

BAB III

PELAKSANAAN KERJA PROFSI

3.1 Bidang Kerja

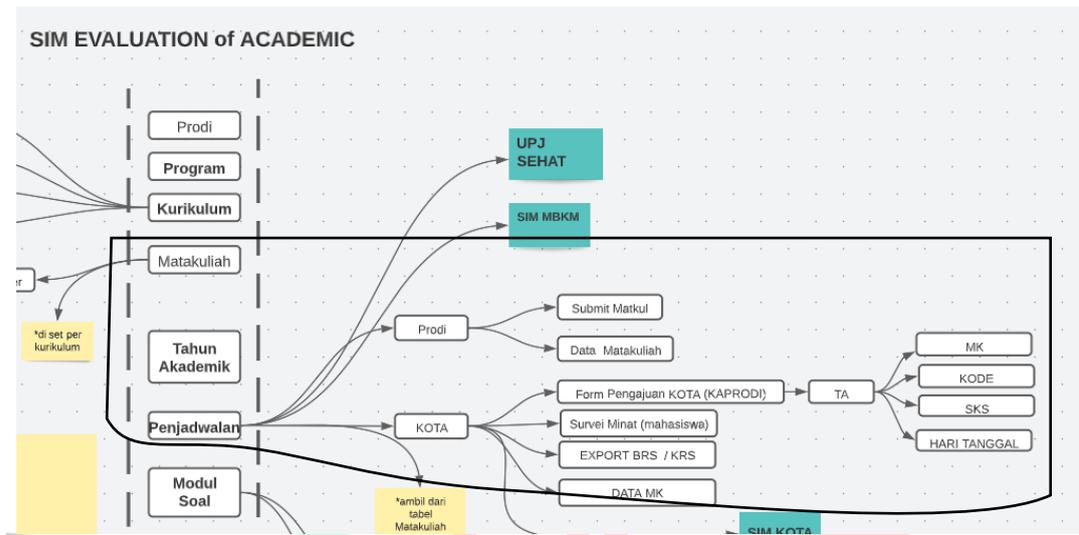
Pelaksanaan kerja profesi yang di laksanakan di Universitas Pembangunan Jaya, praktikan bekerja pada divisi *software development*. Pada divisi ini praktikan bertugas untuk membuat aplikasi sistem evaluasi akademik berbasis *outcome base evaluation* bagian modul penjadwalan. Aplikasi ini memiliki tiga menu utama yaitu menu akademik, kurikulum, penjadwalan. Dalam pembuatan menu ini praktikan membuat sub menu didalam akademik dan penjadwalan. Pada menu akademik praktikan membuat sub menu prodi, dosen, matakuliah, dan tahun akademik. Pada menu penjadwalan praktikan membuat sub menu peminatan dosen dan peminatan mahasiswa. Untuk menu lainnya dibuat oleh dua rekan praktikan lainnya. Tujuan dibuatnya aplikasi ini adalah sebagai sistem informasi yang berguna untuk melakukan evaluasi baik itu untuk matakuliah, dosen dan mahasiswa. Dengan adanya aplikasi ini dapat memudahkan mahasiswa dalam melihat informasi apa saja yang terdapat di Universitas Pembangunan Jaya.

3.2 Pelaksanaan Kerja

Dalam melaksanakan kerja profesi ini praktikan bekerja sejak tanggal 1 Juni – 31 Agustus di Universitas Pembangunan Jaya departement ICT. Dalam membuat aplikasi sistem evaluasi mahasiswa berbasis *outcome base evaluation* bagian modul penjadwalan, praktikan dibimbing oleh mentor mengenai tahapan atau proses dalam membuat aplikasi ini. Aplikasi ini dibuat sesuai dengan rancangan yang diberikan oleh mentor. Berikut adalah rancangan yang dibuat oleh mentor untuk mengembangkan aplikasi ini.

3.2.1 Perancangan

Pembuatan aplikasi sistem evaluasi mahasiswa dikerjakan oleh 3 orang yang masing-masing mempunyai tugas yang berbeda. Pembagian tugas dilakukan dengan berdiskusi dengan mentor. Praktikan mendapatkan tugas untuk membuat modul penjadwalan.



Gambar 3.2.1 1 Alur Aplikasi SIM OBE

Berikut adalah tugas-tugas yang dikerjakan praktikan selama mengikuti kerja profesi :

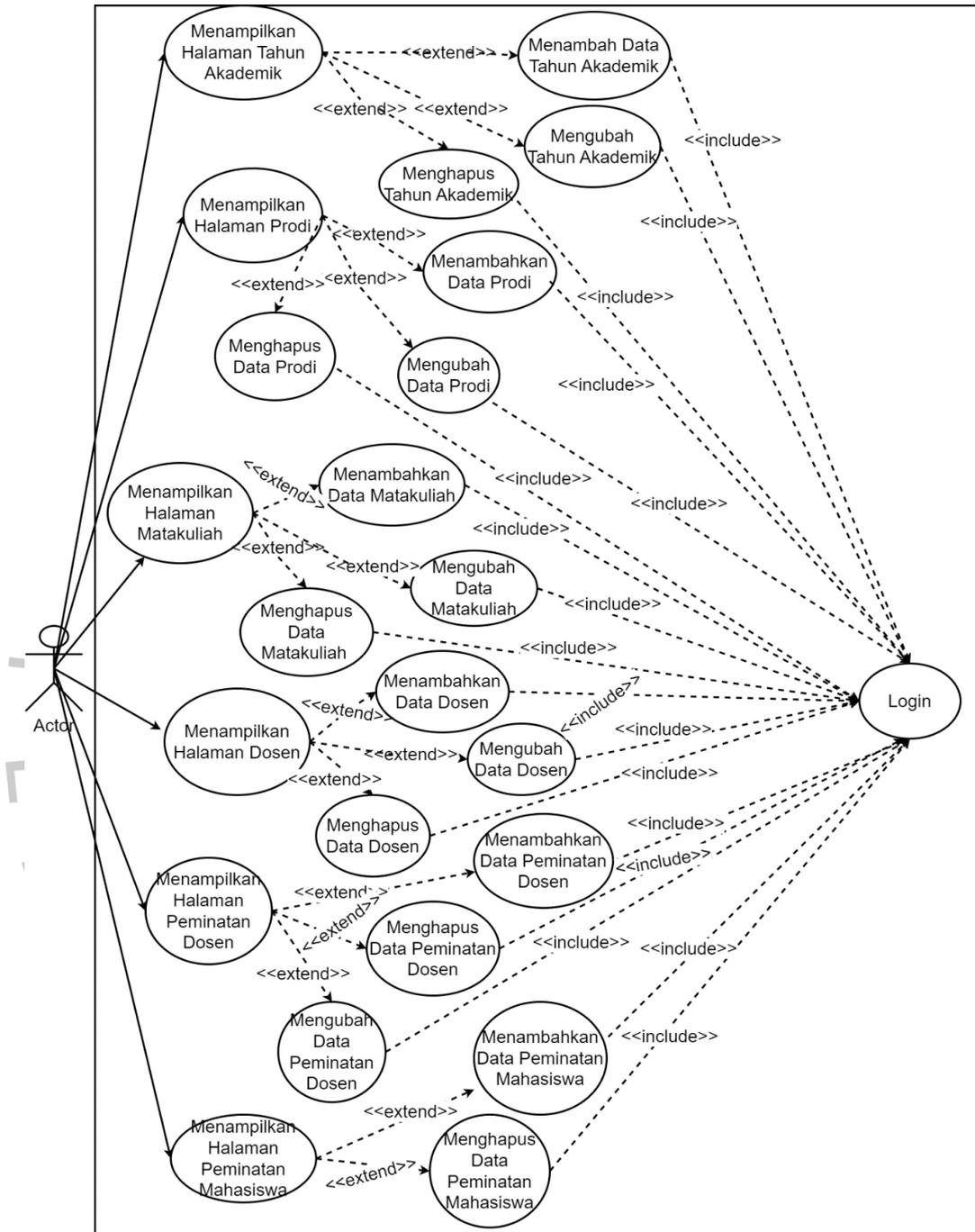
1. Pembuatan Aplikasi dan User requirement

Aplikasi sistem evaluasi mahasiswa dibuat menggunakan framework codeigniter 3. Aplikasi ini dibuat sebagai sistem informasi untuk mahasiswa dan dosen, yang isinya merupakan seputar akademik, kurikulum, dan penjadwalan matakuliah. Semua informasi mengenai prodi di Universitas Pembangunan Jaya akan muncul pada aplikasi ini.

Tabel 3.2. 1 User Requirement

No	Kebutuhan	Keterangan
1.	Menu Tahun Akademik	Admin pada menu tahun akademik dapat menambahkan, mengubah dan menghapus data yang berisi tahun id, nama, dan status.
2	Menu Dosen	Admin pada menu dosen dapat menambahkan, mengubah, dan menghapus data yang berisi nik, nidn, nama, email, prodi, dan foto.
3	Menu Prodi	Admin pada menu prodi dapat menambahkan, mengubah, dan menghapus data yang berisi kode, nama, nama en, keterangan, fakultas.
4	Menu Matakuliah	Admin pada menu matakuliah dapat menambahkan, mengubah, dan menghapus data yang berisi matakuliah, hari, jam mulai, jam selesai, ruang.
5	Menu Peminatan Dosen	Admin pada menu peminatan dosen dapat menambahkan, mengubah, dan menghapus data yang berisi dosen, tahun akademik, matakuliah, hari, jam mulai, jam selesai, ruang, kuota mahasiswa.
6	Menu Peminatan Mahasiswa	Admin pada menu peminatan mahasiswa dapat menambahkan data yang sudah divalidasi pada menu peminatan dosen. Kemudian admin dapat menghapus data menu peminatan mahasiswa.

2. Membuat Use Case Diagram



Gambar 3.2.1 2 Use Case Aplikasi SIM OBE

Diagram di atas merupakan *use case* untuk menggambarkan aplikasi yang dibuat oleh praktikan. Pada *use case* tersebut praktikan membuat menu akademik dan penjadwalan. Pada menu akademik praktikan membuat sub menu prodi, dosen, matakuliah, dan tahun akademik. Pada menu penjadwalan praktikan

membuat sub menu peminatan dosen dan peminatan mahasiswa. Aplikasi yang dibuat praktikan memiliki *Create, Read, Update, dan Delete*(CRUD). Untuk melakukan *crud* hanya yang login menggunakan akun admin.

Tabel 3.2.1 1 Skenario Use Case Menampilkan Tahun Akademik

Nama Use case	:	Menampilkan Halaman Tahun Akademik.
Aktor	:	Admin.
Deksripsi Use Case	:	Menampilkan data yang telah disimpan pada halaman tahun akademik.
<i>Precondition</i>	:	Di halaman tahun akademik pengguna dapat melihat data yang sudah tercantum di <i>database</i> .
Langkah-langkah	:	<ol style="list-style-type: none"> 1. Pengguna masuk ke halaman <i>dashboard</i>. 2. Pengguna memilih menu tahun akademik. 3. Halaman menu tahun akademik akan muncul data tahunid, nama, dan status.
<i>Postcondition</i>	:	Data akan muncul pada halaman tahun akademik

Tabel 3.2.1 2 Skenario Use Case Menambah Data Tahun Akademik

Nama Use case	:	Menambah Data Tahun Akademik
Aktor	:	Admin
Deksripsi Use Case	:	Menambahkan data ke <i>database</i> , kemudian ditampilkan ke halaman tahun akademik.
<i>Precondition</i>	:	Di halaman tahun akademik pengguna dapat menambah data ke <i>database</i> .
Langkah-langkah	:	<ol style="list-style-type: none"> 1. Pengguna masuk ke halaman tahun akademik. 2. Pengguna mengklik tombol tambah.

	<ol style="list-style-type: none"> 3. Pengguna mengisi data tahunid, nama, dan status pada <i>form</i> tambah tahun akademik. 4. Pengguna mengklik tombol tambah.
<i>Postcondition</i>	Data yang telah ditambahkan akan muncul pada halaman tahun akademik.

Tabel 3.2.1 3 Skenario Use Case Mengubah Data Tahun Akademik

Nama Use case	: Mengubah Data Tahun Akademik
Aktor	: Admin
Deksripsi Use Case	: Mengubah data pada halaman tahun akademik, kemudian data terbaru ditampilkan kembali.
<i>Precondition</i>	: Di halaman tahun akademik pengguna dapat mengubah data yang telah tercantum di <i>database</i> .
Langkah-langkah	<ol style="list-style-type: none"> 1. Pengguna masuk ke halaman tahun akademik. 2. Pengguna mengklik tombol <i>edit</i> pada kolom aksi. 3. Pengguna mengubah data tahunid, nama, dan status pada <i>form edit</i> tahun akademik. 4. Pengguna mengklik tombol <i>edit</i>.
<i>Postcondition</i>	Data yang diubah akan tersimpan ke <i>database</i> , kemudian akan ditampilkan ke halaman tahun akademik.

Tabel 3.2.1 4 Skenario Use Case Menghapus Data Tahun Akademik

Nama Use case	: Menghapus Data Tahun Akademik
Aktor	: Admin
Deksripsi Use Case	: Menghapus data pada halaman tahun akademik, kemudian data terbaru ditampilkan kembali.

<i>Precondition</i>	:	Di halaman tahun akademik pengguna dapat menghapus data yang telah tercantum di <i>database</i> .
Langkah-langkah	:	<ol style="list-style-type: none"> 1. Pengguna masuk ke halaman tahun akademik. 2. Pengguna mengklik tombol hapus pada kolom aksi. 3. Sistem akan menghapus data di <i>database</i>, kemudian data yang tidak terhapus akan muncul pada halaman tahun akademik.
<i>Postcondition</i>	:	Data yang di hapus akan hilang di <i>database</i> , kemudian sistem akan menampilkan data yang tidak terhapus di <i>database</i> ke halaman tahun akademik.

Tabel 3.2.1 5 Skenario Use Case Menampilkan Halaman Prodi

Nama Use case	:	Menampilkan Halaman Prodi.
Aktor	:	Admin.
Deksripsi Use Case	:	Menampilkan data yang telah disimpan pada halaman prodi.
<i>Precondition</i>	:	Di halaman prodi pengguna dapat melihat data yang telah tercantum di <i>database</i> .
Langkah-langkah	:	<ol style="list-style-type: none"> 1. Pengguna masuk ke halaman <i>dashboard</i>. 2. Pengguna memilih menu prodi. 3. Halaman menu prodi akan muncul data prodi, keterangan, dan status.
<i>Postcondition</i>	:	Data akan muncul pada halaman prodi.

Tabel 3.2.1 6 Skenario Use Case Menambah Data Prodi

Nama Use case	:	Menambah Data Prodi
Aktor	:	Admin

Deksripsi Use Case	:	Menambahkan data ke <i>database</i> , kemudian ditampilkan ke halaman prodi.
<i>Precondition</i>	:	Di halaman prodi pengguna dapat menambah data ke <i>database</i> .
Langkah-langkah	:	<ol style="list-style-type: none"> 1. Pengguna masuk ke halaman prodi. 2. Pengguna mengklik tombol tambah. 3. Pengguna mengisi data kode, nama, namaEn, keterangan, fakultas pada <i>form</i> tambah prodi. 4. Pengguna mengklik tombol tambah.
<i>Postcondition</i>	:	Data yang telah ditambahkan akan muncul pada halaman prodi.

Tabel 3.2.1 7 Skenario Use Case Mengubah Data Prodi

Nama Use case	:	Mengubah Data Prodi
Aktor	:	Admin
Deksripsi Use Case	:	Mengubah data pada halaman prodi, kemudian data terbaru ditampilkan kembali.
<i>Precondition</i>	:	Di halaman prodi pengguna dapat mengubah data yang telah tercantum di <i>database</i> .
Langkah-langkah	:	<ol style="list-style-type: none"> 1. Pengguna masuk ke halaman prodi. 2. Pengguna mengklik tombol <i>edit</i> pada kolom aksi. 3. Pengguna mengubah data kode, nama, namaEn, keterangan, fakultas pada <i>form edit</i> prodi. 4. Pengguna mengklik tombol <i>edit</i>.
<i>Postcondition</i>	:	Data yang diubah akan tersimpan ke <i>database</i> , kemudian akan ditampilkan ke halaman prodi.

Tabel 3.2.1 8 Skenario Use Case Menghapus Data Prodi

Nama Use case	: Menghapus Data Prodi
Aktor	: Admin
Deksripsi Use Case	: Menghapus data pada halaman prodi, kemudian data terbaru ditampilkan kembali.
<i>Precondition</i>	: Di halaman prodi pengguna dapat menghapus data yang telah tercantum di <i>database</i> .
Langkah-langkah	: <ol style="list-style-type: none"> 1. Pengguna masuk ke halaman prodi. 2. Pengguna mengklik tombol hapus pada kolom aksi. 3. Sistem akan menghapus data di <i>database</i>, kemudian data yang tidak terhapus akan muncul pada halaman prodi.
<i>Postcondition</i>	: Data yang di hapus akan hilang di <i>database</i> , kemudian sistem akan menampilkan data yang tidak terhapus di <i>database</i> ke halaman prodi.

Tabel 3.2.1 9 Skenario Use Case Menampilkan Halaman Matakuliah

Nama Use case	: Menampilkan Halaman Matakuliah.
Aktor	: Admin.
Deksripsi Use Case	: Menampilkan data yang telah disimpan pada halaman matakuliah.
<i>Precondition</i>	: Di halaman matakuliah pengguna dapat melihat data yang telah tercantum di <i>database</i> .
Langkah-langkah	: <ol style="list-style-type: none"> 1. Pengguna masuk ke halaman <i>dashboard</i>. 2. Pengguna memilih menu matakuliah. 3. Halaman menu matakuliah akan muncul data nama matakuliah, hari, ruang, jam mulai, jam selesai.

