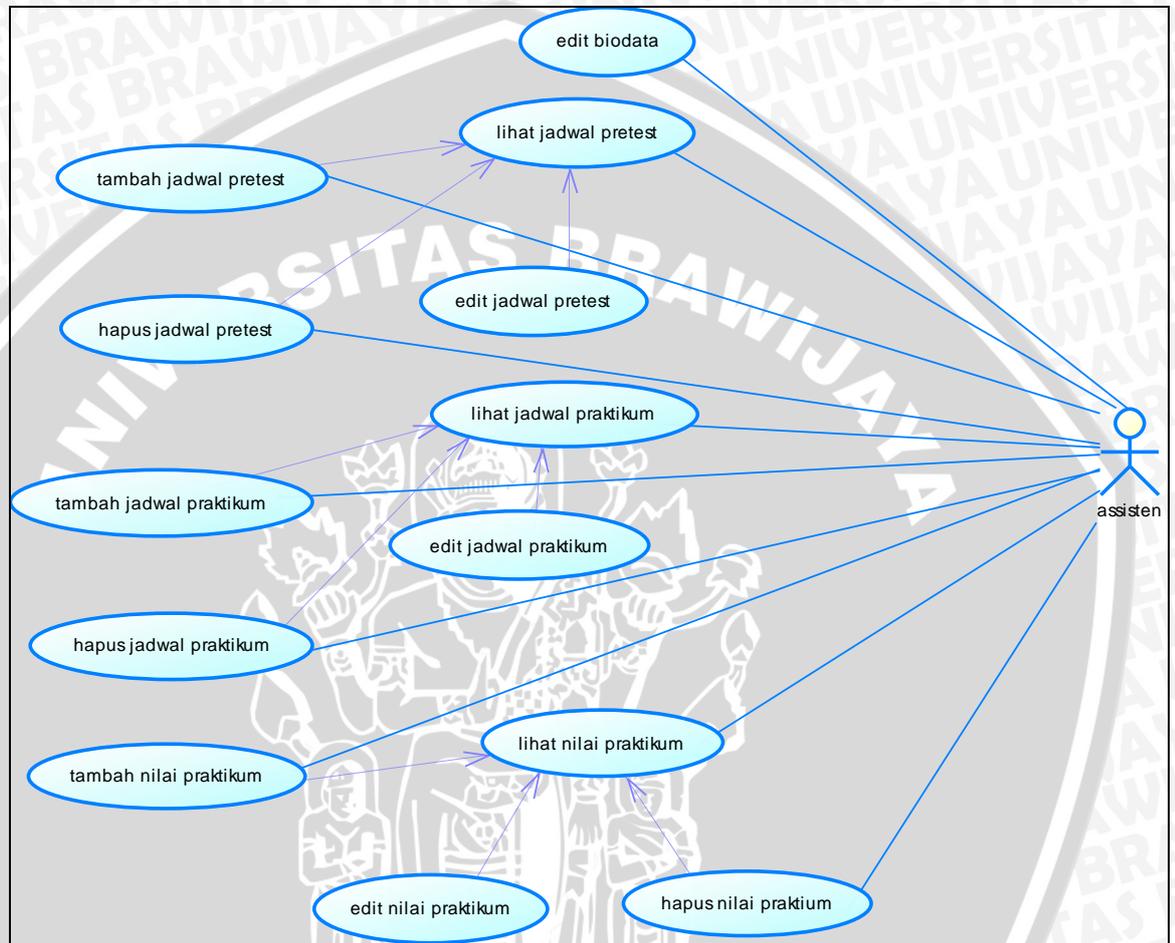
**Tabel 4.42** Spesifikasi *use case* hapus nilai posttest

<b>Nama use case</b>	hapus nilai posttest
<b>Aktor</b>	dosen
<b>Tujuan</b>	Untuk menghapus nilai posttest.
<b>Deskripsi</b>	penghapusan nilai posttest dilakukan dengan menekan tombol hapus pada list nilai posttest.
<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case</i> lihat nilai posttest.
<b>Pasca-kondisi</b>	jadwal posttest telah dihapus.
<b>Aliran utama</b>	
<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>
1. Dosen memilih menekan tombol hapus pada daftar nilai posttest.	2. Sistem menampilkan pesan penghapusan nilai posttest (pesan).
3. Dosen menekan tombol “ok” pada pesan.	4. Sistem menghapus informasi tersebut dari



	basis data sistem.
5. Dosen menekan tombol “batal”	6. Sistem akan kembali ke langkah 2.

Sumber : [Perancangan]



Gambar 4.11 Spesifikasi use case untuk aktor asisten

Sumber : [Perancangan]

### Use case edit biodata

Tabel 4.43 Spesifikasi use case edit biodata

<b>Nama use case</b>	Edit biodata
<b>Aktor</b>	Assisten
<b>Tujuan</b>	Untuk melakukan perubahan data assiten .
<b>Deskripsi</b>	Perubahan data laboran dilakukan dengan mengganti data user, data user meliputi: user id, password, nama, NIM, paket, alamat, telepon dan



	email.
<b>Pra-kondisi</b>	Sistem telah dijalankan pada use case login.
<b>Pasca-kondisi</b>	Data assiten telah dirubah.
<b>Aliran utama</b>	
<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>
1. Assisten memilih menu edit user.	2. sistem menampilkan halaman perubahan biodata asisten. Halaman ini menampilkan 8 textfield yaitu: user id, password, nama, NIM, paket, alamat, telepon dan email serta tombol simpan dan hapus.
3. asisten memperbaharui data asisten baru dan menekan tombol simpan.	4. sistem memeriksa input yang dimasukan asisten, apakah karakter yang dimasukan sah, apakah data yang dimasukan lengkap? Jika telah benar, maka selanjutnya sistem mnyimpan data asisten tersebut kedalam basis data.
5. Asisten menekan tombol “Batal”	6. Sistem akan kembali ke langkah 2.
<b>aliran alternative 1: data tidak lengkap</b>	
	7. sistem menampilkan pesan kesalahan dan kembali ke langkah 2.
<b>aliran alternative 2: karakter yang dimasukan tidak sah</b>	
	8. sistem menampilkan pesan kesalahan dan kembali ke langkah 2.

Sumber : [Perancangan]

### Use case lihat jadwal praktikum

**Tabel 4.44** Spesifikasi *use case* lihat jadwal praktikum

<b>Nama use case</b>	Lihat jadwal praktikum
<b>Aktor</b>	asisten
<b>Tujuan</b>	Untuk melihat jadwal praktikum yang telah diterbitkan oleh asisten.
<b>Deskripsi</b>	Lihat jadwal praktikum menampilkan jadwal praktikum yang dikeluarkan oleh pihak asisten yang bersangkutan.
<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case login</i> .
<b>Pasca-kondisi</b>	Sistem menampilkan jadwal praktikum yang diterbitkan asisten.
<b>Aliran utama</b>	
<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>
1. asisten memilih menu lihat jadwal	2. sistem menampilkan jadwal praktikum



praktikum.	yang telah dikeluarkan asisten, jika jadwal belum diterbitkan maka sistem hanya menampilkan halaman kosong.
------------	---

Sumber : [Perancangan]

### Use case tambah jadwal praktikum

Tabel 4.45 Spesifikasi *use case* tambah jadwal praktikum

<b>Nama use case</b>	tambah jadwal praktikum	
<b>Aktor</b>	asisten	
<b>Tujuan</b>	Untuk menambah jadwal praktikum.	
<b>Deskripsi</b>	Penambahan jadwal praktikum dilakukan dengan waktu.	
<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case login</i> .	
<b>Pasca-kondisi</b>	jadwal praktikum baru telah ditambahkan.	
<b>Aliran utama</b>		
<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>	
1. asisten memilih menu tambah jadwal praktikum.	2. sistem menampilkan halaman penambahan jadwal praktikum, halaman ini menampilkan form dengan beberapa buah textfield yaitu: waktu dan bab praktikum serta tombol simpan dan hapus.	
3. asisten mengisikan jadwal praktikum dan menekan tombol simpan.	4. sistem memeriksa apakah data yang diisikan telah lengkap dan menggunakan karakter yang diijinkan? Jika benar maka sistem menyimpan jadwal praktikum baru tersebut.	
5. Asisten menekan tombol "Batal."	6. Sistem akan kembali ke langkah 2.	
<b>aliran alternative 1: data periode tidak lengkap</b>		
	7. sistem menampilkan pesan kesalahan dan kembali ke langkah 2.	
<b>aliran alternative 2: menggunakan karakter ilegal</b>		
	8. sistem menampilkan pesan kesalahan dan kembali ke langkah 2.	

Sumber : [Perancangan]

### Use case edit jadwal praktikum

Tabel 4.46 Spesifikasi *use case* edit jadwal praktikum

<b>Nama use case</b>	edit jadwal praktikum
----------------------	-----------------------



<b>Aktor</b>	asisten	
<b>Tujuan</b>	Untuk merubah jadwal praktikum yang telah diterbitkan.	
<b>Deskripsi</b>	Perubahan jadwal praktikum dilakukan dengan merubah waktu dan bab praktikum.	
<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case</i> lihat jadwal praktikum.	
<b>Pasca-kondisi</b>	jadwal praktikum telah dirubah	
<b>Aliran utama</b>		
	<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>
	1. asisten memilih menu rubah jadwal praktikum.	2. sistem menampilkan halaman perubahan jadwal praktikum, halaman ini menampilkan form dengan beberapa buah textfield dan nilainya yaitu: waktu dan bab praktikum serta tombol simpan dan hapus.
	3. asisten memperbaharui jadwal dan menekan tombol simpan.	4. sistem memeriksa apakah data yang diisikan telah lengkap dan menggunakan karakter yang diijinkan? Jika benar maka sistem menyimpan perubahan beberapa buah textfield yaitu: waktu dan bab praktikum tersebut.
	5. Asisten menekan tombol “Batal.”	6. Sistem akan kembali ke langkah 2.
<b>aliran alternative 1: data periode tidak lengkap</b>		
		7. sistem menampilkan pesan kesalahan dan kembali ke langkah 2.
<b>aliran alternative 2: menggunakan karakter ilegal</b>		
		8. sistem menampilkan pesan kesalahan dan kembali ke langkah 2.

Sumber : [Perancangan]

### Use case hapus jadwal praktikum

**Tabel 4.47** Spesifikasi *use case* hapus jadwal praktikum

<b>Nama use case</b>	hapus jadwal praktikum
<b>Aktor</b>	asisten
<b>Tujuan</b>	Untuk menghapus jadwal praktikum.
<b>Deskripsi</b>	penghapusan jadwal praktikum dilakukan dengan menekan tombol hapus pada list jadwal praktikum.
<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case</i> lihat jadwal praktikum.
<b>Pasca-kondisi</b>	jadwal praktikum telah dihapus.



Aliran utama	
Aksi dari aktor	Tanggapan dari sistem
1. asisten memilih menekan tombol hapus pada daftar jadwal praktikum.	2. Sistem menampilkan pesan penghapusan jadwal praktikum (pesan).
3. asisten menekan tombol “ok” pada pesan.	4. Sistem menghapus informasi tersebut dari basis data sistem.
5. Asisten menekan tombol “Batal.”	6. Sistem akan kembali ke langkah 2.

Sumber : [Perancangan]

### Use case lihat nilai praktikum

Tabel 4.48 Spesifikasi use case lihat nilai praktikum

<b>Nama use case</b>	Lihat nilai praktikum
<b>Aktor</b>	asisten
<b>Tujuan</b>	Untuk melihat nilai praktikum yang telah diterbitkan oleh asisten.
<b>Deskripsi</b>	Lihat nilai postest menampilkan nilai praktikum yang dikeluarkan oleh asisten yang bersangkutan.
<b>Pra-kondisi</b>	Sistem telah dijalankan pada use case login.
<b>Pasca-kondisi</b>	Sistem menampilkan nilai praktikum yang diterbitkan dosen.
Aliran utama	
Aksi dari aktor	Tanggapan dari sistem
1. asisten memilih menu lihat nilai praktikum.	2. sistem menampilkan nilai praktikum yang telah dikeluarkan asisten, jika jadwal belum diterbitkan makan sistem hanya menampilkan halaman kosong.

Sumber : [Perancangan]

### Use case tambah nilai praktikum

Tabel 4.49 Spesifikasi use case tambah nilai praktikum

<b>Nama use case</b>	tambah nilai praktikum
<b>Aktor</b>	asisten
<b>Tujuan</b>	Untuk menambah nilai praktikum.
<b>Deskripsi</b>	Penambahan jadwal praktikum dilakukan dengan nilai dan peserta.
<b>Pra-kondisi</b>	Sistem telah dijalankan pada use case login.
<b>Pasca-kondisi</b>	nilai praktikum baru telah ditambahkan.
Aliran utama	
Aksi dari aktor	Tanggapan dari sistem
1. asisten memilih menu tambah nilai	2. sistem menampilkan halaman penambahan



praktikum.	nilai praktikum, halaman ini menampilkan form dengan beberapa buah textfield yaitu: nilai dan peserta praktikum serta tombol simpan dan hapus.
3. asisten mengisikan nilai dan menekan tombol simpan.	4. sistem memeriksa apakah data yang diisikan telah lengkap dan menggunakan karakter yang diijinkan? Jika benar maka sistem menyimpan nilai praktikum baru tersebut.
5. Asisten menekan tombol batal	6. Sistem akan kembali ke langkah 2.
<b>aliran alternative 1: data periode tidak lengkap</b>	
	7. sistem menampilkan pesan kesalahan dan kembali ke langkah 2.
<b>aliran alternative 2: menggunakan karakter ilegal</b>	
	8. sistem menampilkan pesan kesalahan dan kembali ke langkah 2.