<i>Postcondition</i>	Data akan muncul pada halaman matakuliah.
----------------------	---

Tabel 3.2.1 10 Skenario Use Case Menambah Data Matakuliah

Nama Use case	: Menambah Data Matakuliah
Aktor	: Admin
Deksripsi Use Case	: Menambahkan data ke <i>database</i> , kemudian ditampilkan ke halaman matakuliah.
<i>Precondition</i>	: Di halaman matakuliah pengguna dapat menambah data ke <i>database</i> .
Langkah-langkah	: <ol style="list-style-type: none"> 1. Pengguna masuk ke halaman matakuliah. 2. Pengguna mengklik tombol tambah. 3. Pengguna mengisi data nama matakuliah, hari, ruang, jam mulai, jam selesai pada <i>form</i> tambah matakuliah. 4. Pengguna mengklik tombol tambah.
<i>Postcondition</i>	Data yang telah ditambahkan akan muncul pada halaman matakuliah.

Tabel 3.2.1 11 Skenario Use Case Mengubah Data Matakuliah

Nama Use case	: Mengubah Data Matakuliah
Aktor	: Admin
Deksripsi Use Case	: Mengubah data pada halaman matakuliah, kemudian data terbaru ditampilkan kembali.
<i>Precondition</i>	: Di halaman matakuliah pengguna dapat mengubah data yang telah tercantum di <i>database</i> .
Langkah-langkah	: <ol style="list-style-type: none"> 1. Pengguna masuk ke halaman matakuliah. 2. Pengguna mengklik tombol <i>edit</i> pada kolom aksi.

	<ol style="list-style-type: none"> 3. Pengguna mengubah data nama matakuliah, hari, ruang, jam mulai, jam selesai pada <i>form edit</i> matakuliah. 4. Pengguna mengklik tombol <i>edit</i>.
<i>Postcondition</i>	Data yang diubah akan tersimpan ke <i>database</i> , kemudian akan ditampilkan ke halaman matakuliah.

Tabel 3.2.1 12 Skenario Use Case Menghapus Data Matakuliah

Nama Use case	: Menghapus Data Matakuliah
Aktor	: Admin
Deksripsi Use Case	: Menghapus data pada halaman matakuliah, kemudian data terbaru ditampilkan kembali.
<i>Precondition</i>	: Di halaman matakuliah pengguna dapat menghapus data yang telah tercantum di <i>database</i> .
Langkah-langkah	<ol style="list-style-type: none"> 1. Pengguna masuk ke halaman matakuliah. 2. Pengguna mengklik tombol hapus pada kolom aksi. 3. Sistem akan menghapus data di <i>database</i>, kemudian data yang tidak terhapus akan muncul pada halaman matakuliah.
<i>Postcondition</i>	Data yang di hapus akan hilang di <i>database</i> , kemudian sistem akan menampilkan data yang tidak terhapus di <i>database</i> ke halaman matakuliah.

Tabel 3.2.1 13 Skenario Use Case Menampilkan Halaman Dosen

Nama Use case	: Menampilkan Halaman Dosen.
Aktor	: Admin.
Deksripsi Use Case	: Menampilkan data yang telah disimpan pada halaman dosen.
<i>Precondition</i>	: Di halaman dosen pengguna dapat melihat data yang telah

		tercantum di <i>database</i> .
Langkah-langkah	:	<ol style="list-style-type: none"> 1. Pengguna masuk ke halaman <i>dashboard</i>. 2. Pengguna memilih menu dosen. 3. Halaman menu dosen akan muncul data nik, nidn, nama, prodi, foto, email, dan status.
<i>Postcondition</i>		Data akan muncul pada halaman dosen.

Tabel 3.2.1 14 Skenario Use Case Menambahkan Dosen

Nama Use case	:	Menambah Data Dosen
Aktor	:	Admin
Deksripsi Use Case	:	Menambahkan data ke <i>database</i> , kemudian ditampilkan ke halaman dosen.
<i>Precondition</i>	:	Di halaman dosen pengguna dapat menambah data ke <i>database</i> .
Langkah-langkah	:	<ol style="list-style-type: none"> 1. Pengguna masuk ke halaman dosen. 2. Pengguna mengklik tombol tambah. 3. Pengguna mengisi data nik, nidn, nama, prodi, foto, email, dan status pada <i>form</i> tambah dosen. 4. Pengguna mengklik tombol tambah.
<i>Postcondition</i>		Data yang telah ditambahkan akan muncul pada halaman dosen.

Tabel 3.2.1 15 Skenario Use Case Mengubah Data Dosen

Nama Use case	:	Mengubah Data Dosen
Aktor	:	Admin
Deksripsi Use Case	:	Mengubah data pada halaman dosen, kemudian data terbaru ditampilkan kembali.

<i>Precondition</i>	:	Di halaman dosen pengguna dapat mengubah data yang telah tercantum di <i>database</i> .
Langkah-langkah	:	<ol style="list-style-type: none"> 1. Pengguna masuk ke halaman dosen. 2. Pengguna mengklik tombol <i>edit</i> pada kolom aksi. 3. Pengguna mengubah data nik, nidn, nama, prodi, foto, email, dan status pada <i>form edit</i> dosen. 4. Pengguna mengklik tombol <i>edit</i>.
<i>Postcondition</i>	:	Data yang diubah akan tersimpan ke <i>database</i> , kemudian akan ditampilkan ke halaman dosen.

Tabel 3.2.1 16 Skenario Use Case Menghapus Data Dosen

Nama Use case	:	Menghapus Data Dosen
Aktor	:	Admin
Deksripsi Use Case	:	Menghapus data pada halaman dosen, kemudian data terbaru ditampilkan kembali.
<i>Precondition</i>	:	Di halaman dosen pengguna dapat menghapus data yang telah tercantum di <i>database</i> .
Langkah-langkah	:	<ol style="list-style-type: none"> 1. Pengguna masuk ke halaman dosen. 2. Pengguna mengklik tombol hapus pada kolom aksi. 3. Sistem akan menghapus data di <i>database</i>, kemudian data yang tidak terhapus akan muncul pada halaman dosen.
<i>Postcondition</i>	:	Data yang di hapus akan hilang di <i>database</i> , kemudian sistem akan menampilkan data yang tidak terhapus di <i>database</i> ke halaman dosen.

Tabel 3.2.1 17 Skenario Use Case Menampilkan Halaman Peminatan Dosen

Nama Use case	:	Menampilkan Halaman Peminatan Dosen.
Aktor	:	Admin.
Deksripsi Use Case	:	Menampilkan data yang telah disimpan pada halaman peminatan dosen.
<i>Precondition</i>	:	Di halaman peminatan dosen pengguna dapat melihat data yang telah tercantum di <i>database</i> .
Langkah-langkah	:	<ol style="list-style-type: none"> 1. Pengguna masuk ke halaman <i>dashboard</i>. 2. Pengguna memilih menu peminatan dosen. 3. Halaman menu peminatan dosen akan muncul data dosen, tahun akademik, matakuliah, hari, jam mulai-selesai, ruang.
<i>Postcondition</i>	:	Data akan muncul pada halaman peminatan dosen.

Tabel 3.2.1 18 Skenario Use Case Menambah Data Peminatan Dosen

Nama Use case	:	Menambah Data Peminatan Dosen
Aktor	:	Admin
Deksripsi Use Case	:	Menambahkan data ke <i>database</i> , kemudian ditampilkan ke halaman peminatan dosen.
<i>Precondition</i>	:	Di halaman peminatan dosen pengguna dapat menambah data ke <i>database</i> .
Langkah-langkah	:	<ol style="list-style-type: none"> 1. Pengguna masuk ke halaman peminatan dosen. 2. Pengguna mengklik tombol tambah. 3. Pengguna mengisi data dosen, tahun akademik, matakuliah, hari, jam mulai, jam selesai, ruang pada <i>form</i> tambah peminatan dosen. 4. Pengguna mengklik tombol tambah.

<i>Postcondition</i>	Data yang telah ditambahkan akan muncul pada halaman peminatan dosen.
----------------------	---

Tabel 3.2.1 19 Skenario Use Case Mengubah Data Peminatan Dosen

Nama Use case	: Mengubah Data Peminatan Dosen
Aktor	: Admin
Deksripsi Use Case	: Mengubah data pada halaman peminatan dosen, kemudian data terbaru ditampilkan kembali.
<i>Precondition</i>	: Di halaman peminatan dosen pengguna dapat mengubah data yang telah tercantum di <i>database</i> .
Langkah-langkah	: <ol style="list-style-type: none"> 1. Pengguna masuk ke halaman peminatan dosen. 2. Pengguna mengklik tombol <i>edit</i> pada kolom aksi. 3. Pengguna mengubah data dosen, tahun akademik, matakuliah, hari, jam mulai, jam selesai, ruang pada <i>form edit</i> peminatan dosen. 4. Pengguna mengklik tombol <i>edit</i>.
<i>Postcondition</i>	Data yang diubah akan tersimpan ke <i>database</i> , kemudian akan ditampilkan ke halaman peminatan dosen.

Tabel 3.2.1 20 Skenario Use Case Menghapus Data Peminatan Dosen

Nama Use case	: Menghapus Data Peminatan Dosen
Aktor	: Admin
Deksripsi Use Case	: Menghapus data pada halaman peminatan dosen, kemudian data terbaru ditampilkan kembali.
<i>Precondition</i>	: Di halaman peminatan dosen pengguna dapat menghapus data yang telah tercantum di <i>database</i> .

Langkah-langkah	:	<ol style="list-style-type: none"> 1. Pengguna masuk ke halaman peminatan dosen. 2. Pengguna mengklik tombol hapus pada kolom aksi. 3. Sistem akan menghapus data di <i>database</i>, kemudian data yang tidak terhapus akan muncul pada halaman peminatan dosen.
<i>Postcondition</i>	:	Data yang di hapus akan hilang di <i>database</i> , kemudian sistem akan menampilkan informasi yang tidak terhapus di <i>database</i> ke halaman peminatan dosen.

Tabel 3.2.1 21 Skenario Use Case Menampilkan Halaman Peminatan Mahasiswa

Nama Use case	:	Menampilkan Halaman Peminatan Mahasiswa.
Aktor	:	Admin.
Deksripsi Use Case	:	Menampilkan data yang telah disimpan pada halaman peminatan mahasiswa.
<i>Precondition</i>	:	Di halaman peminatan mahasiswa pengguna dapat melihat data yang telah tercantum di <i>database</i> .
Langkah-langkah	:	<ol style="list-style-type: none"> 1. Pengguna masuk ke halaman <i>dashboard</i>. 2. Pengguna memilih menu peminatan mahasiswa. 3. Halaman menu peminatan mahasiswa akan muncul data matakuliah, jam, dosen.
<i>Postcondition</i>	:	Data akan muncul pada halaman peminatan mahasiswa.

Tabel 3.2.1 22 Skenario Use Case Menambah Data Peminatan Mahasiswa

Nama Use case	:	Menambah Data Peminatan Mahasiswa
Aktor	:	Admin
Deksripsi Use Case	:	Menambahkan data ke <i>database</i> , kemudian ditampilkan ke halaman peminatan mahasiswa.

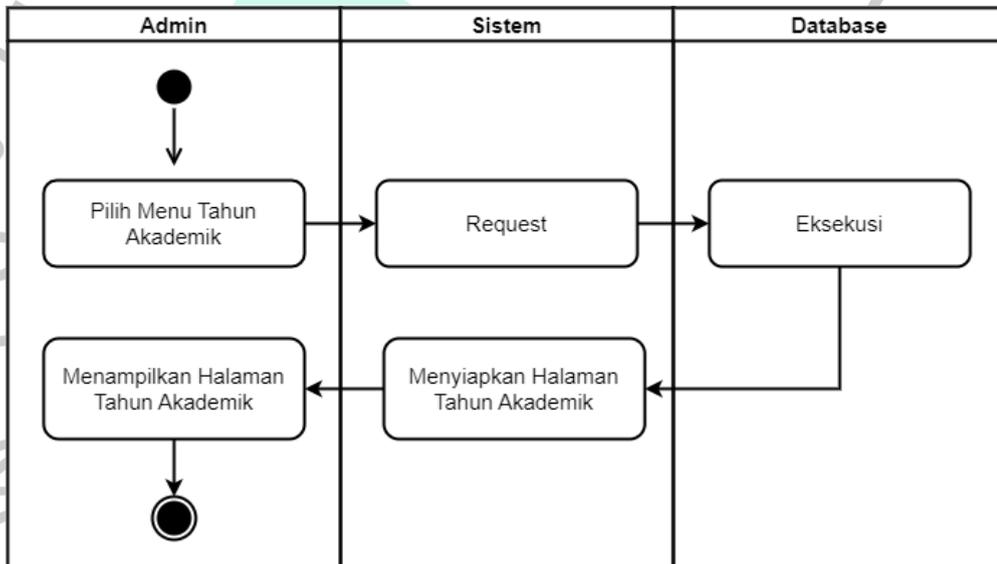
<i>Precondition</i>	:	Di halaman peminatan mahasiswa pengguna dapat menambah data ke <i>database</i> .
Langkah-langkah	:	<ol style="list-style-type: none"> 1. Pengguna masuk ke halaman peminatan mahasiswa. 2. Pengguna mengklik tombol tambah. 3. Pengguna mengisi data matakuliah, jam, dosen pada <i>form</i> tambah peminatan mahasiswa. 4. Pengguna mengklik tombol tambah.
<i>Postcondition</i>	:	Data yang telah ditambahkan akan muncul pada halaman peminatan mahasiswa.

Tabel 3.2.1 23 Skenario Use Case Menghapus Data Peminatan Mahasiswa

Nama Use case	:	Menghapus Data Peminatan Mahasiswa
Aktor	:	Admin
Deksripsi Use Case	:	Menghapus data pada halaman peminatan mahasiswa, kemudian data terbaru ditampilkan kembali.
<i>Precondition</i>	:	Di halaman peminatan mahasiswa pengguna dapat menghapus data yang telah tercantum di <i>database</i> .
Langkah-langkah	:	<ol style="list-style-type: none"> 1. Pengguna masuk ke halaman peminatan mahasiswa. 2. Pengguna mengklik tombol hapus pada kolom aksi. 3. Sistem akan menghapus data di <i>database</i>, kemudian data yang tidak terhapus akan muncul pada halaman peminatan mahasiswa.
<i>Postcondition</i>	:	Data yang di hapus akan hilang di <i>database</i> , kemudian sistem akan menampilkan informasi yang tidak terhapus di <i>database</i> ke halaman peminatan mahasiswa.

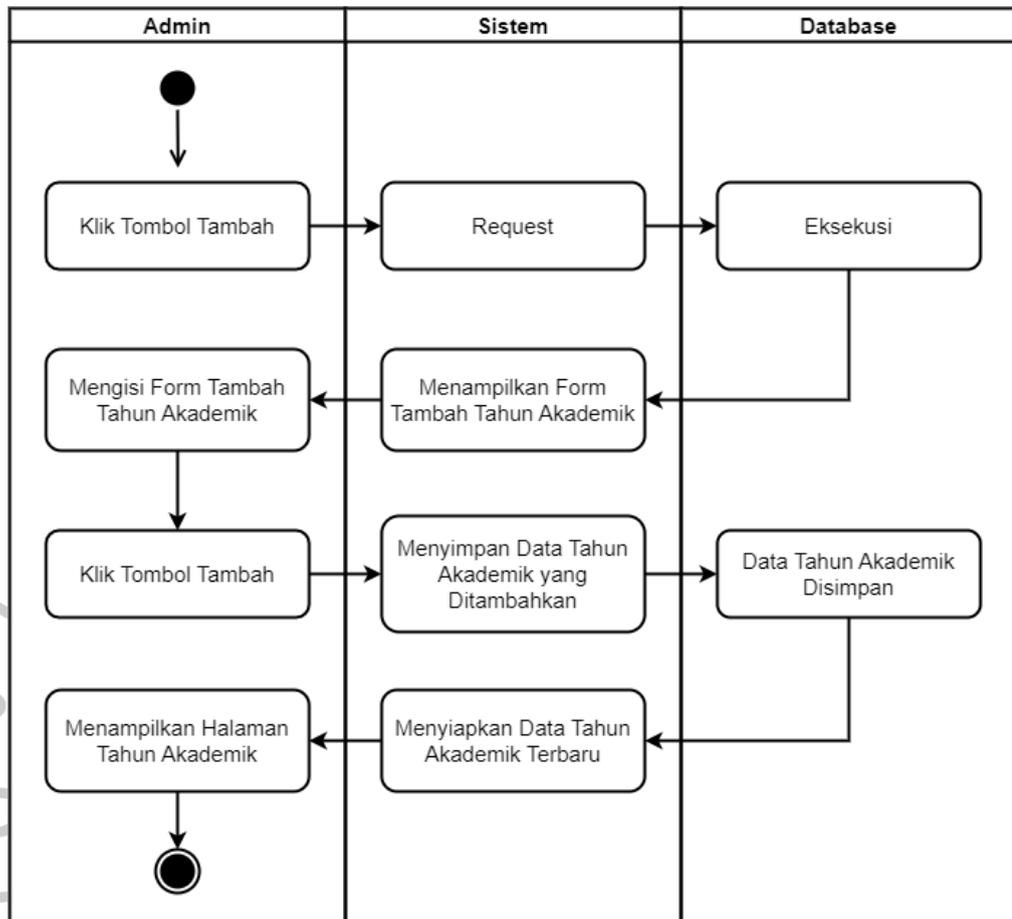
3. Membuat *Activity* Diagram

Activity diagram adalah gambaran sebuah proses yang terdapat pada sistem. *Activity* diagram merupakan lanjutan dari pengembangan use case yang memiliki alur aktivitas. Diagram ini hanya untuk menggambarkan alur kerja atau aktivitas dari sistem saja. Tujuan dari *activity* diagram adalah untuk menjelaskan urutan aktivitas dari sebuah proses dan mengetahui aktivitas pengguna berdasarkan use case diagram. Berikut adalah diagram *activity* pada perancangan aplikasi SIM OBE.