Sumber : [Perancangan]

### Use case edit nilai praktikum

Tabel 4.50 Spesifikasi *use case* edit nilai praktikum

<b>Nama use case</b>	edit nilai praktikum	
<b>Aktor</b>	asisten	
<b>Tujuan</b>	Untuk merubah nilai praktikum yang telah diterbitkan.	
<b>Deskripsi</b>	Perubahan nilai praktikum dilakukan dengan merubah nilai dan peserta.	
<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case</i> lihat jadwal praktikum.	
<b>Pasca-kondisi</b>	jadwal praktikum telah dirubah	
<b>Aliran utama</b>		
<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>	
1. asisten memilih menu rubah nilai praktikum.	2. sistem menampilkan halaman perubahan nilai praktikum, halaman ini menampilkan form dengan beberapa buah textfield dan nilainya yaitu: nilai dan peserta praktikum serta tombol simpan dan hapus.	
3. asisten memperbaharui nilai dan menekan tombol simpan.	4. sistem memeriksa apakah data yang diisikan telah lengkap dan menggunakan karakter yang diijinkan? Jika benar maka sistem menyimpan perubahan beberapa	



	buah textfield yaitu: nilai dan peserta praktikum tersebut.
5. Asisten menekan tombol “Batal”	6. Sistem akan kembali ke langkah 2.
<b>aliran alternative 1: data periode tidak lengkap</b>	
	7. sistem menampilkan pesan kesalahan dan kembali ke langkah 2.
<b>aliran alternative 2: menggunakan karakter ilegal</b>	
	8. sistem menampilkan pesan kesalahan dan kembali ke langkah 2.

Sumber : [Perancangan]

### Use case hapus nilai praktikum

Tabel 4.51 Spesifikasi use case hapus nilai praktikum

<b>Nama use case</b>	hapus nilai praktikum
<b>Aktor</b>	asisten
<b>Tujuan</b>	Untuk menghapus nilai praktikum.
<b>Deskripsi</b>	penghapusan nilai praktikum dilakukan dengan menekan tombol hapus pada list nilai praktikum.
<b>Pra-kondisi</b>	Sistem telah dijalankan pada use case lihat nilai praktikum.
<b>Pasca-kondisi</b>	jadwal praktikum telah dihapus.
<b>Aliran utama</b>	
<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>
1. asisten memilih menekan tombol hapus pada daftar nilai praktikum.	2. sistem menampilkan pesan penghapusan nilai praktikum (pesan).
3. asisten menekan tombol “ok” pada pesan.	4. sistem menghapus informasi tersebut dari basis data sistem.
5. Asisten menekan tombol “Batal.”	6. Sistem akan kembali ke langkah 2.

Sumber : [Perancangan]

### Use case lihat jadwal pretest

Tabel 4.52 Spesifikasi use case lihat jadwal pretest

<b>Nama use case</b>	Lihat jadwal pretest
<b>Aktor</b>	asisten
<b>Tujuan</b>	Untuk melihat jadwal pretest yang telah diterbitkan oleh asisten.
<b>Deskripsi</b>	Lihat jadwal pretest menampilkan jadwal pretest yang dikeluarkan oleh pihak asisten yang bersangkutan.
<b>Pra-kondisi</b>	Sistem telah dijalankan pada use case login.



<b>Pasca-kondisi</b>	Sistem menampilkan jadwal pretest yang diterbitkan asisten.
<b>Aliran utama</b>	
<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>
1. asisten memilih menu lihat jadwal pretest.	2. sistem menampilkan jadwal pretest yang telah dikeluarkan asisten, jika jadwal belum diterbitkan maka sistem hanya menampilkan halaman kosong.

Sumber : [Perancangan]

### Use case tambah jadwal pretest

Tabel 4.53 Spesifikasi *use case* tambah jadwal pretest

<b>Nama use case</b>	tambah jadwal pretest	
<b>Aktor</b>	asisten	
<b>Tujuan</b>	Untuk menambah jadwal pretest.	
<b>Deskripsi</b>	Penambahan jadwal pretest dilakukan dengan waktu.	
<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case login</i> .	
<b>Pasca-kondisi</b>	jadwal pretest baru telah ditambahkan.	
<b>Aliran utama</b>		
<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>	
1. asisten memilih menu tambah jadwal praktiku pretest m.	2. sistem menampilkan halaman penambahan jadwal pretest, halaman ini menampilkan form dengan beberapa buah textfield yaitu: waktu dan tempat pretest serta tombol simpan dan hapus.	
3. asisten mengisikan jadwal pretest dan menekan tombol simpan.	4. sistem memeriksa apakah data yang diisikan telah lengkap dan menggunakan karakter yang diijinkan? Jika benar maka sistem menyimpan jadwal pretest baru tersebut.	
5. Asisten menekan tombol "Batal."	6. Sistem akan kembali ke langkah 2.	
<b>aliran alternative 1: data periode tidak lengkap</b>		
	7. sistem menampilkan pesan kesalahan dan kembali ke langkah 2.	
<b>aliran alternative 2: menggunakan karakter ilegal</b>		
	8. sistem menampilkan pesan kesalahan dan kembali ke langkah 2.	

Sumber : [Perancangan]



## Use case edit jadwal pretest

Tabel 4.54 Spesifikasi *use case* edit jadwal pretest

<b>Nama use case</b>	edit jadwal pretest
<b>Aktor</b>	asisten
<b>Tujuan</b>	Untuk merubah jadwal pretest yang telah diterbitkan.
<b>Deskripsi</b>	Perubahan jadwal pretest dilakukan dengan merubah waktu dan tempat pretest.
<b>Pra-kondisi</b>	Sistem telah dijalankan pada <i>use case</i> lihat jadwal pretest.
<b>Pasca-kondisi</b>	jadwal pretest telah dirubah
<b>Aliran utama</b>	
<b>Aksi dari aktor</b>	<b>Tanggapan dari sistem</b>
1. asisten memilih menu rubah jadwal pretest.	2. sistem menampilkan halaman perubahan jadwal pretest, halaman ini menampilkan form dengan beberapa buah textfield dan nilainya yaitu: waktu dan tempat pretest serta tombol simpan dan hapus.
3. asisten memperbaharui jadwal pretest dan menekan tombol simpan.	4. sistem memeriksa apakah data yang diisikan telah lengkap dan menggunakan karakter yang diijinkan? Jika benar maka sistem menyimpan perubahan beberapa buah textfield yaitu: waktu dan bab praktikum tersebut.
5. Asisten menkan tombol “Batal”	6. Sistem akan kembali ke langkah 2.
<b>aliran alternative 1: data periode tidak lengkap</b>	
	7. sistem menampilkan pesan kesalahan dan kembali ke langkah 2.
<b>aliran alternative 2: menggunakan karakter ilegal</b>	
	8. sistem menampilkan pesan kesalahan dan kembali ke langkah 2.