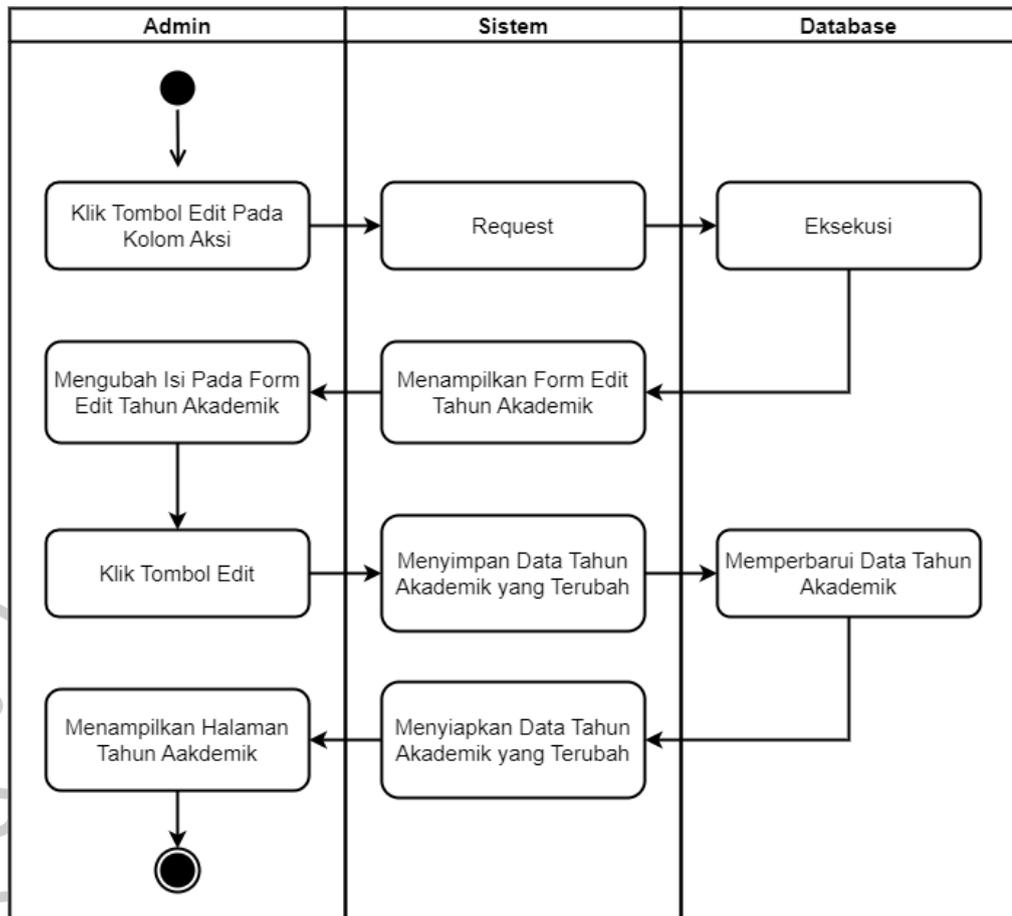
Gambar 3.2.1 3 Activity Diagram Menampilkan Tahun Akademik

Gambar diatas merupakan *activity* diagram untuk menu tahun akademik. *Activity* ini dimulai dari memilih menu tahun akademik, kemudian mengirim *request* ke sistem untuk menampilkan halaman tahun akademik. Sistem meneruskan ke *database* untuk dieksekusi. Setelah dieksekusi oleh *database*, sistem menyiapkan halaman tahun akademik yang datanya berasal dari *database*. Kemudian sistem akan menampilkan halaman tahun akademik.



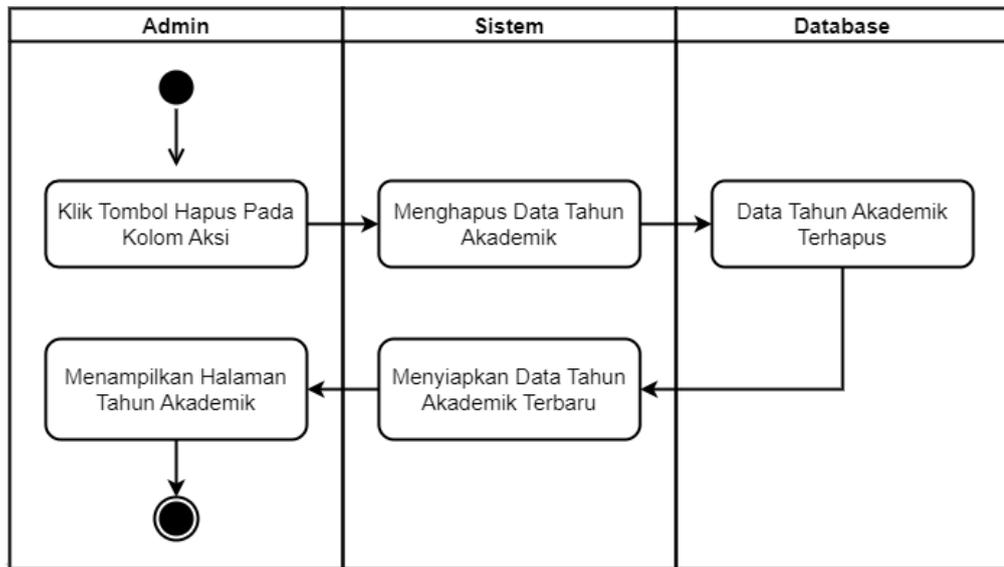
Gambar 3.2.1 4 Activity Diagram Menambah Data Tahun Akademik

Gambar diatas merupakan *activity* diagram menambah data tahun akademik. *Activity* ini dimulai dari admin mengklik tombol tambah pada halaman tahun akademik. Kemudian admin mengirim *request* ke sistem yang dilanjutkan ke *database* untuk dieksekusi. Setelah dieksekusi, sistem akan menampilkan *form* tambah tahun akademik. Admin mengisi *form* tambah tahun akademik kemudian mengklik tombol tambah. Setelah data tersimpan di *database*, kemudian sistem akan menampilkan halaman tahun akademik dengan data terbaru.



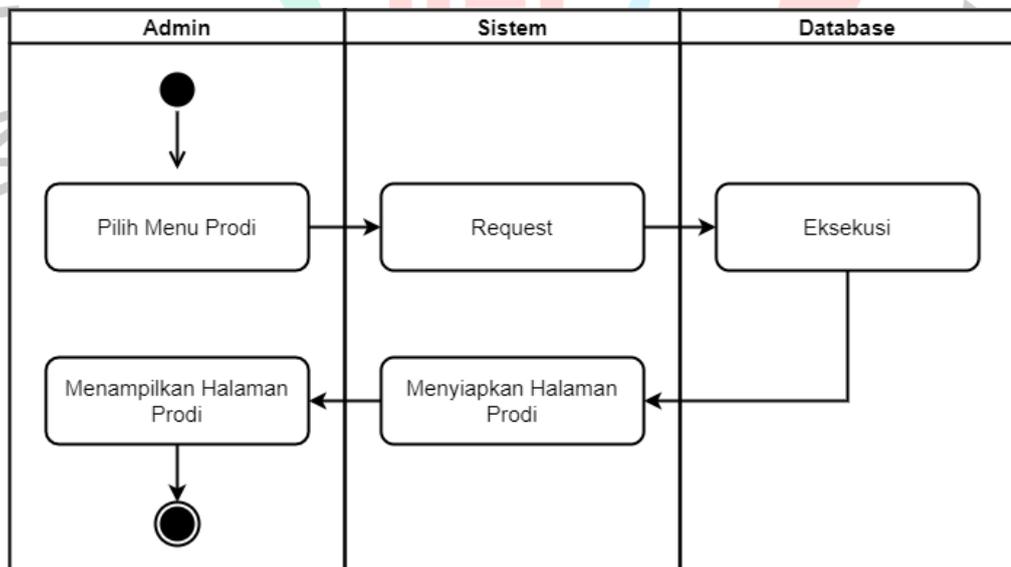
Gambar 3.2.1 5 Activity Diagram Mengubah Data Tahun Akademik

Gambar diatas merupakan *activity* diagram mengubah data tahun akademik. *Activity* ini dimulai dari admin mengklik tombol *edit* pada kolom aksi. Kemudian admin mengirim *request* ke sistem yang dilanjutkan ke *database* untuk dieksekusi. Setelah dieksekusi, sistem akan menampilkan *form edit* tahun akademik. Admin mengubah isi data pada *form edit* tahun akademik kemudian mengklik tombol *edit*. Setelah memperbarui data tahun akademik di *database*, kemudian sistem akan menampilkan halaman tahun akademik dengan data yang sudah diubah.



Gambar 3.2.1 6 Activity Diagram Menghapus Data Tahun Akademik

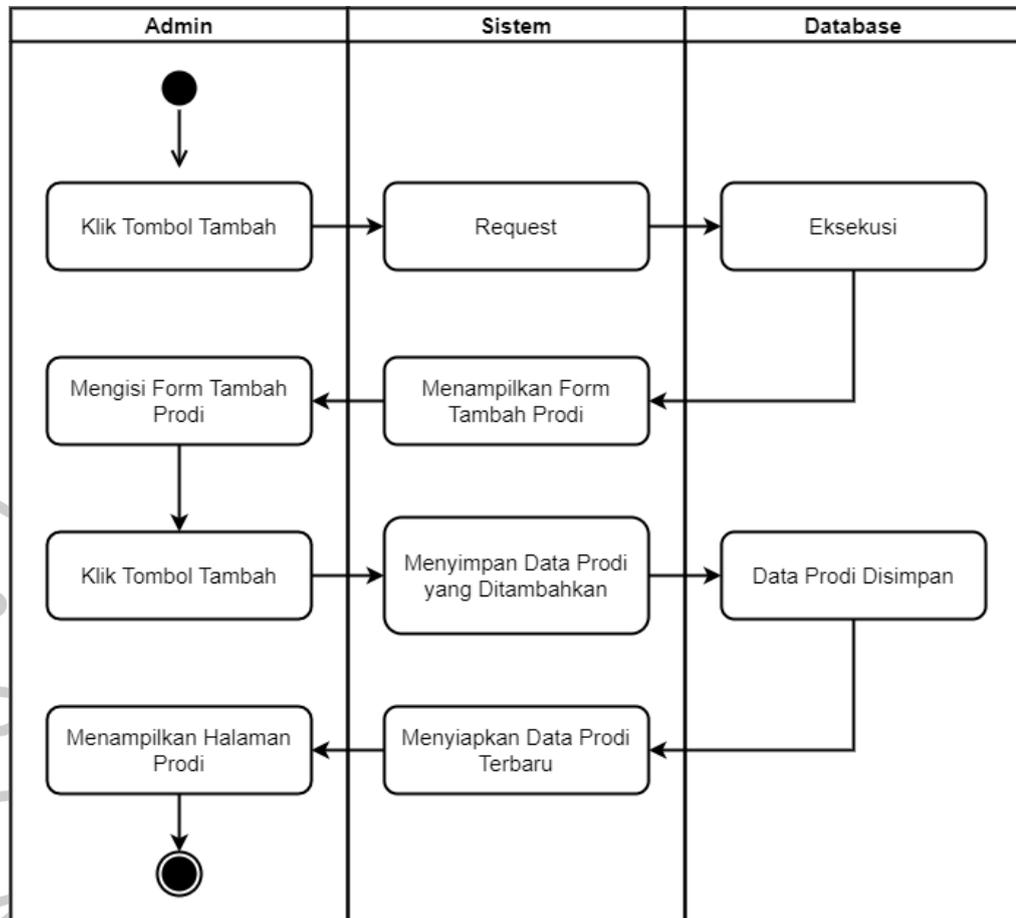
Gambar diatas merupakan *activity* diagram menghapus data tahun akademik. *Activity* ini dimulai dari admin mengklik tombol hapus pada kolom aksi. Kemudian sistem akan menghapus data tahun akademik, dan melanjutkan ke *database* untuk menghapus data. Setelah data terhapus, sistem akan menampilkan halaman tahun akademik dengan data terbaru.



Gambar 3.2.1 7 Activity Diagram Menampilkan Prodi

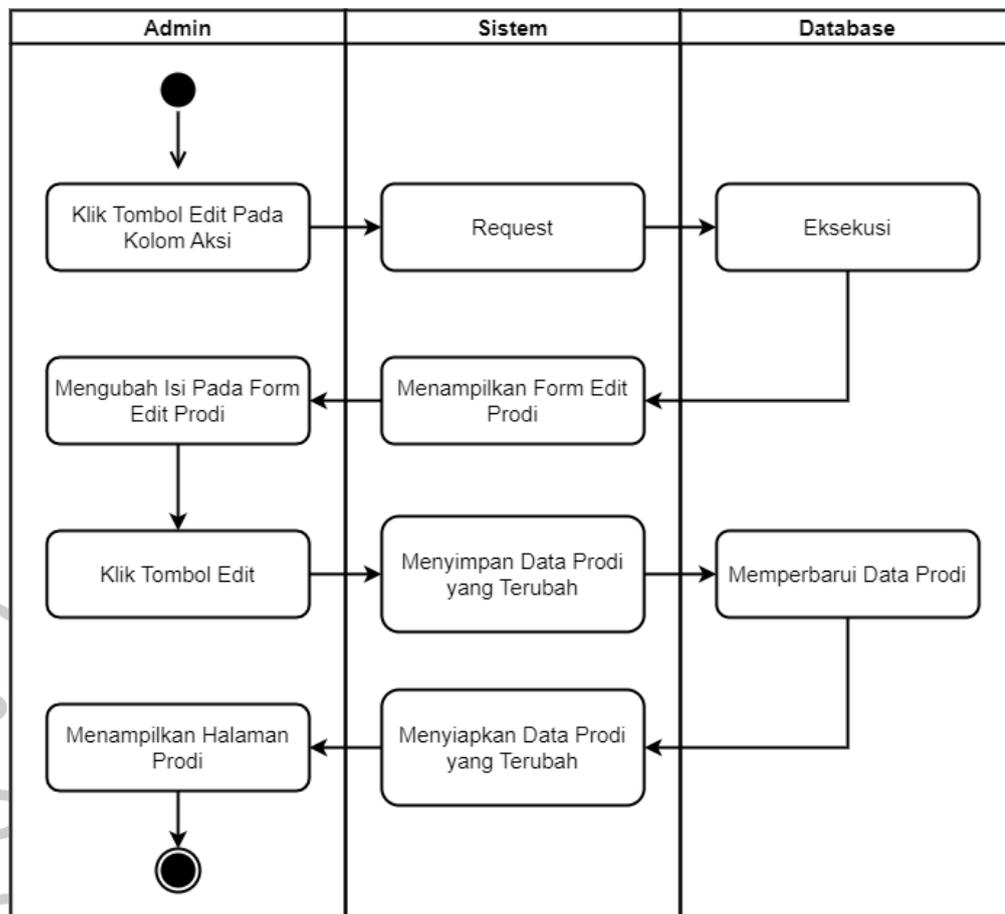
Gambar diatas merupakan *activity* diagram untuk menu prodi. *Activity* ini dimulai dari memilih menu prodi, kemudian mengirim *request* ke sistem untuk menampilkan halaman tahun akademik. Sistem meneruskan ke *database* untuk dieksekusi. Setelah dieksekusi oleh *database*, sistem menyiapkan

halaman prodi yang datanya berasal dari *database*. Kemudian sistem akan menampilkan halaman prodi.



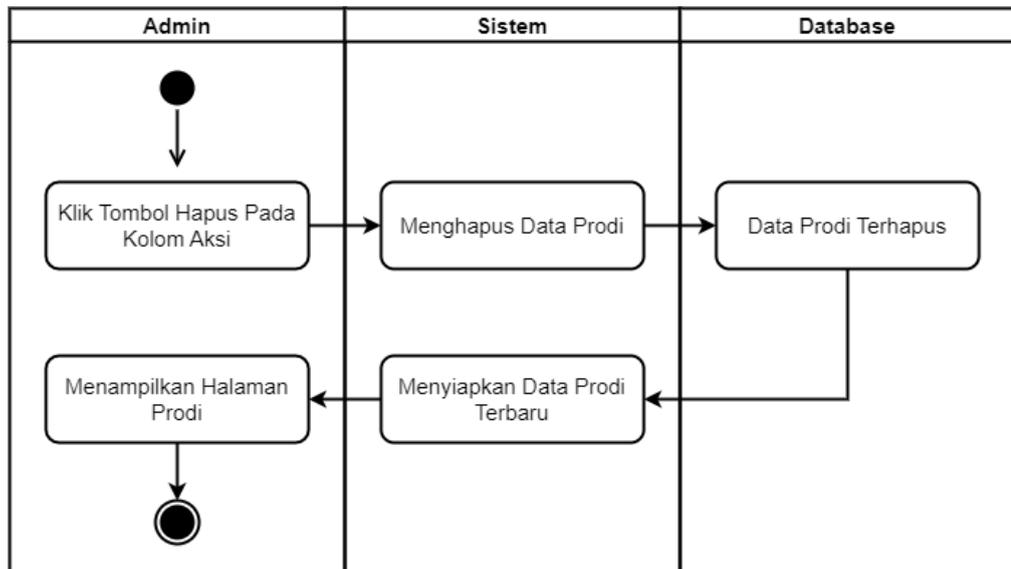
Gambar 3.2.1 8 Activity Diagram Menambah Data Prodi

Gambar diatas merupakan *activity* diagram menambah data prodi. *Activity* ini dimulai dari admin mengklik tombol tambah pada halaman prodi. Kemudian admin mengirim *request* ke sistem yang dilanjutkan ke *database* untuk dieksekusi. Setelah dieksekusi, sistem akan menampilkan *form* tambah prodi. Admin mengisi *form* prodi kemudian mengklik tombol tambah. Setelah data tersimpan di *database*, kemudian sistem akan menampilkan halaman prodi dengan data terbaru.