Sumber : [Perancangan]

### 4.2. Perancangan Berorientasi Obyek

Perancangan berorientasi obyek terbagi dalam dua aktivitas utama: perancangan sistem dan perancangan obyek. Perancangan sistem mendefinisikan arsitektur produk (fungsi sistem dan kelas yang dienkapsulasi dalam subsistem). Perancangan sistem difokuskan pada spesifikasi dari tiga komponen: *user interface*, fungsi manajemen data, dan fasilitas manajemen kerja. Perancangan

obyek difokuskan pada detail internal dari kelas individu dan skema penyampaian pesan [PRE-05].

Proses perancangan perangkat lunak dilakukan dalam dua tahap, yaitu perancangan umum dan perancangan detail. Perancangan umum menggambarkan relasi antar paket (*package*) dan kelas (*class*) sebagai pemodelan sistem secara keseluruhan. Perancangan detail menggunakan *class diagram* dan *sequence diagram* sebagai pemodelan perangkat lunak. Perancangan detail melakukan perancangan terhadap pola hubungan antar komponen-komponen detail (dalam konteks berorientasi objek adalah kelas dan objek) sehingga mampu membentuk sebuah fungsi kerja yang mampu memberikan pelayanan terhadap kebutuhan aktor.

#### 4.2.1. Perancangan Umum

Perancangan umum menggambarkan relasi antar paket (*package*) dan kelas (*class*) sebagai pemodelan sistem secara keseluruhan. Perancangan umum memberikan pandangan keseluruhan sistem tanpa melihat detail dari masing-masing subsistem yang ada.

##### 4.2.1.1 Arsitektur Sistem

Sistem Informasi Manajemen Teknik Elektro Universitas Brawijaya dibagi menjadi 3 buah *layer*, yaitu :

1. *Presentation layer*
2. *Business logic layer*
3. *Data source layer*

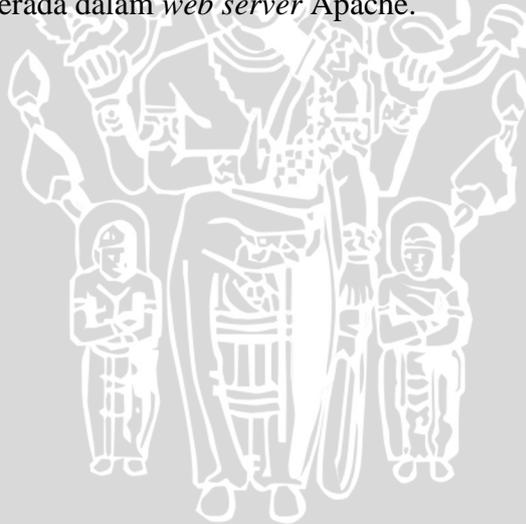
*Presentation layer* adalah suatu lapisan yang bertanggung jawab untuk memproses masukan dari aktor atau menampilkan sesuatu pada aktor. *Presentation layer* pada Sistem Informasi Manajemen Teknik Elektro Universitas Brawijaya menggunakan HTML, CSS dan PHP *script*.

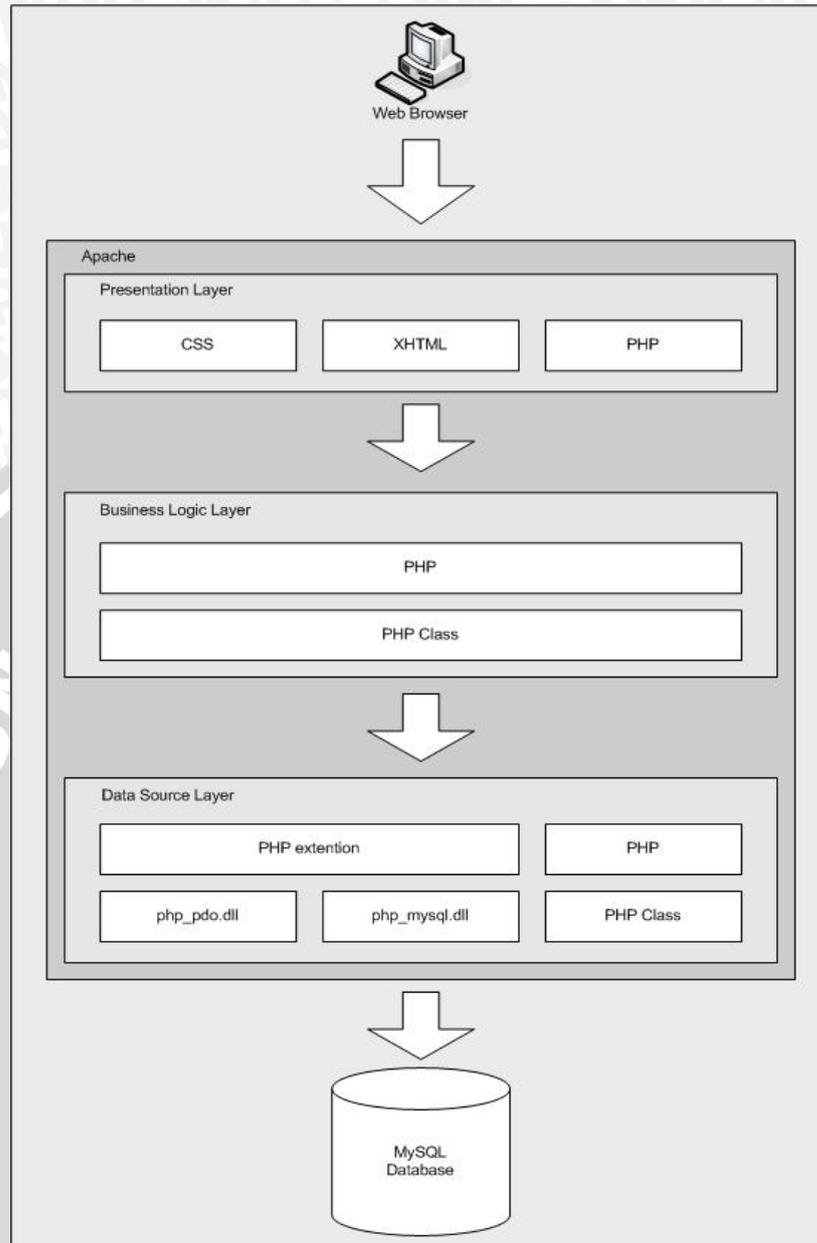
*Business logic layer* adalah suatu lapisan letak logika-logika dari sistem (bisnis proses). *Business logic layer* pada Sistem Informasi Manajemen Teknik Elektro Universitas Brawijaya menggunakan PHP.