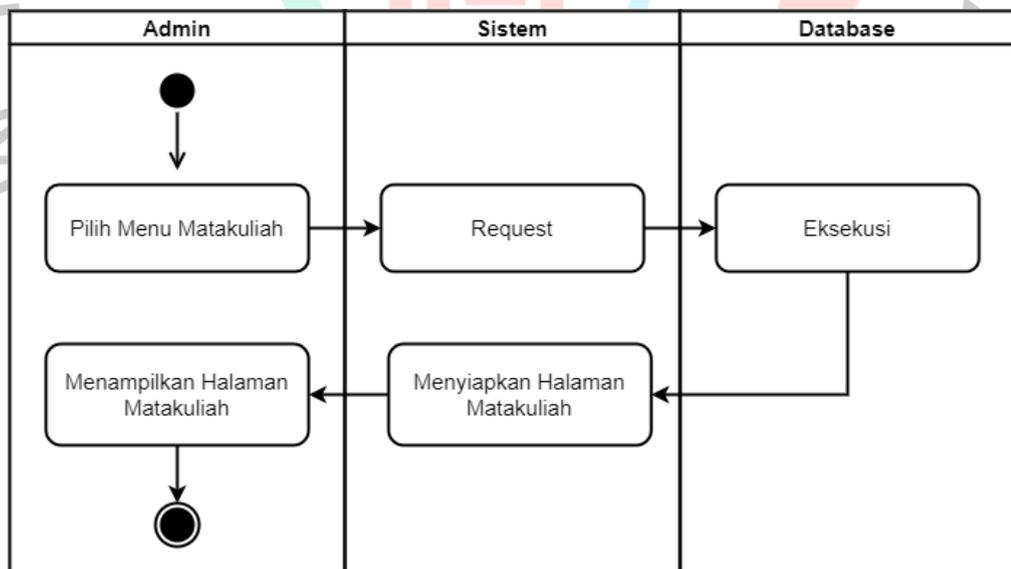
Gambar 3.2.1 9 Activity Diagram Mengubah Data Prodi

Gambar diatas merupakan *activity* diagram mengubah data prodi. *Activity* ini dimulai dari admin mengklik tombol *edit* pada kolom aksi. Kemudian admin mengirim *request* ke sistem yang dilanjutkan ke *database* untuk dieksekusi. Setelah dieksekusi, sistem akan menampilkan *form edit* prodi. Admin mengubah isi data pada *form edit* prodi kemudian mengklik tombol *edit*. Setelah memperbarui data prodi di *database*, kemudian sistem akan menampilkan halaman prodi dengan data yang sudah diubah.



Gambar 3.2.1 10 Activity Diagram Menghapus Data Prodi

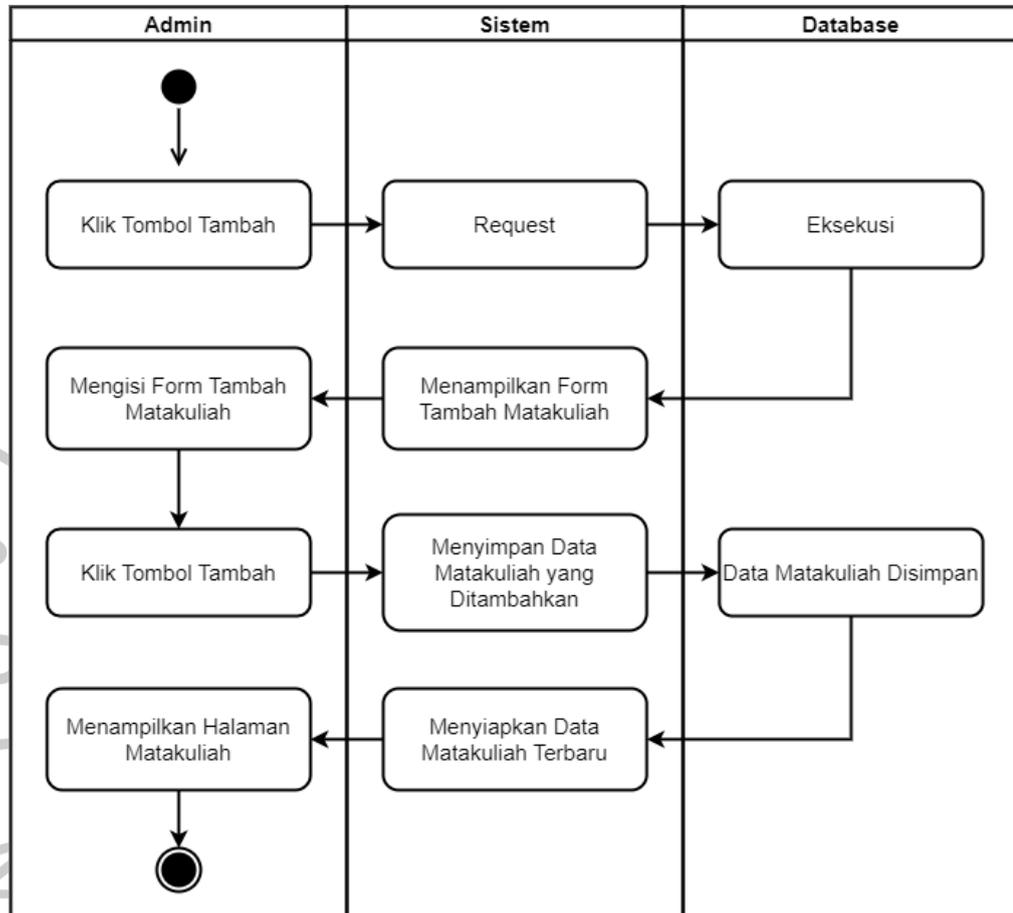
Gambar diatas merupakan *activity* diagram menghapus data prodi. *Activity* ini dimulai dari admin mengklik tombol hapus pada kolom aksi. Kemudian sistem akan menghapus data prodi, dan melanjutkan ke *database* untuk menghapus data. Setelah data terhapus, sistem akan menampilkan halaman prodi dengan data terbaru.



Gambar 3.2.1 11 Activity Diagram Menampilkan Matakuliah

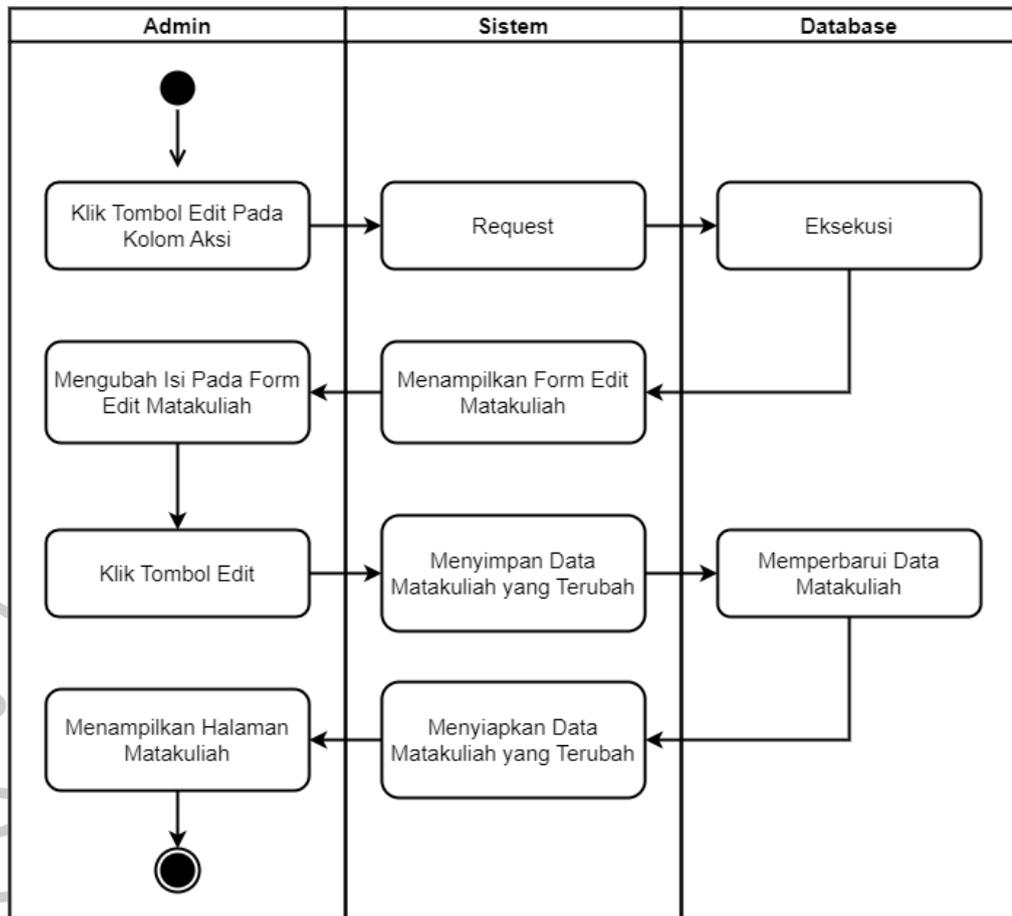
Gambar diatas merupakan *activity* diagram untuk menu matakuliah. *Activity* ini dimulai dari memilih menu matakuliah, kemudian mengirim *request* ke sistem untuk menampilkan halaman matakuliah. Sistem meneruskan ke

database untuk dieksekusi. Setelah dieksekusi oleh *database*, sistem menyiapkan halaman matakuliah yang datanya berasal dari *database*. Kemudian sistem akan menampilkan halaman matakuliah.



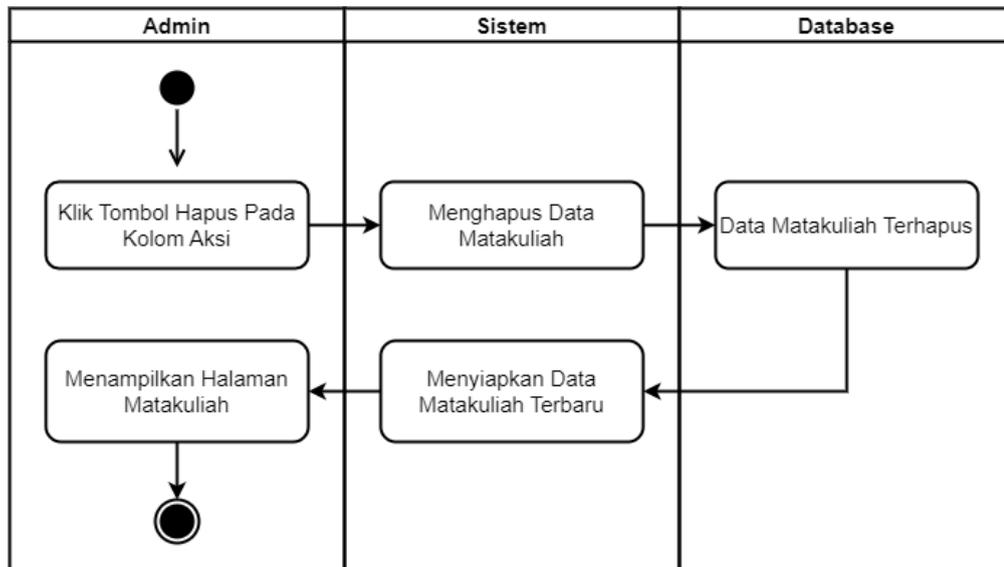
Gambar 3.2.1 12 Activity Diagram Menambah Data Matakuliah

Gambar diatas merupakan *activity* diagram menambah data matakuliah. *Activity* ini dimulai dari admin mengklik tombol tambah pada halaman matakuliah. Kemudian admin mengirim *request* ke sistem yang dilanjutkan ke *database* untuk dieksekusi. Setelah dieksekusi, sistem akan menampilkan *form* tambah matakuliah. Admin mengisi *form* matakuliah kemudian mengklik tombol tambah. Setelah data tersimpan di *database*, kemudian sistem akan menampilkan halaman matakuliah dengan data terbaru.



Gambar 3.2.1 13 Activity Diagram Mengubah Data Matakuliah

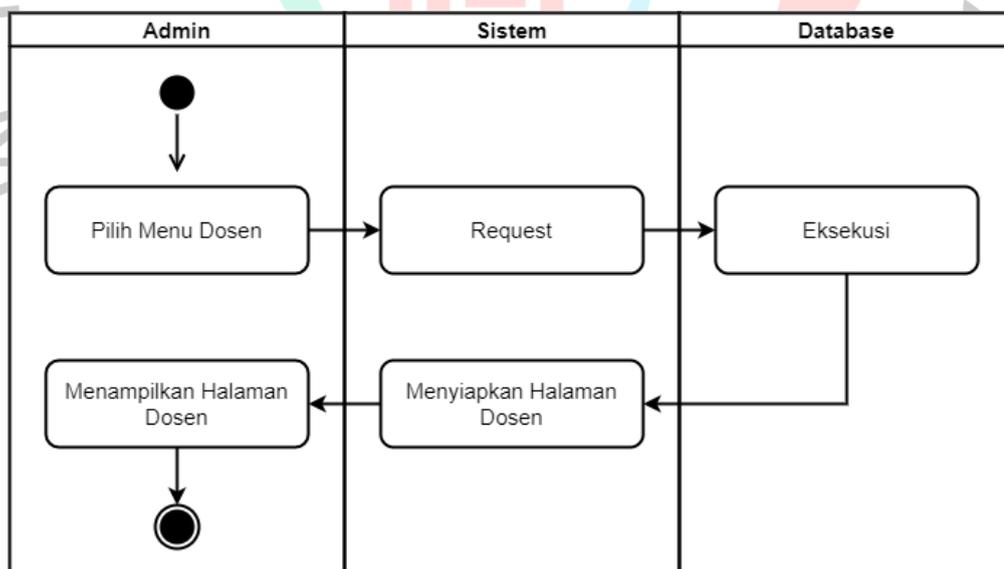
Gambar diatas merupakan *activity* diagram mengubah data matakuliah. *Activity* ini dimulai dari admin mengklik tombol *edit* pada kolom aksi. Kemudian admin mengirim *request* ke sistem yang dilanjutkan ke *database* untuk dieksekusi. Setelah dieksekusi, sistem akan menampilkan *form edit* matakuliah. Admin mengubah isi data pada *form edit* matakuliah kemudian mengklik tombol *edit*. Setelah memperbarui data matakuliah di *database*, kemudian sistem akan menampilkan halaman matakuliah dengan data yang sudah diubah.



Gambar 3.2.1 14 Activity Diagram Menghapus Data Matakuliah

Gambar diatas merupakan *activity* diagram menghapus data matakuliah.

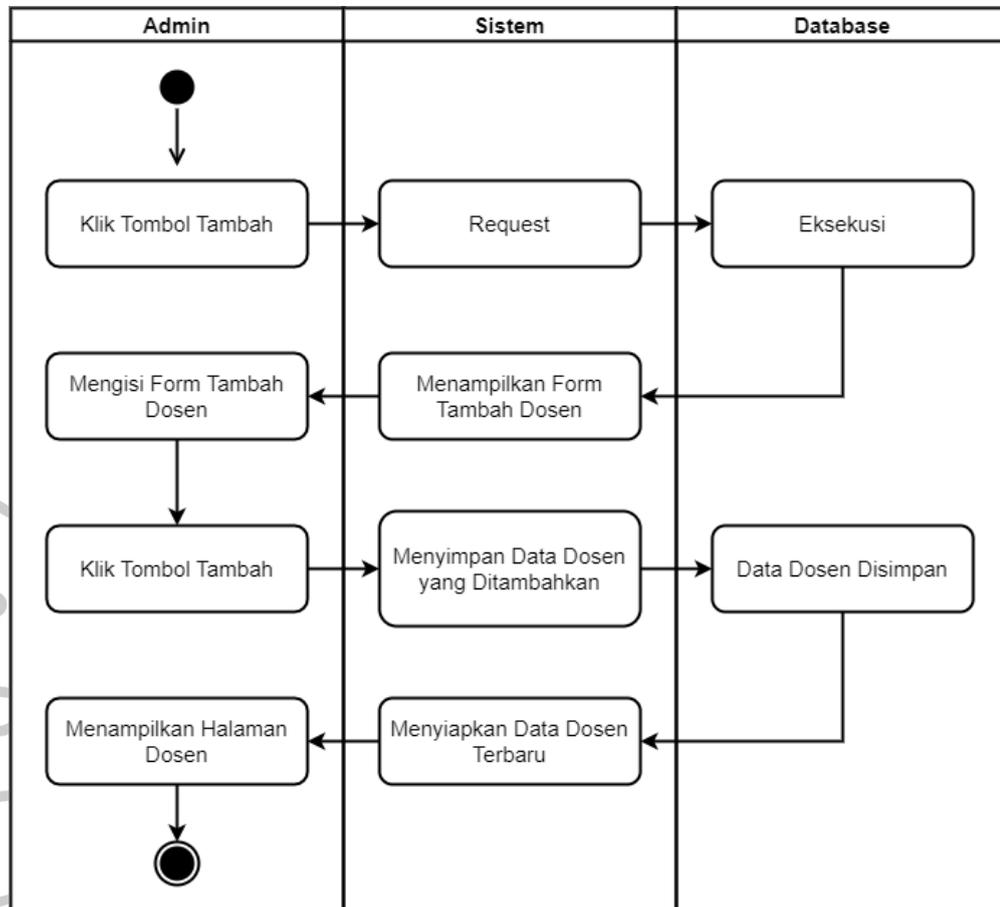
- *Activity* ini dimulai dari admin mengklik tombol hapus pada kolom aksi. Kemudian sistem akan menghapus data matakuliah, dan melanjutkan ke *database* untuk menghapus data. Setelah data terhapus, sistem akan menampilkan halaman matakuliah dengan data terbaru.



Gambar 3.2.1 15 Activity Diagram Menampilkan Dosen

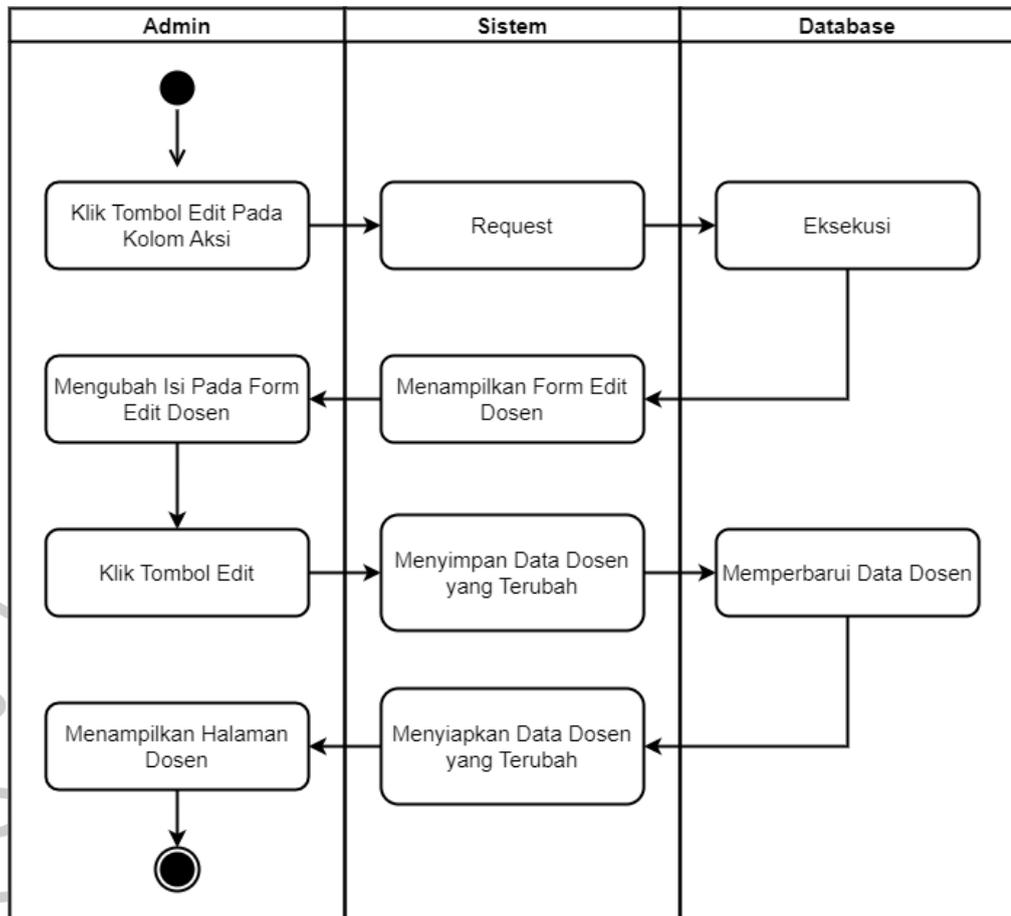
Gambar diatas merupakan *activity* diagram untuk menu dosen. *Activity* ini dimulai dari memilih menu dosen, kemudian mengirim *request* ke sistem untuk menampilkan halaman dosen. Sistem meneruskan ke *database* untuk dieksekusi. Setelah dieksekusi oleh *database*, sistem menyiapkan halaman

dosen yang datanya berasal dari *database*. Kemudian sistem akan menampilkan halaman dosen.



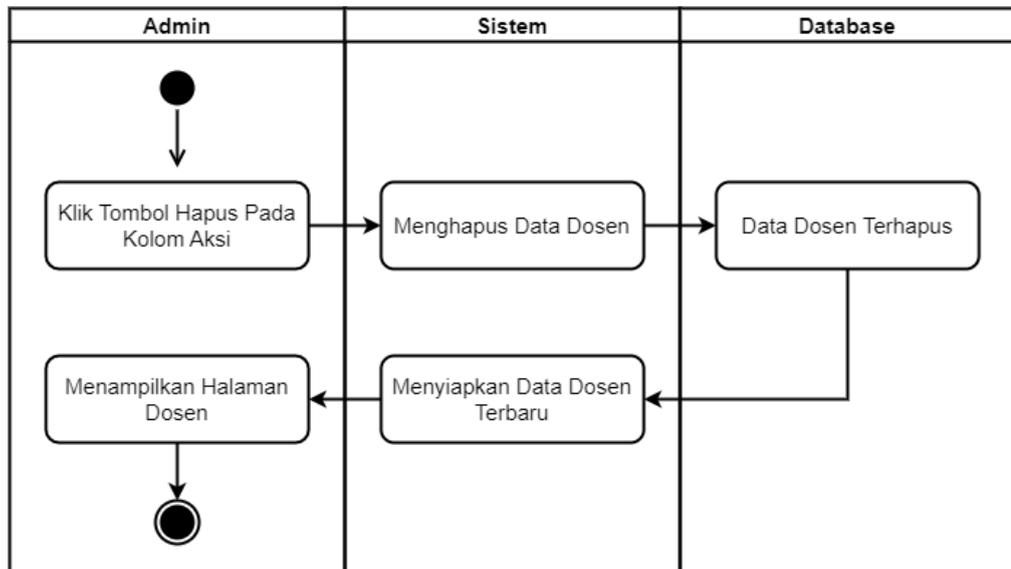
Gambar 3.2.1 16 Activity Diagram Menambah Data Dosen

Gambar diatas merupakan *activity* diagram menambah data dosen. *Activity* ini dimulai dari admin mengklik tombol tambah pada halaman dosen. Kemudian admin mengirim *request* ke sistem yang dilanjutkan ke *database* untuk dieksekusi. Setelah dieksekusi, sistem akan menampilkan *form* tambah dosen. Admin mengisi *form* dosen kemudian mengklik tombol tambah. Setelah data tersimpan di *database*, kemudian sistem akan menampilkan halaman dosen dengan data terbaru.



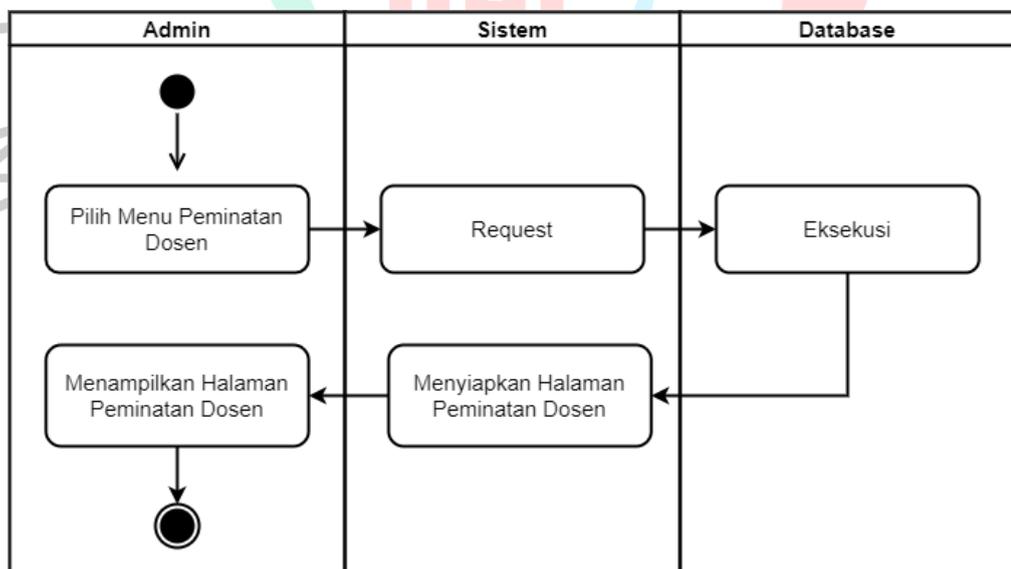
Gambar 3.2.1 17 Activity Diagram Mengubah Data Dosen

Gambar diatas merupakan *activity* diagram mengubah data dosen. *Activity* ini dimulai dari admin mengklik tombol *edit* pada kolom aksi. Kemudian admin mengirim *request* ke sistem yang dilanjutkan ke *database* untuk dieksekusi. Setelah dieksekusi, sistem akan menampilkan *form edit* dosen. Admin mengubah isi data pada *form edit* dosen kemudian mengklik tombol *edit*. Setelah memperbarui data dosen di *database*, kemudian sistem akan menampilkan halaman dosen dengan data yang sudah diubah.



Gambar 3.2.1 18 Activity Diagram Menghapus Data Dosen

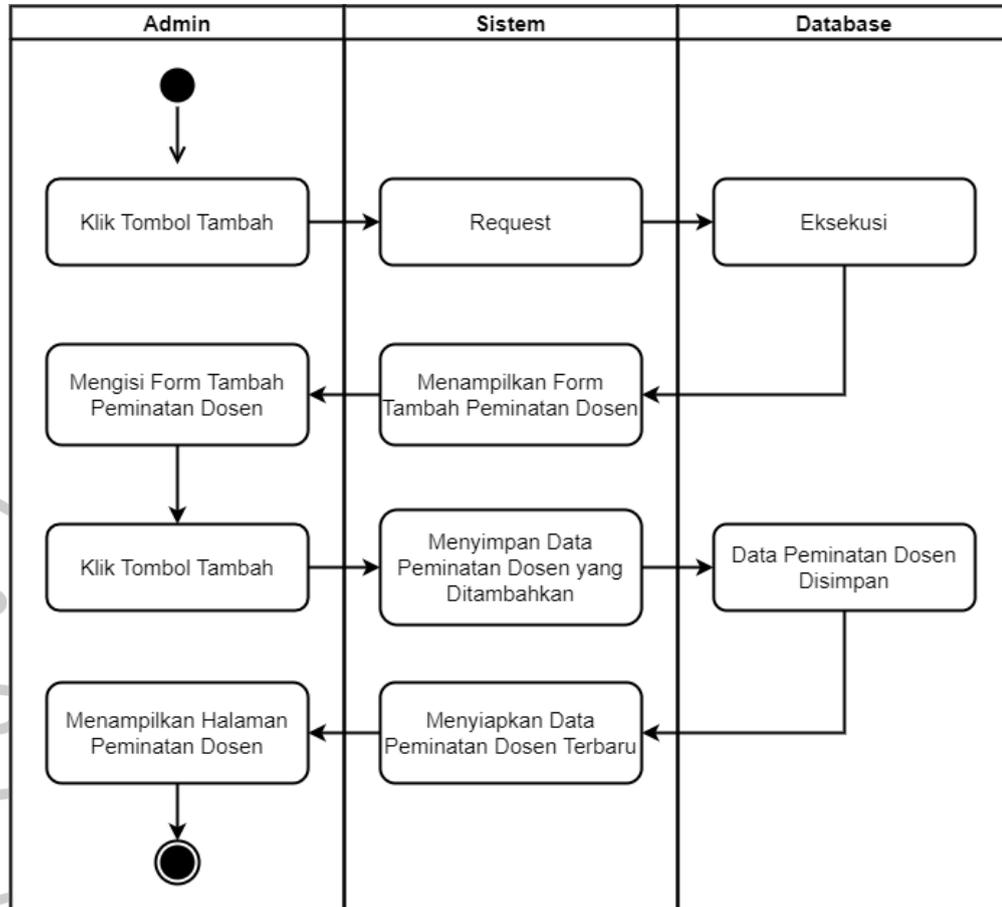
Gambar diatas merupakan *activity* diagram menghapus data dosen. *Activity* ini dimulai dari admin mengklik tombol hapus pada kolom aksi. Kemudian sistem akan menghapus data dosen, dan melanjutkan ke *database* untuk menghapus data. Setelah data terhapus, sistem akan menampilkan halaman dosen dengan data terbaru.



Gambar 3.2.1 19 Activity Diagram Menampilkan Peminatan Dosen

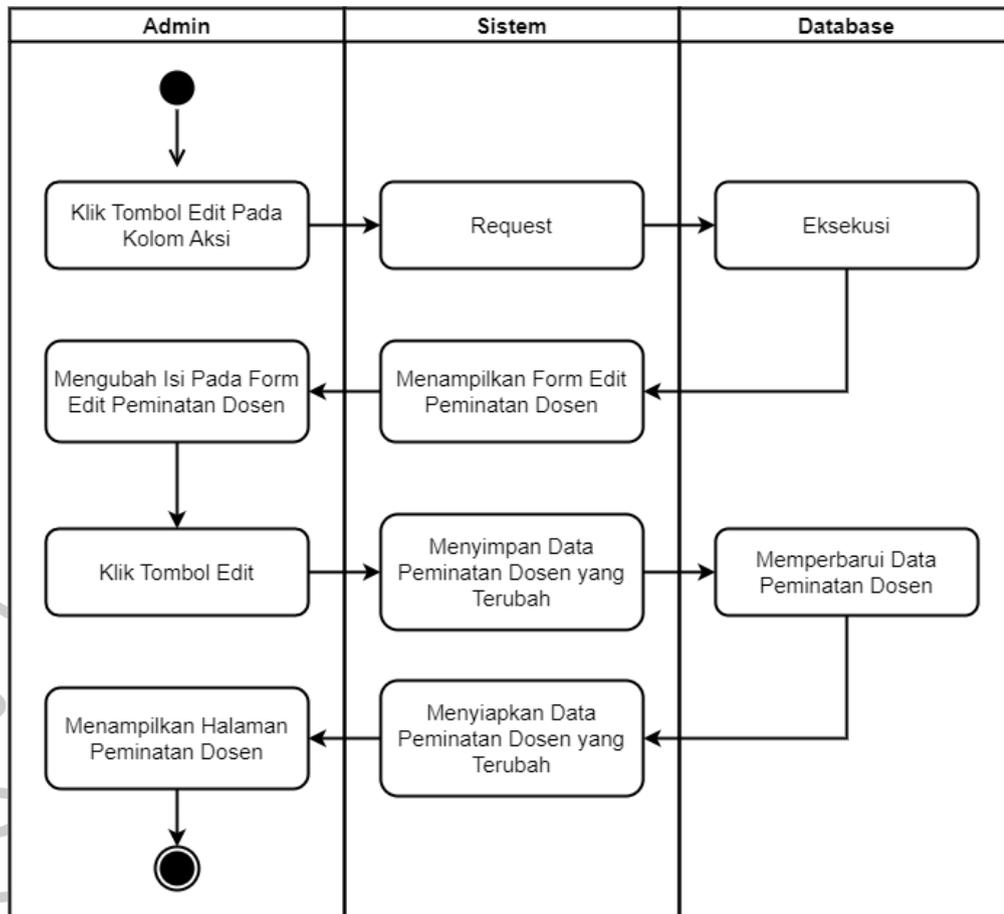
Gambar diatas merupakan *activity* diagram untuk menu peminatan dosen. *Activity* ini dimulai dari memilih menu peminatan dosen, kemudian mengirim *request* ke sistem untuk menampilkan halaman peminatan dosen. Sistem meneruskan ke *database* untuk dieksekusi. Setelah dieksekusi oleh *database*,

sistem menyiapkan halaman peminatan dosen yang datanya berasal dari *database*. Kemudian sistem akan menampilkan halaman peminatan dosen.



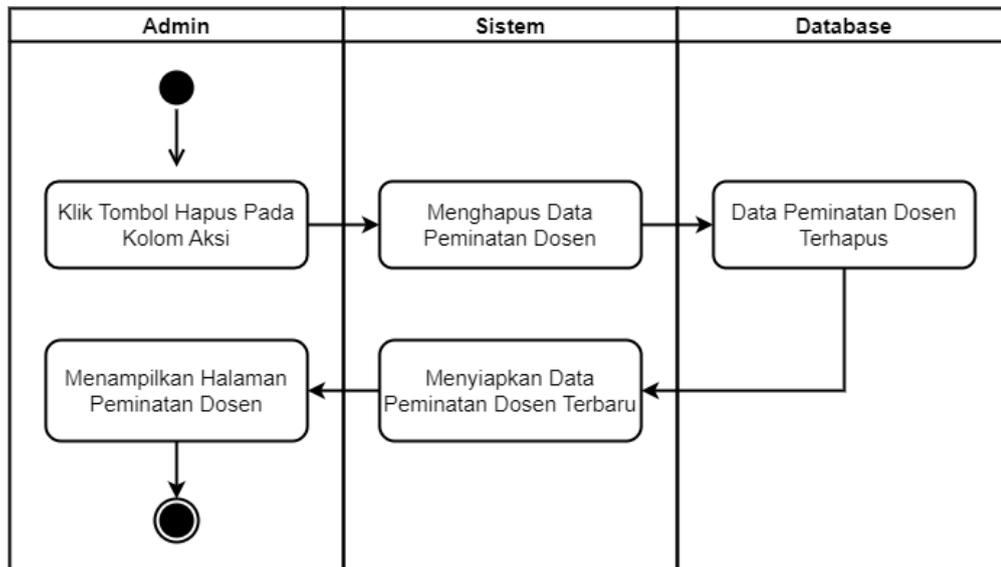
Gambar 3.2.1 20 Activity Diagram Menambah Data Peminatan Dosen

Gambar diatas merupakan *activity* diagram menambah data peminatan dosen. *Activity* ini dimulai dari admin mengklik tombol tambah pada halaman peminatan dosen. Kemudian admin mengirim *request* ke sistem yang dilanjutkan ke *database* untuk dieksekusi. Setelah dieksekusi, sistem akan menampilkan *form* tambah peminatan dosen. Admin mengisi *form* peminatan dosen kemudian mengklik tombol tambah. Setelah data tersimpan di *database*, kemudian sistem akan menampilkan halaman peminatan dosen dengan data terbaru.