*Data source layer* adalah suatu lapisan yang bertanggung jawab untuk berkomunikasi dengan database. *Data source layer* pada Sistem Informasi Manajemen Teknik Elektro Universitas Brawijaya menggunakan PHP.

Aktor yang mengakses sistem ini akan menggunakan *web browser*. *Web browser* akan mengirimkan *request* pada *server*. *Request* ini akan ditanggapi oleh *server* dengan menampilkan sesuatu dalam format HTML. Tanggapan *server* ini dihasilkan oleh *presentation layer*. *Presentation layer* dalam memproses *request* dari aktor akan menggunakan *business logic layer* untuk memproses bisnis proses yang bersesuaian dengan *request* aktor. *Business logic layer* akan menggunakan *data source layer* bila *business logic layer* membutuhkan suatu data yang berhubungan dengan database.

Gambar berikut menjelaskan arsitektur Sistem Informasi Manajemen Teknik Elektro Universitas Brawijaya yang terdiri dari tiga *layer*. Ketiga *layer* tersebut adalah *presentation layer*, *business logic layer* dan *data access layer*. Ketiga *layer* tersebut berada dalam *web server Apache*.





Gambar 4.12 Arsitektur Sistem

Sumber : [Perancangan]

#### 4.2.2. Perancangan Detil

Pada perancangan detil dilakukan spesifikasi dari tipe-tipe atribut, bagaimana fungsi-fungsi beroperasi, dan bagaimana suatu obyek berhubungan dengan obyek lainnya. Perancangan detil dimodelkan dengan *class diagram* dan *sequence diagram*.



#### 4.2.2.1 *Class Diagram*

Kelas adalah kumpulan dari obyek yang berbagi atribut, operasi relasi dan semantik yang sama. *Class diagram* adalah representasi berbentuk diagram dari kelas-kelas yang membentuk perangkat lunak [BOR-05].

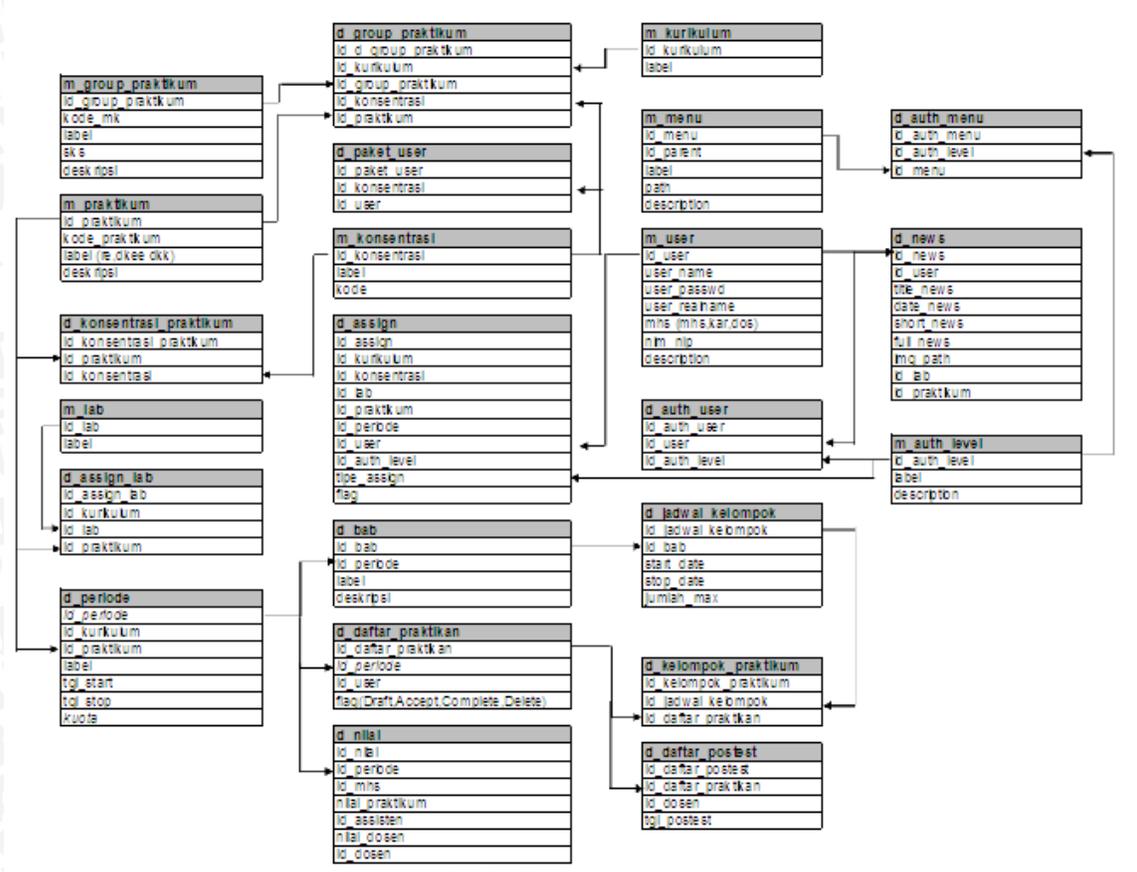
#### 4.2.2.2 **Perancangan Basis Data**

Sistem Informasi Manajemen Laboratorium Teknik Elektro Universitas Brawijaya memerlukan sebuah basis data manajemen laboratorium. Basis data berfungsi sebagai tempat menyimpan data-data manajemen laboratorium. Pemodelan basis data dapat dilakukan dengan menggunakan *Entity-Relationship Diagram*. Gambar 4.35 adalah *physical data model* dari Sistem Informasi Manajemen Laboratorium Teknik Elektro Universitas Brawijaya. *Physical data model* menggambarkan hubungan 24 tabel secara fisikal.