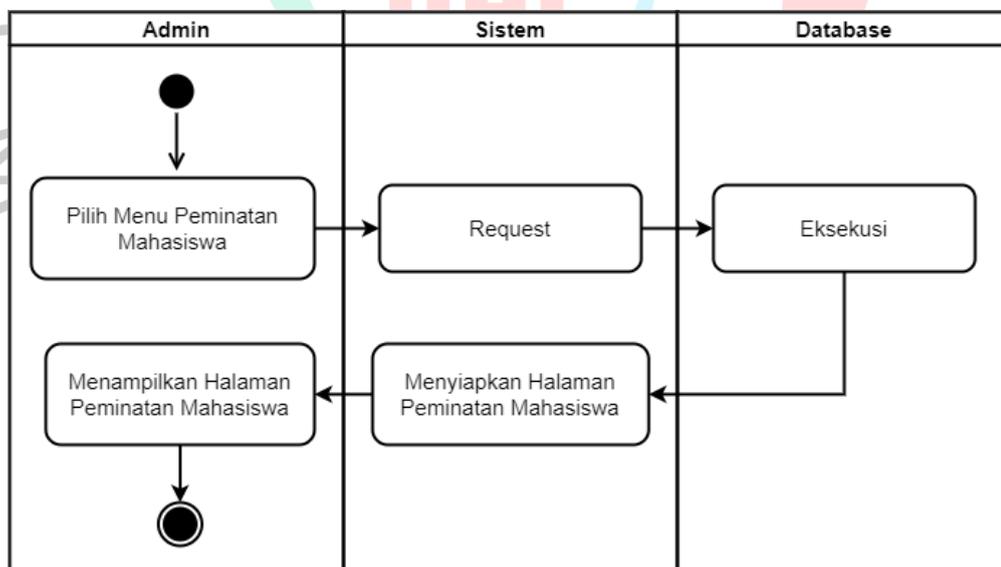
Gambar 3.2.1 21 Activity Diagram Mengubah Data Peminatan Dosen

Gambar diatas merupakan *activity* diagram mengubah data peminatan dosen. *Activity* ini dimulai dari admin mengklik tombol *edit* pada kolom aksi. Kemudian admin mengirim *request* ke sistem yang dilanjutkan ke *database* untuk dieksekusi. Setelah dieksekusi, sistem akan menampilkan *form edit* peminatan dosen. Admin mengubah isi data pada *form edit* peminatan dosen kemudian mengklik tombol *edit*. Setelah memperbarui data peminatan dosen di *database*, kemudian sistem akan menampilkan halaman peminatan dosen dengan data yang sudah diubah.



Gambar 3.2.1 22 Activity Diagram Menghapus Data Peminatan Dosen

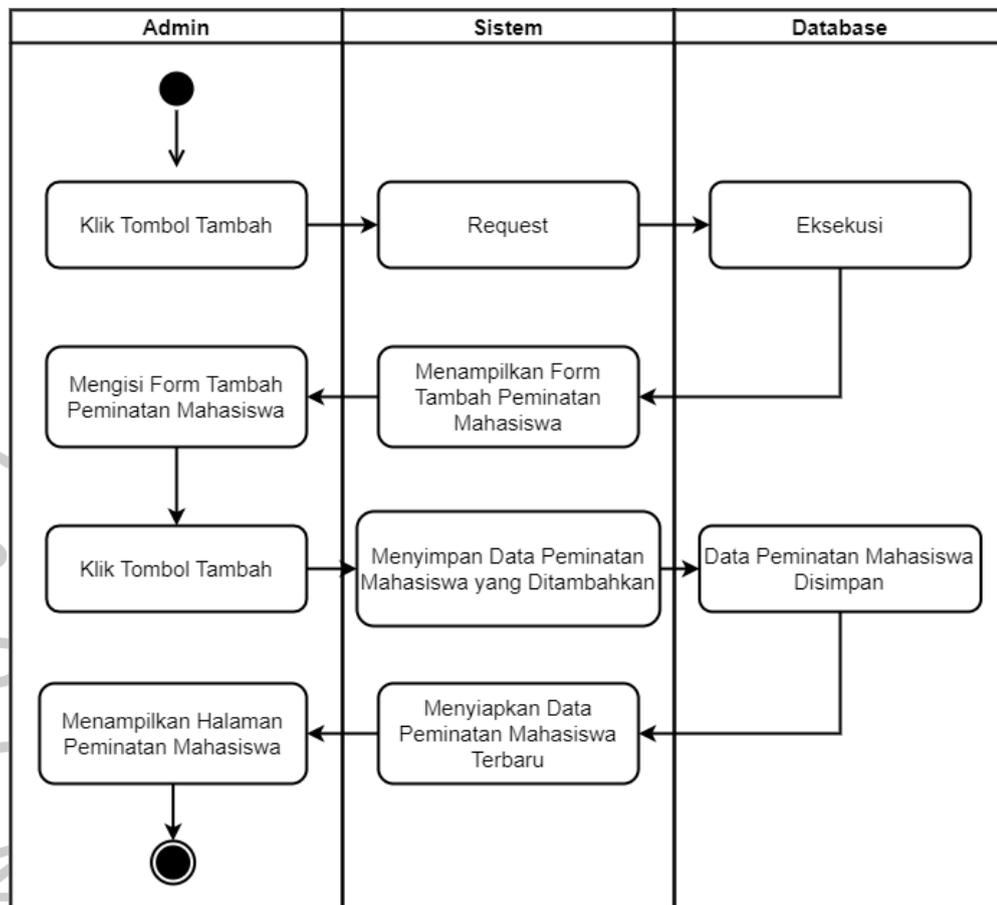
Gambar diatas merupakan *activity* diagram menghapus data peminatan dosen. *Activity* ini dimulai dari admin mengklik tombol hapus pada kolom aksi. Kemudian sistem akan menghapus data peminatan dosen, dan melanjutkan ke *database* untuk menghapus data. Setelah data terhapus, sistem akan menampilkan halaman peminatan dosen dengan data terbaru.



Gambar 3.2.1 23 Activity Diagram Menampilkan Peminatan Mahasiswa

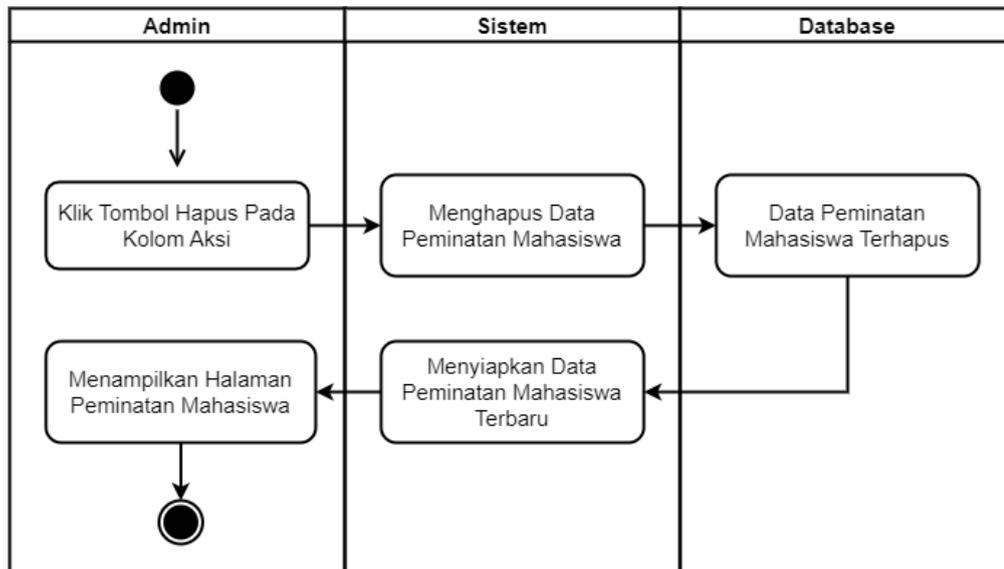
Gambar diatas merupakan *activity* diagram untuk menu peminatan mahasiswa. *Activity* ini dimulai dari memilih menu peminatan mahasiswa, kemudian mengirim *request* ke sistem untuk menampilkan halaman peminatan mahasiswa. Sistem meneruskan ke *database* untuk dieksekusi. Setelah

dieksekusi oleh *database*, sistem menyiapkan halaman peminatan mahasiswa yang datanya berasal dari *database*. Kemudian sistem akan menampilkan halaman peminatan mahasiswa.



Gambar 3.2.1 24 Activity Diagram Menambah Data Peminatan Mahasiswa

Gambar diatas merupakan *activity* diagram menambah data peminatan mahasiswa. *Activity* ini dimulai dari admin mengklik tombol tambah pada halaman peminatan mahasiswa. Kemudian admin mengirim *request* ke sistem yang dilanjutkan ke *database* untuk dieksekusi. Setelah dieksekusi, sistem akan menampilkan *form* tambah peminatan mahasiswa. Admin mengisi *form* peminatan mahasiswa kemudian mengklik tombol tambah. Setelah data tersimpan di *database*, kemudian sistem akan menampilkan halaman peminatan mahasiswa dengan data terbaru.



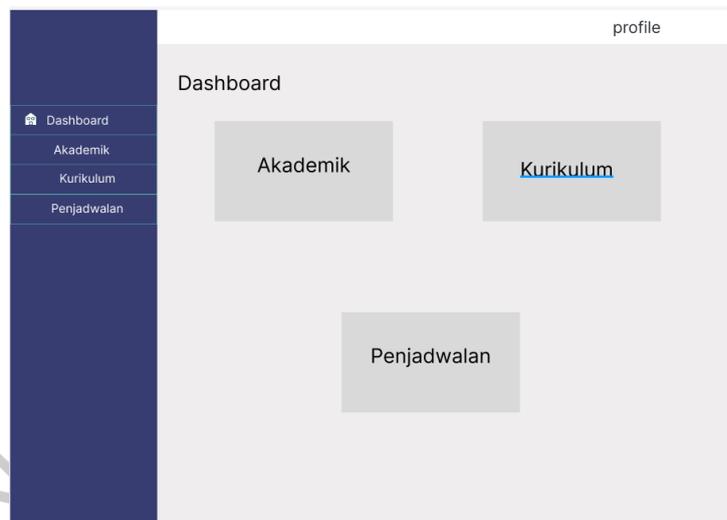
Gambar 3.2.1 25 Activity Diagram Menghapus Data Peminatan Mahasiswa

Gambar diatas merupakan *activity* diagram menghapus data peminatan mahasiswa. *Activity* ini dimulai dari *admin* mengklik tombol hapus pada kolom aksi. Kemudian sistem akan menghapus data peminatan mahasiswa, dan melanjutkan ke *database* untuk menghapus data. Setelah data terhapus, sistem akan menampilkan halaman peminatan mahasiswa dengan data terbaru.

4. Perancangan *Mock Up*

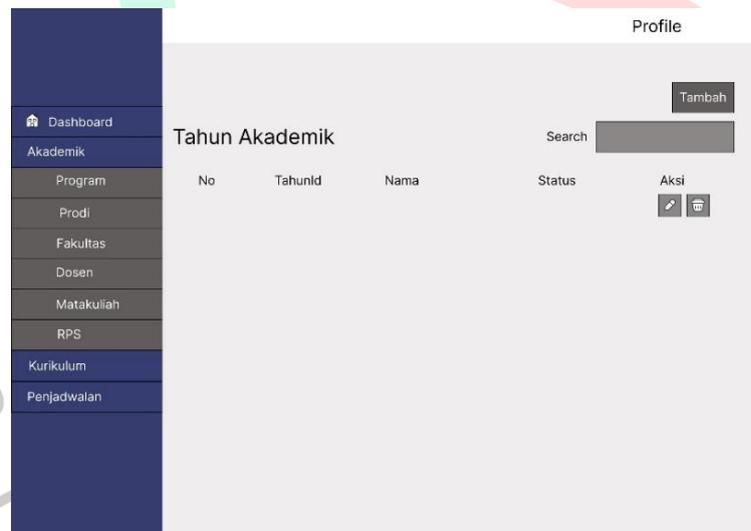
Perancangan *mock up* adalah perancangan yang dibuat untuk memudahkan *programmer* untuk mendesain aplikasi yang diinginkan. Untuk mendapatkan tampilan aplikasi yang bagus, diperlukan rancangan *mock up* yang detail dan rapih sehingga programmer lebih mudah untuk menentukan desain untuk aplikasi yang dibuat.

Berikut adalah rancangan *mock up* untuk aplikasi sistem evaluasi akademik.



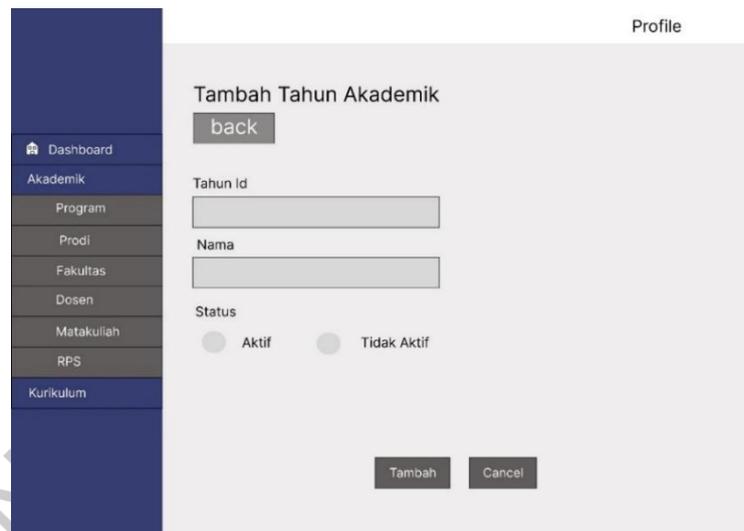
Gambar 3.2.1 26 Perancangan Mock Up Dashboard

Gambar diatas merupakan halaman dari *dashboard* yang merupakan tampilan utama dari aplikasi ini. Halaman ini terdapat pilihan-pilihan menu utama dari aplikasi sistem evaluasi mahasiswa.



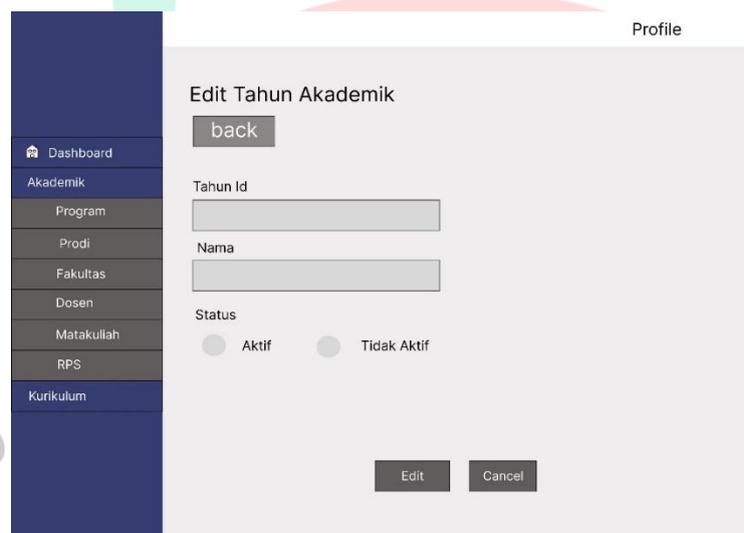
Gambar 3.2.1 27 Perancangan Mock Up Tahun Akademik

Gambar diatas merupakan halaman dari tahun akademik. Halaman ini dapat menampilkan tahunid, nama, status, dan aksi.



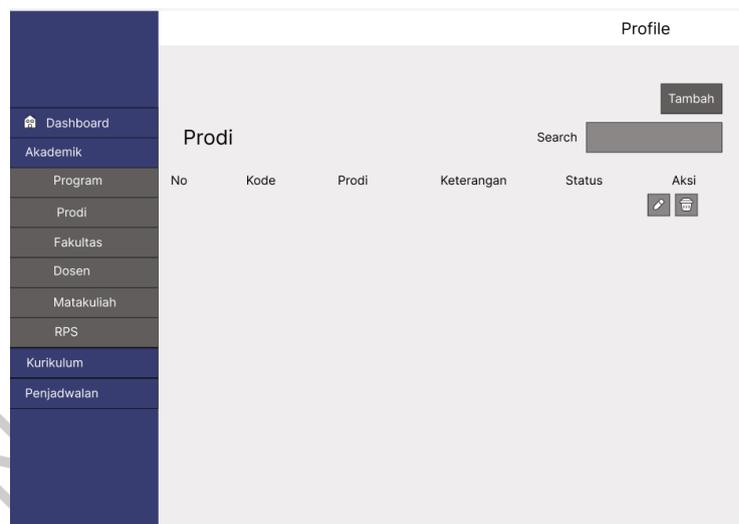
Gambar 3.2.1 28 Perancangan *Mock Up* Tambah Tahun Akademik

Gambar diatas merupakan halaman tambah dari tahun akademik. Halaman ini dapat *input* tahunId, nama, dan status.



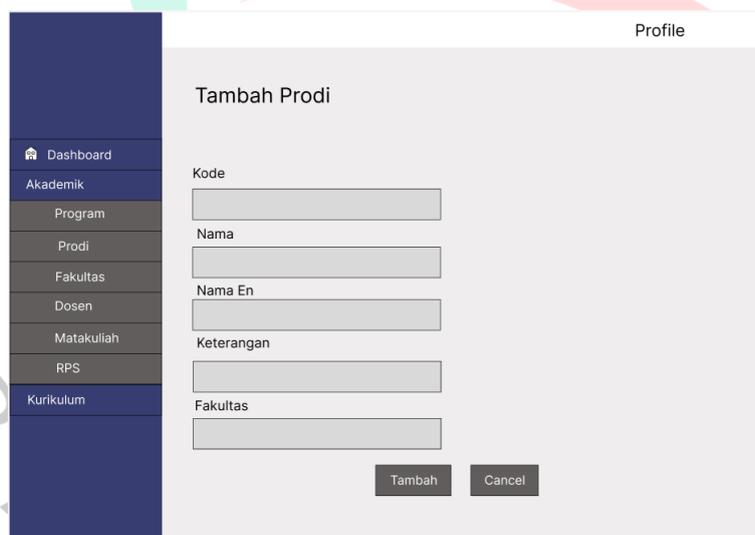
Gambar 3.2.1 29 Perancangan *Mock Up* Edit Tahun Akademik

Gambar diatas merupakan halaman untuk mengubah data yang berada di menu tahun akademik. Data yang muncul di halaman ini berasal dari *database*. Setelah data diubah kemudian akan tersimpan kembali ke *database*.