Gambar 4.15 ERD Sistem Informasi Manajemen Laboratorium Teknik Elektro Universitas Brawijaya

Sumber : [Perancangan]

#### 4.2.2.2.1 Data Object Description

Data Object Description menjelaskan secara rinci mengenai atribut-atribut yang dimiliki oleh masing-masing tabel yang ada pada basis data Sistem Informasi Manajemen Laboratorium Teknik Elektro Universitas Brawijaya. Entitas tabel yang ada pada basis data Sistem Informasi Manajemen Laboratorium Teknik Elektro Universitas Brawijaya masing-masing memiliki struktur tabel basis data.

Rancangan struktur tabel terdiri atas 5 (lima) kolom, yaitu *key*, *column name*, *data type*, *size*, *allow*. Kolom *key* menjelaskan status *field* yang dijadikan *Primary Key* (PK) atau *Foreign Key* (FK). *Column name* berisi nama-nama *field* yang akan digunakan pada masing-masing tabel. *Data type* menjelaskan tipe data yang digunakan oleh setiap *field*. *Size* berisi panjang dari tipe data yang digunakan. *Allow* berarti data yang dimasukkan

apakah boleh bernilai *Null* (kosong). N berarti tidak diperbolehkan bernilai *null*. Y berarti boleh bernilai *null*.

### 1. Data Object Description untuk Tabel d\_assign

**Tabel 4.55** Data Object Description untuk Tabel **d\_assign**

Nama Tabel : d\_assign

KEY	COLUMN NAME	DATA TYPE	SIZE	ALLOW
PK	id_assign	integer	10	N
FK	id_kurikulum	integer	10	Y
FK	id_kosentrasi	integer	10	Y
FK	id_lab	Integer	10	Y
FK	id_praktikum	integer	10	Y
FK	id_periode	Integer	10	Y
FK	id_user	Integer	10	Y
FK	id_auth_level	integer	10	Y
	flag	Enum (outstanding, accept, reject, delete)		Y

Sumber : [Perancangan]

### 2. Data Object Description untuk Tabel d\_assign\_lab

**Tabel 4.56** Data Object Description untuk Tabel **d\_assign\_lab**

Nama Tabel : d\_assign\_lab

KEY	COLUMN NAME	DATA TYPE	SIZE	ALLOW
PK	id_assign_lab	integer	10	N
FK	id_kurikulum	integer	10	Y
FK	id_lab	Integer	10	Y
FK	id_praktikum	integer	10	Y

Sumber : [Perancangan]

### 3. Data Object Description untuk Tabel d\_auth\_menu

**Tabel 4.57** Data Object Description untuk Tabel **d\_auth\_menu**

Nama Tabel : d\_auth\_menu

KEY	COLUMN NAME	DATA TYPE	SIZE	ALLOW
-----	-------------	-----------	------	-------

PK	id_auth_menu	integer	10	N
FK	id_auth_level	integer	10	Y
FK	id_menu	Integer	10	Y

Sumber : [Perancangan]

#### 4. Data Object Description untuk Tabel d\_auth\_user

**Tabel 4.58** Data Object Description untuk Tabel **d\_auth\_user**

Nama Tabel : d\_auth\_user

KEY	COLUMN NAME	DATA TYPE	SIZE	ALLOW
PK	id_auth_user	integer	10	N
FK	id_user	integer	10	Y
FK	id_auth_level	Integer	10	Y

Sumber : [Perancangan]

#### 5. Data Object Description untuk Tabel d\_bab

**Tabel 4.59** Data Object Description untuk Tabel **d\_bab**

Nama Tabel : d\_bab

KEY	COLUMN NAME	DATA TYPE	SIZE	ALLOW
PK	id_bab	integer	10	N
FK	id_periode	integer	10	Y
FK	id_praktikum	Integer	10	Y
	label	Varchar	255	Y
	deskripsi	varchar	255	

Sumber : [Perancangan]

#### 6. Data Object Description untuk Tabel d\_group\_praktikum

**Tabel 4.60** Data Object Description untuk Tabel **d\_group\_praktikum**

Nama Tabel : d\_group\_praktikum

KEY	COLUMN NAME	DATA TYPE	SIZE	ALLOW
PK	id_d_group_praktikum	integer	10	N
FK	id_praktikum	integer	10	Y
FK	id_group_praktikum	Integer	10	Y
FK	Id_kosentrasi	Integer	10	Y
FK	id_praktikum	integer	10	Y

Sumber : [Perancangan]

**7. Data Object Description untuk Tabel d\_jadwal\_mhs**

**Tabel 4.61 Data Object Description untuk Tabel d\_jadwal\_mhs**

Nama Tabel : d\_jadwal\_mhs

KEY	COLUMN NAME	DATA TYPE	SIZE	ALLOW
PK	id_jadwal_mhs	integer	10	N
FK	id_jadwal_praktikum	integer	10	Y
FK	id_praktikan	Integer	10	Y
	start_date	timestamp		Y
	stop_date	timestamp		Y

Sumber : [Perancangan]

**8. Data Object Description untuk Tabel d\_jadwal\_praktikum**

**Tabel 4.62 Data Object Description untuk Tabel d\_jadwal\_praktikum**

Nama Tabel : d\_jadwal\_praktikum

KEY	COLUMN NAME	DATA TYPE	SIZE	ALLOW
PK	id_jadwal_praktikum	integer	10	N
FK	id_bab	integer	10	Y
	start_date	timestamp		Y
	stop_date	timestamp		Y
	jumlah_max	tinyint	3	N

Sumber : [Perancangan]

**9. Data Object Description untuk Tabel d\_jadwal\_umum**

**Tabel 4.63 Data Object Description untuk Tabel d\_jadwal\_umum**

Nama Tabel : d\_jadwal\_umum

KEY	COLUMN NAME	DATA TYPE	SIZE	ALLOW
PK	id_jadwal_umum	integer	10	N
FK	id_periode	integer	10	N
	type_jadwal	Enum (pre, post)		N
	Start_date	timestamp		Y
	Stop_date	timestamp		Y

Sumber : [Perancangan]

**10. Data Object Description untuk Tabel d\_kosentrasi\_praktikum**

**Tabel 4.1 Data Object Description untuk Tabel d\_kosentrasi\_praktikum**

Nama Tabel : d\_kosentrasi\_praktikum

KEY	COLUMN NAME	DATA TYPE	SIZE	ALLOW
PK	id_kosentrasi_praktikum	integer	10	N
FK	id_kurikulum	integer	10	Y
FK	id_kosentrasi	Integer	10	Y
FK	id_praktikum	integer	10	Y

Sumber : [Perancangan]

**11. Data Object Description untuk Tabel d\_lab\_praktikum**

**Tabel 4.65 Data Object Description untuk Tabel d\_lab\_praktikum**

Nama Tabel : d\_lab\_praktikum

KEY	COLUMN NAME	DATA TYPE	SIZE	ALLOW
PK	id_lab_praktikum	integer	10	N
FK	id_lab	integer	10	Y
FK	id_praktikum	integer	10	Y