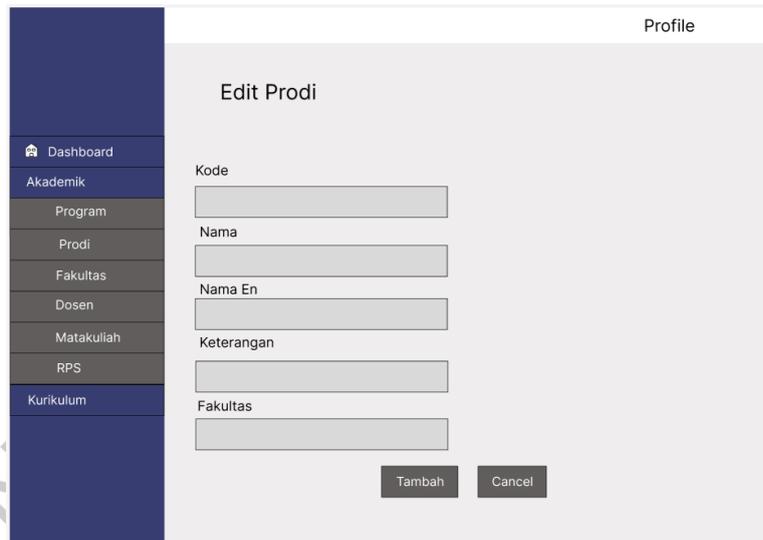
Gambar 3.2.1 30 Perancangan *Mock up* Prodi

Gambar diatas merupakan halaman dari tahun akademik, pada halaman ini menampilkan kode, prodi, keterangan, status, dan aksi.



Gambar 3.2.1 31 Perancangan *Mock Up* Tambah Prodi

Gambar diatas merupakan halaman untuk mengubah data yang berada di menu prodi. Data yang muncul di halaman ini berasal dari *database*. Setelah data diubah kemudian akan tersimpan kembali ke *database*.



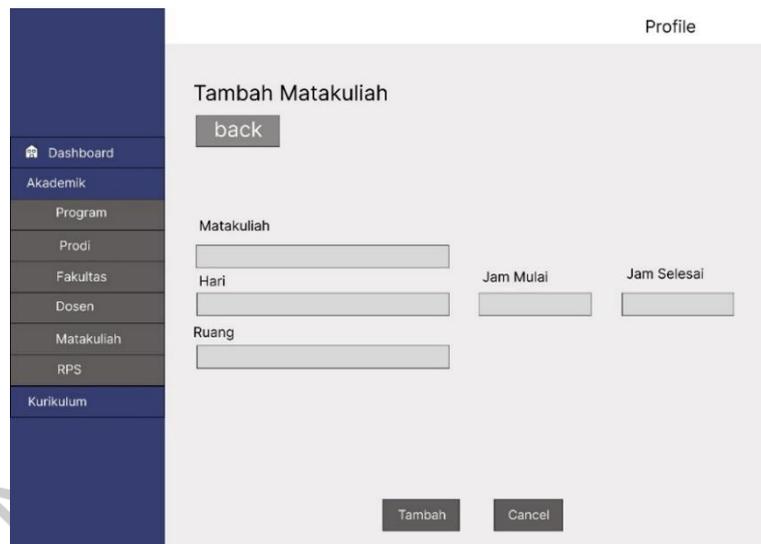
Gambar 3.2.1 32 Perancangan *Mock Up Edit Prodi*

Gambar diatas merupakan halaman untuk mengubah data yang berada di menu prodi. Data yang muncul di halaman ini berasal dari *database*. Setelah data diubah kemudian akan tersimpan kembali ke *database*.



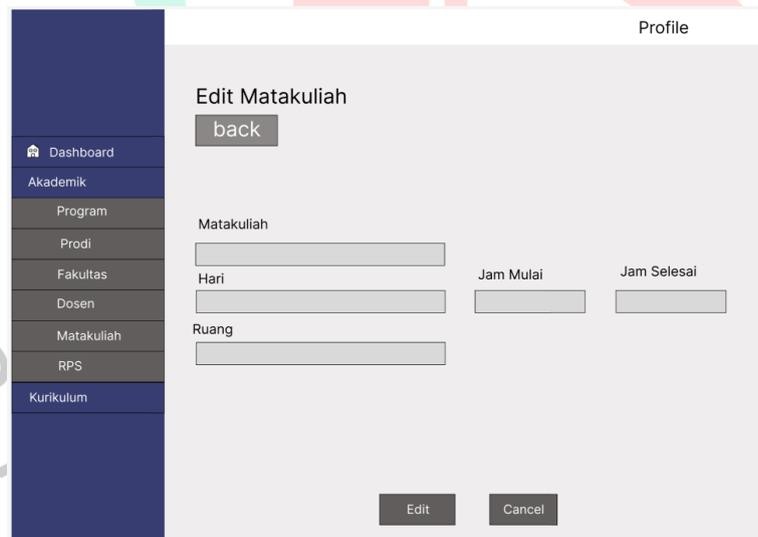
Gambar 3.2.1 33 Perancangan *Mock Up Matakuliah*

Gambar diatas merupakan halaman untuk matakuliah, halaman ini menampilkan Nama Matakuliah, Hari, Ruang, Jam Mulai, Jam Selesai, dan Aksi.



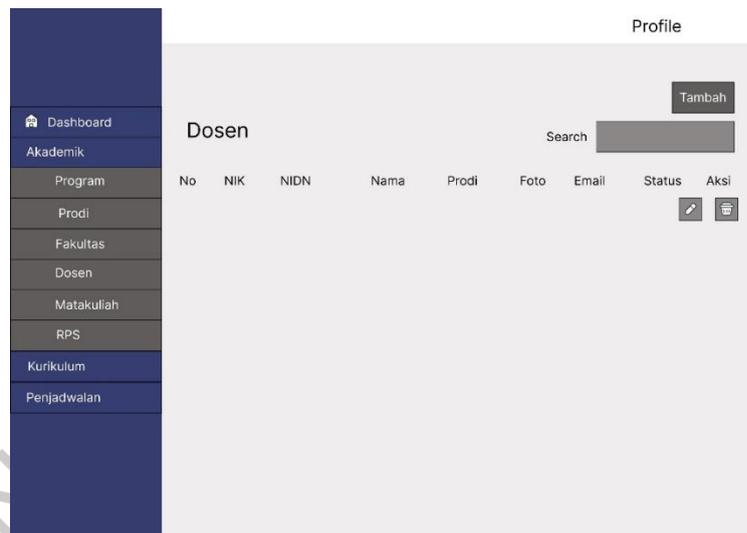
Gambar 3.2.1 34 Perancangan *Mock Up* Tambah Matakuliah

Gambar diatas merupakan halaman untuk tambah matakuliah, untuk *input* hari menggunakan pilihan hari dan untuk *input* ruang mengambil dari *database* ruang. *Input* jam mulai dan jam selesai menggunakan *javascript*.



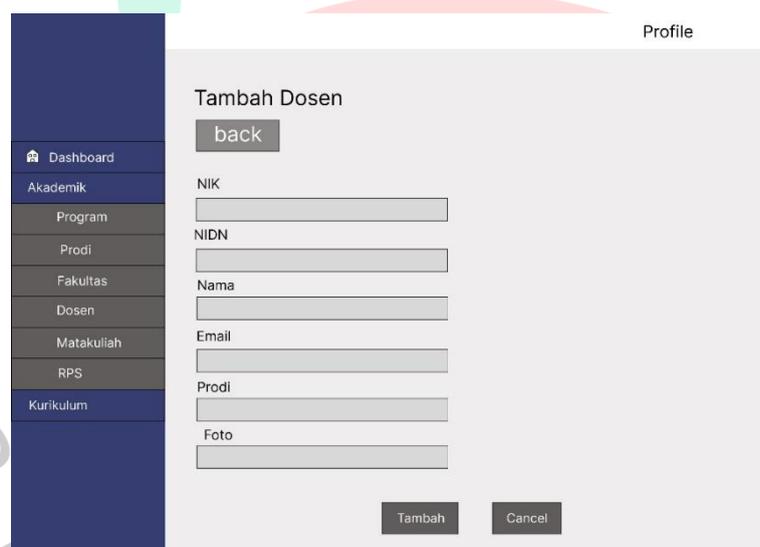
Gambar 3.2.1 35 Perancangan *Mock Up* Edit Matakuliah

Gambar diatas merupakan halaman untuk *edit* matakuliah, untuk edit hari menggunakan pilihan hari dan untuk *edit* ruang mengambil dari *database* ruang. *Edit* jam mulai dan jam selesai menggunakan *javascript*.



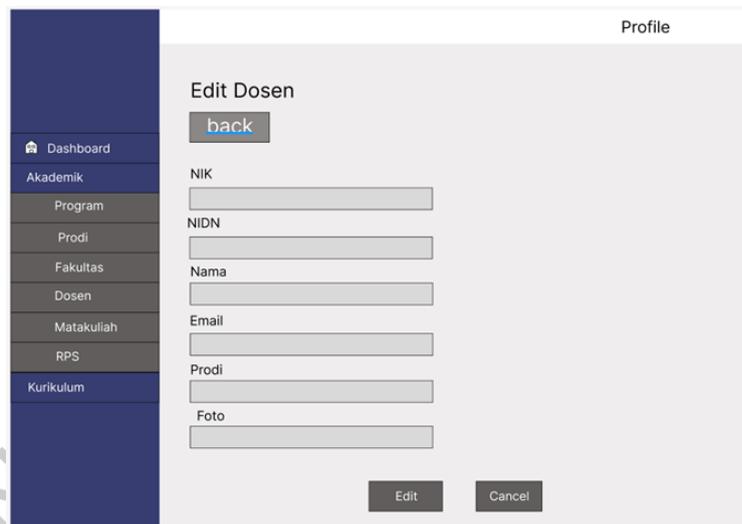
Gambar 3.2.1 36 Perancangan *Mock Up* Dosen

Gambar diatas merupakan halaman dari dosen. Halaman ini menampilkan NIK, NIDN, Nama, Prodi, Foto, Email, Status, dan aksi.



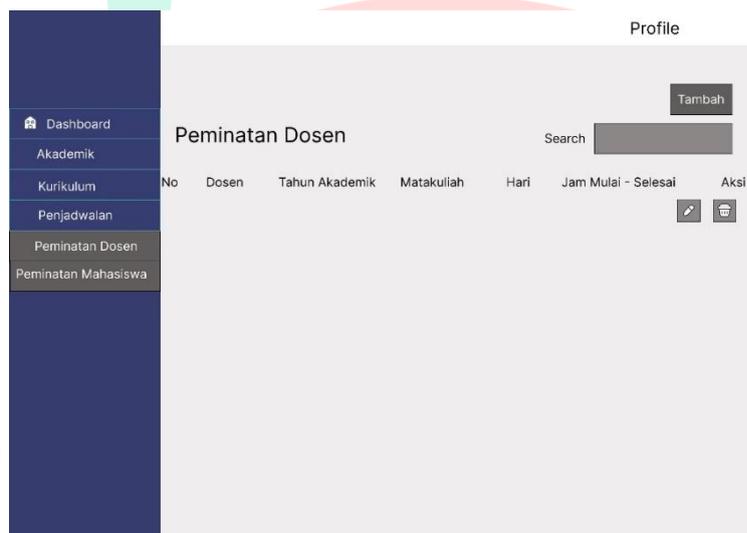
Gambar 3.2.1 37 Perancangan *Mock Up* Tambah Dosen

Gambar diatas merupakan halaman untuk *edit* dosen, untuk *edit* prodi mengambil *database* dari prodi.



Gambar 3.2.1 38 Perancangan *Mock Up* Edit Dosen

Gambar diatas merupakan halaman untuk *edit* dosen, untuk *edit* prodi mengambil *database* dari prodi.



Gambar 3.2.1 39 Perancangan *Mock Up* Peminatan Dosen

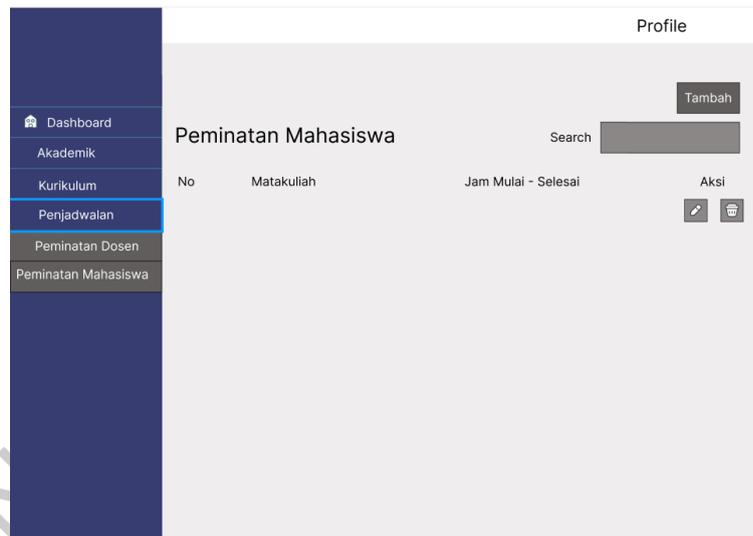
Gambar diatas merupakan halaman untuk peminatan dosen, halaman ini menampilkan dosen, tahun akademik, matakuliah, hari, jam mulai, dan jam selesai.

Gambar 3.2.1 40 Perancangan *Mock Up* Tambah Peminatan Dosen

Gambar diatas merupakan halaman untuk tambah peminatan dosen, untuk *input* dosen menggunakan database dosen, untuk matakuliah menggunakan *database* matakuliah, untuk ruang menggunakan database ruang. *Input* jam mulai dan jam selesai menggunakan *javascript*.

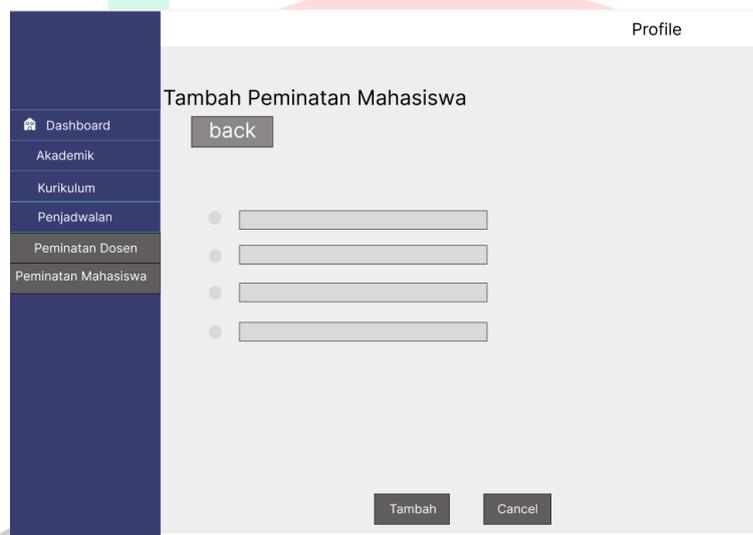
Gambar 3.2.1 41 Perancangan *Mock Up* Edit Peminatan Dosen

Gambar diatas merupakan halaman untuk peminatan dosen, untuk *edit* dosen menggunakan database dosen, untuk *edit* matakuliah menggunakan database matakuliah, untuk *edit* ruang menggunakan database ruang. *Edit* jam mulai dan jam selesai menggunakan *javascript*.



Gambar 3.2.1 42 Perancangan *Mock Up* Peminatan Mahasiswa

Gambar diatas merupakan halaman untuk peminatan mahasiswa, halaman ini menampilkan matakuliah, jam mulai, dan jam selesai, aksi.