Sumber : [Perancangan]

**12. Data Object Description untuk Tabel d\_news**

**Tabel 4.66 Data Object Description untuk Tabel d\_news**

Nama Tabel : d\_news

KEY	COLUMN NAME	DATA TYPE	SIZE	ALLOW
PK	id_news	integer	10	N
FK	id_user	integer	10	N
	tittle_news	Varchar	255	N
	date_news	Timestamp	10	N
	short_news	Tinytext		N
	full_news	Text		N
	img_path	Varchar	255	N
FK	id_lab	Integer	10	Y
FK	id_praktikum	integer	10	Y

Sumber : [Perancangan]

### 13. Data Object Description untuk Tabel d\_nilai

**Tabel 4.67** Data Object Description untuk Tabel d\_nilai

Nama Tabel : d\_nilai

KEY	COLUMN NAME	DATA TYPE	SIZE	ALLOW
PK	id_nilai	integer	10	N
FK	id_periode	integer	10	N
FK	id_user	Integer	10	N
	prosen_praktikum	Decimal	10	N
	nilai_praktikum	Decimal		N
FK	id_asisten	Integer	10	N
FK	id_dosen	Integer	10	N
	prosen_dosen	Decimal		N
	nilai_dosen	decimal		N

Sumber : [Perancangan]

### 14. Data Object Description untuk Tabel d\_paket\_user

**Tabel 4.68** Data Object Description untuk Tabel d\_paket\_user

Nama Tabel : d\_paket\_user

KEY	COLUMN NAME	DATA TYPE	SIZE	ALLOW
PK	id_paket_user	integer	10	N
FK	id_kosentrasi	integer	10	N
FK	id_user	Integer	10	N

Sumber : [Perancangan]

### 15. Data Object Description untuk Tabel d\_periode

**Tabel 4.69** Data Object Description untuk Tabel d\_periode

Nama Tabel : d\_periode

KEY	COLUMN NAME	DATA TYPE	SIZE	ALLOW
PK	id_periode	integer	10	N
FK	id_kurikulum	integer	10	Y
FK	id_praktikum	Integer	10	Y
	label	Varchar	255	Y

	start_date	Timestamp		Y
	stop_date	timestamp		Y

Sumber : [Perancangan]

## 16. Data Object Description untuk Tabel m\_auth\_level

Tabel 4.70 Data Object Description untuk Tabel m\_auth\_level

Nama Tabel : m\_auth\_level

KEY	COLUMN NAME	DATA TYPE	SIZE	ALLOW
PK	id_auth_level	integer	10	N
	label	Varchar	255	Y
	description	Text		Y

Sumber : [Perancangan]

## 17. Data Object Description untuk Tabel m\_config

Tabel 4.71 Data Object Description untuk Tabel m\_config

Nama Tabel : m\_config

KEY	COLUMN NAME	DATA TYPE	SIZE	ALLOW
	Config_var	Text		
	id_kurikulum	Text		

Sumber : [Perancangan]

## 18. Data Object Description untuk Tabel m\_group\_praktikum

Tabel 4.72 Data Object Description untuk Tabel m\_group\_praktikum

Nama Tabel : m\_group\_praktikum

KEY	COLUMN NAME	DATA TYPE	SIZE	ALLOW
PK	id_group_praktikum	integer	10	N
	Kode_mk	Varchar	255	N
	Label	Varchar	255	N
	sks	tiny	3	N
	deskripsi	Text		Y

Sumber : [Perancangan]

## 19. Data Object Description untuk Tabel m\_kosentrasi

Tabel 4.73 Data Object Description untuk Tabel m\_kosentrasi

Nama Tabel : m\_kosentrasi

KEY	COLUMN NAME	DATA TYPE	SIZE	ALLOW
PK	id_kosentrasi	integer	10	N
	label	Varchar	255	Y
	kode	Varchar	1	Y

Sumber : [Perancangan]

## 20. Data Object Description untuk Tabel m\_kurikulum

**Tabel 4.74** Data Object Description untuk Tabel m\_kurikulum

Nama Tabel : m\_kurikulum

KEY	COLUMN NAME	DATA TYPE	SIZE	ALLOW
PK	id_kurikulum	integer	10	N
	label	Varchar	255	Y

Sumber : [Perancangan]

## 21. Data Object Description untuk Tabel m\_lab

**Tabel 4.75** Data Object Description untuk Tabel m\_lab

Nama Tabel : m\_lab

KEY	COLUMN NAME	DATA TYPE	SIZE	ALLOW
PK	id_lab	integer	10	N
	Label	Varchar	255	Y

Sumber : [Perancangan]

## 22. Data Object Description untuk Tabel m\_menu

**Tabel 4.76** Data Object Description untuk Tabel m\_menu

Nama Tabel : m\_menu

KEY	COLUMN NAME	DATA TYPE	SIZE	ALLOW
PK	id_menu	integer	10	N
FK	id_parent	integer	10	Y
	label	Varchar	255	Y
	path	Varchar	255	Y
	deskripsi	Text		Y

Sumber : [Perancangan]

## 23. Data Object Description untuk Tabel m\_praktikum

**Tabel 4.77 Data Object Description** untuk Tabel **m\_praktikum**

Nama Tabel : m\_praktikum

KEY	COLUMN NAME	DATA TYPE	SIZE	ALLOW
PK	id_praktikum	integer	10	N
	kode_praktikum	Varchar	255	N
	label	Varchar	255	N
	deskripsi	Text		N

Sumber : [Perancangan]

**24. Data Object Description** untuk Tabel **m\_user****Tabel 4.78 Data Object Description** untuk Tabel **m\_user**

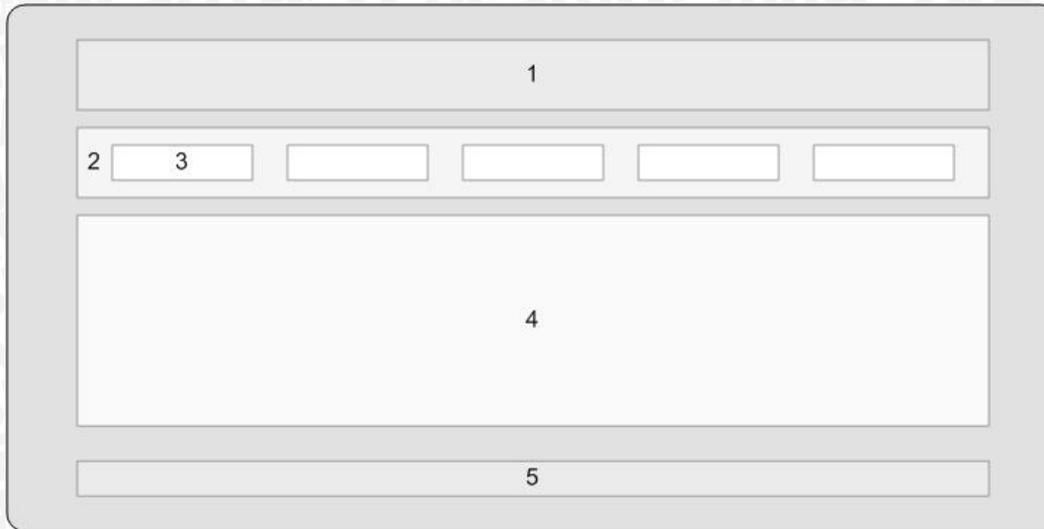
Nama Tabel : m\_user

KEY	COLUMN NAME	DATA TYPE	SIZE	ALLOW
PK	id_user	integer	10	N
	user_name	Varchar	255	Y
	password	Varchar	255	Y
	User_realname	Varchar	255	Y
	mhs	Enum ( 'mahasiswa', 'dosen', 'karyawan' )		N
	nip_nim	Varchar	255	Y
	deskripsi	Text		Y
	Flag	Enum (complete, delete)		Y

Sumber : [Perancangan]

**4.2.2.3 Perancangan User Interface**

*User interface* merupakan tampilan dimana *user* berinteraksi dengan sistem. Sistem Informasi Manajemen Teknik Elektro Universitas Brawijaya memiliki *user* dengan tingkat akses berbeda-beda sehingga *user interface* untuk tiap *user* sesuai dengan level atau tingkat akses dari *user* tersebut. Desain dasar untuk *user interface* dari Sistem Informasi Manajemen Teknik Elektro Universitas Brawijaya ditunjukkan oleh gambar 4.37.



**Gambar 4.16** Desain Dasar *User Interface*

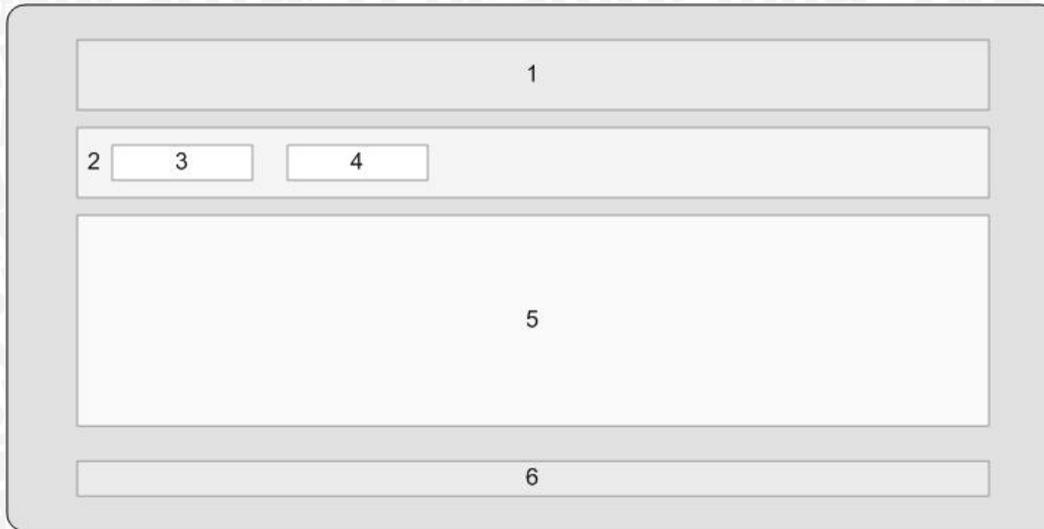
**Sumber :** [Perancangan]

Desain dasar untuk *user interface* dari Sistem Informasi Manajemen Teknik Elektro Universitas Brawijaya terdiri dari beberapa bagian sebagai berikut :

1. *Header*, berisi nama sistem.
2. *Menu Bar*, berisi menu-menu dari sistem.
3. Menu, merupakan menu dari sistem.
4. *Content*, berisi *content* dari halaman utama atau *content* dari menu.
5. *Footer*, berisi nama sistem, *copyright* maupun informasi-informasi lainnya.

#### 4.2.2.3.1 Perancangan Halaman Utama

Halaman utama dari *user interface* Sistem Informasi Manajemen Teknik Elektro Universitas Brawijaya merupakan halaman pertama yang akan diakses oleh *user* yang belum melakukan proses *login* ke dalam sistem.



**Gambar 4.17** Desain Halaman Utama

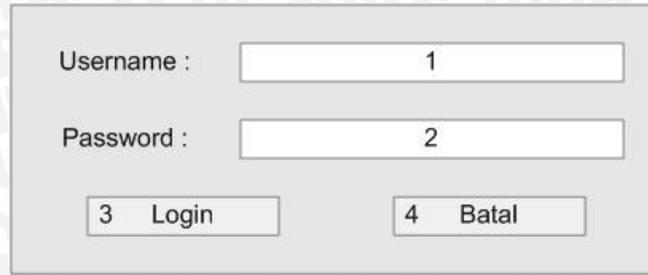
**Sumber :** [Perancangan]

Desain dasar untuk halaman utama dari Sistem Informasi Manajemen Teknik Elektro Universitas Brawijaya terdiri dari beberapa bagian sebagai berikut :

1. *Header*, berisi nama sistem.
2. *Menu Bar*, berisi menu-menu dari sistem.
3. Menu *News*, menu yang berisi informasi berita tentang seputar Sistem Informasi Manajemen Teknik Elektro Universitas Brawijaya.
4. Menu *Login*, menu yang berisi *form login*.
5. *Content*, berisi *welcome message* dan informasi sekilas tentang Sistem Informasi Manajemen Teknik Elektro Universitas Brawijaya.
6. *Footer*, berisi nama sistem, *copyright* maupun informasi-informasi lainnya.

#### 4.2.2.3.2 Perancangan *Form Login*

*Form login* adalah *user interface* yang digunakan untuk melakukan proses *login* ke dalam sistem. Desain *form login* ditunjukkan oleh gambar berikut



Username :

Password :

3 Login      4 Batal

**Gambar 4.18** Desain *Form Login*

**Sumber :** [Perancangan]

Desain dasar untuk halaman utama dari Sistem Informasi Manajemen Teknik Elektro Universitas Brawijaya terdiri dari beberapa bagian sebagai berikut :

1. *Username*, berupa *text box* untuk mengisikan *username*.
2. *Password*, berupa *text box* untuk mengisikan *password*.
3. Tombol *Login*, untuk melakukan *login*.
4. Tombol *Batal*, untuk membatalkan *login*.