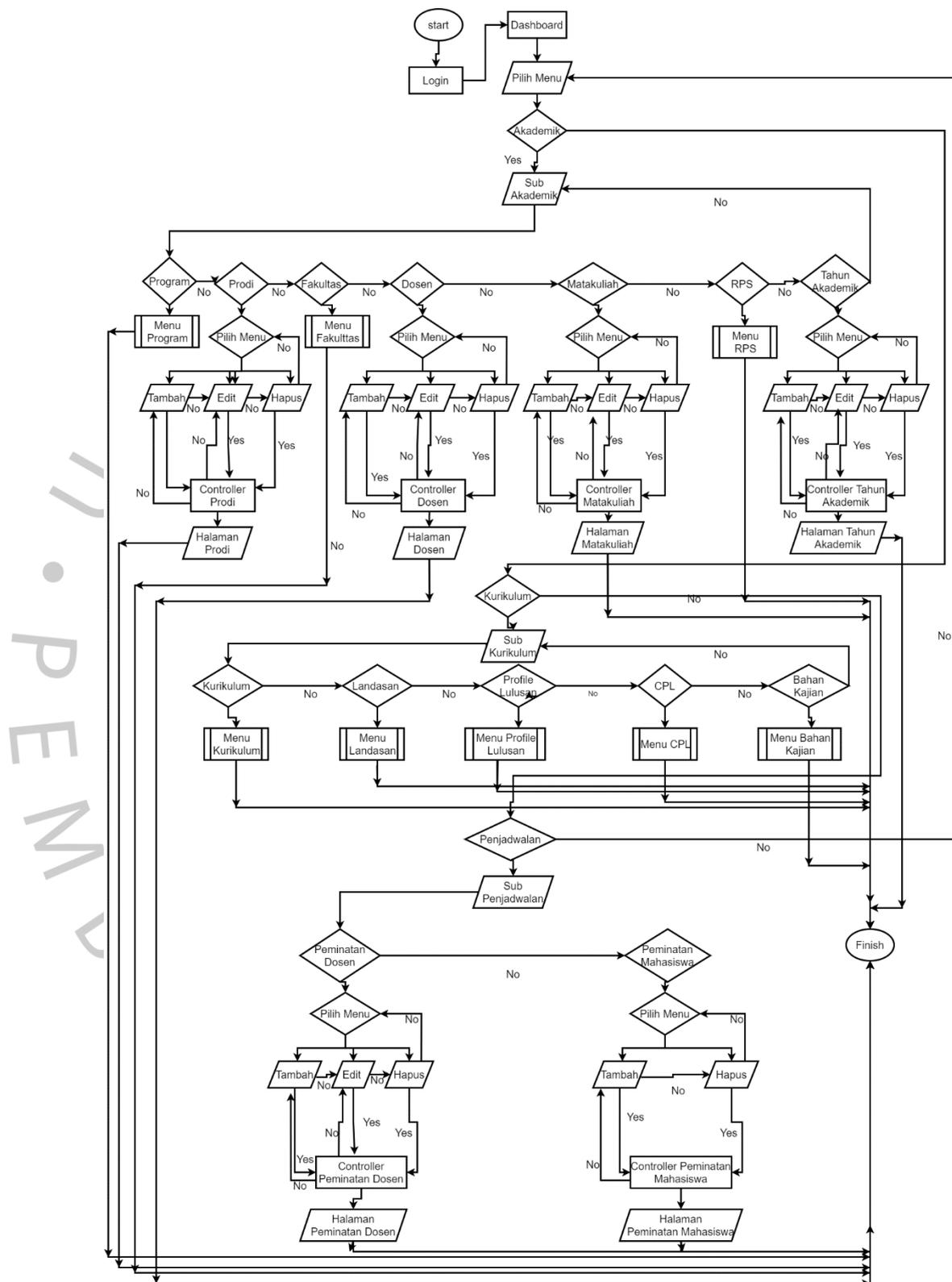


Gambar 3.2.1 43 Perancangan *Mock Up* Tambah Peminatan Mahasiswa

Gambar diatas merupakan halaman untuk tambah peminatan mahasiswa, untuk *input* pada halaman ini berdasarkan menu peminatan dosen yang telah divalidasi.

5. Flowchart

Flowchart adalah rancangan yang dibuat untuk menentukan alur dari aplikasi yang dibuat. Semakin jelas rancangan yang dibuat maka akan memudahkan programmer dalam membuat aplikasi.



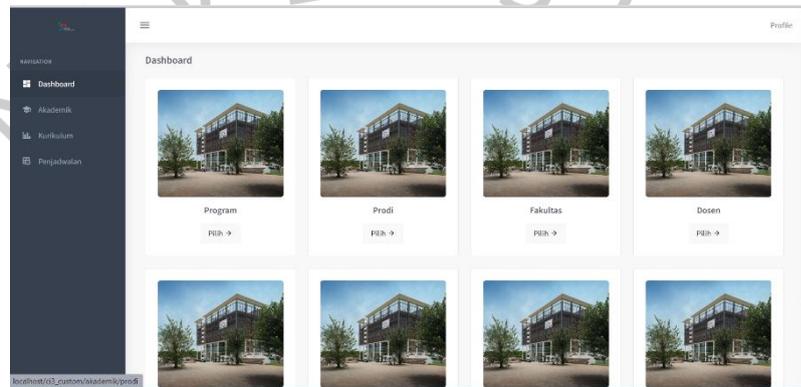
Gambar 3.2.1 44 Flowchart

3.2.2 Implementasi

Implementasi adalah hasil yang diperoleh dari perancangan yang telah dibuat sebelumnya. Implementasi pada laporan ini adalah bagian pekerjaan yang dibuat oleh praktikan. Berikut adalah implementasi dari aplikasi SIM OBE.

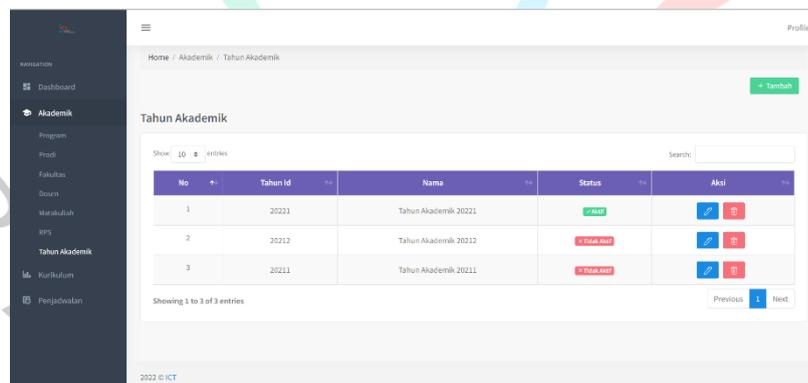
1. Implementasi *Front End*

Implementasi *Front End* merupakan realiasi dari perancangan front end yang telah dibuat. Pada implementasi ini dibuat semirip mungkin dengan perancangan *front end*. Berikut adalah Implementasi *front end*.



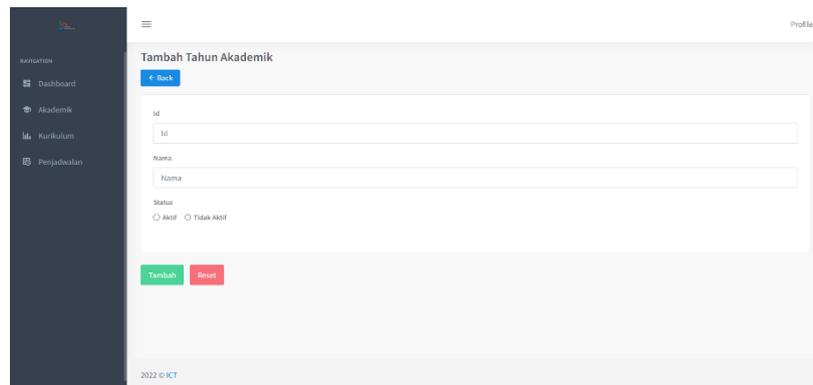
Gambar 3.2.2 1 Implementasi *dashboard*

Gambar diatas merupakan implementasi *dashboard* yang telah dibuat pada perancangan front end.



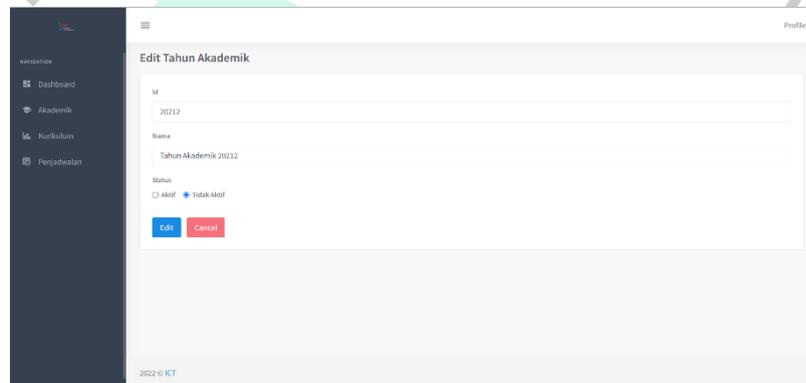
Gambar 3.2.2 2 Implementasi Halaman Utama Tahun Akademik

Gambar diatas merupakan halaman utama tahun akademik, pada halaman tersebut dapat mengklik tombol tambah, *edit*, dan hapus.



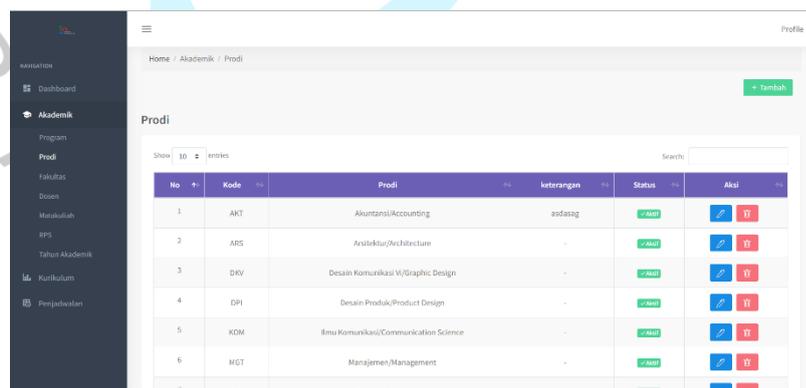
Gambar 3.2.2 3 Implementasi Tambah Tahun Akademik

Gambar diatas merupakan halaman tambah tahun akademik, pada halaman tersebut dapat menambahkan data tahun id, nama, dan status.



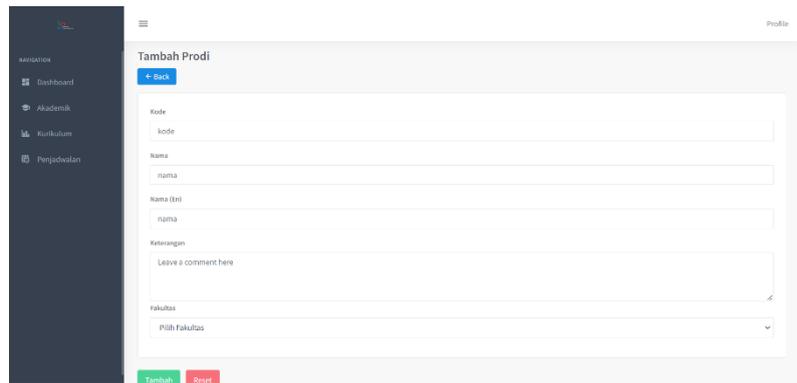
Gambar 3.2.2 4 Implementasi Edit Tahun Akademik

Gambar diatas merupakan halaman *edit* tahun akademik, pada halaman tersebut dapat mengubah data tahun id, nama, dan status.



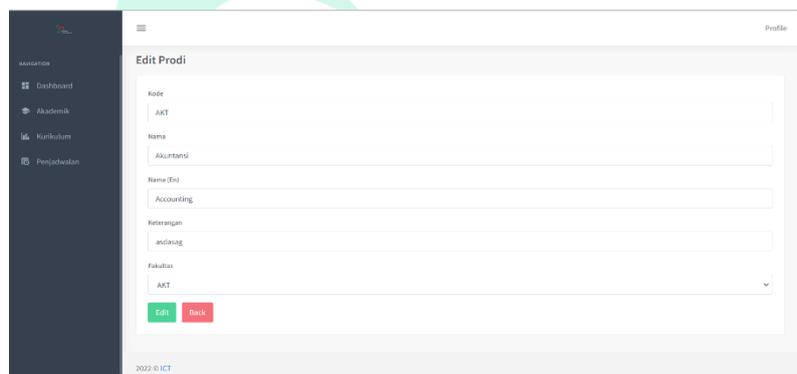
Gambar 3.2.2 5 Implementasi Halaman Utama Prodi

Gambar diatas merupakan halaman utama dari menu prodi. Pada halaman ini dapat mengklik tombol tambah, *edit*, dan hapus.



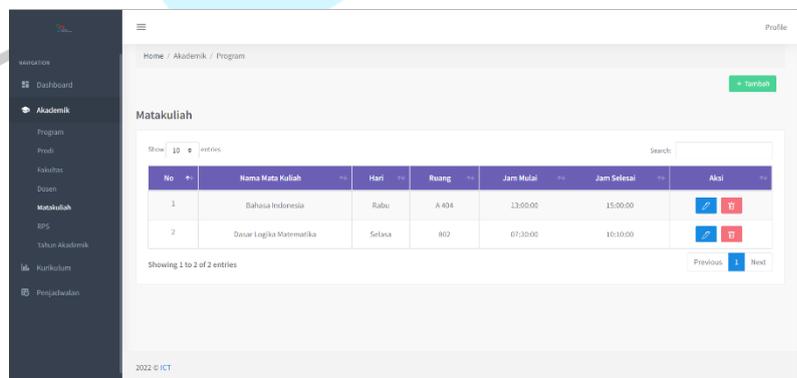
Gambar 3.2.2 6 Implementasi Tambah Prodi

Gambar diatas merupakan halaman tambah padad menu prodi. Pada halaman ini dapat menambahkan data kode, nama, nama En, keterangan, fakultas.



Gambar 3.2.2 7 Implementasi Edit Prodi

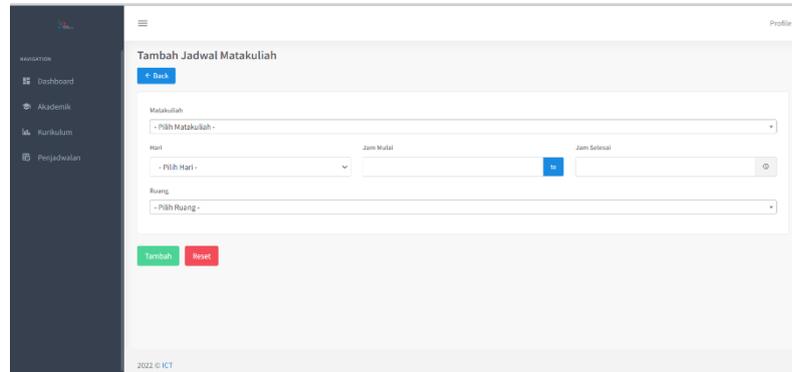
Gambar diatas merupakan halaman *edit* dari menu prodi. Pada halaman ini dapat mengubah data kode, nama, nama En, keterangan, fakultas.



Gambar 3.2.2 8 Implementasi Halaman Utama Matakuliah

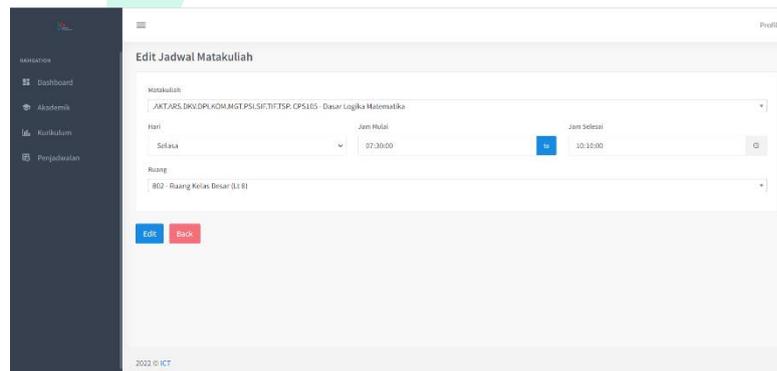
Gambar diatas merupakan halaman utama pada menu matakuliah,

pada halaman tersebut dapat mengklik tombol tambah, *edit*, dan hapus.



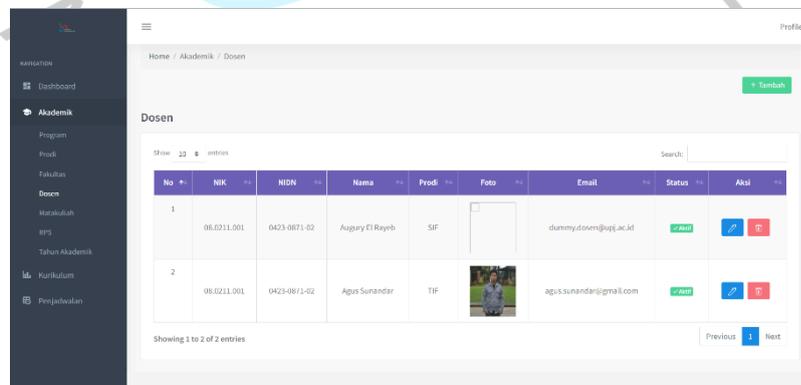
Gambar 3.2.2 9 Implementasi Tambah Matakuliah

Gambar diatas merupakan halaman tambah jadwal matakuliah, pada halaman tersebut dapat menambahkan data matakuliah, hari, jam mulai, jam selesai, ruang



Gambar 3.2.2 10 Implementasi *Edit* Matakuliah

Gambar diatas merupakan halaman *edit* jadwal matakuliah, pada halaman tersebut dapat mengubah data matakuliah, hari, jam mulai, jam selesai, ruang.



Gambar 3.2.2 11 Implementasi Halaman Utama Dosen

Gambar diatas merupakan halaman utama pada menu dosen, pada

halaman tersebut dapat mengklik tombol tambah, *edit*, dan hapus.

The screenshot shows a web form titled "Tambah Dosen". It has a dark sidebar on the left with navigation options: Dashboard, Akademik, Kurikulum, and Penjadwalan. The form fields are: NIK (input), NIDN (input), Nama (input), Email (input), Prodi (dropdown menu), and Foto (file upload area with "Choose File" and "No file chosen" text). At the bottom, there are "Tambah" (green) and "Reset" (red) buttons.

Gambar 3.2.2 12 Implementasi Tambah Dosen

Gambar diatas merupakan halaman tambah dosen, pada halaman tersebut dapat menambahkan data nik, nidn, nama, email, prodi, foto.

The screenshot shows the "Edit Dosen" form. The data is pre-filled: Nama: Agus Sunandar, NIK: 08.0211.001, NIDN: 0423-0871-02, Email: agus.sunandar@gmail.com, Prodi: TIF. There is a photo upload area with a "Choose File" button and "No file chosen" text. At the bottom, there are "Edit" (blue) and "Cancel" (red) buttons.

Gambar 3.2.2 13 Implementasi *Edit* Dosen

Gambar diatas merupakan halaman *edit* dosen, pada halaman tersebut dapat mengubah data nik, nidn, nama, email, prodi, foto.

The screenshot shows the "Peminatan Dosen" page. It features a table with the following data:

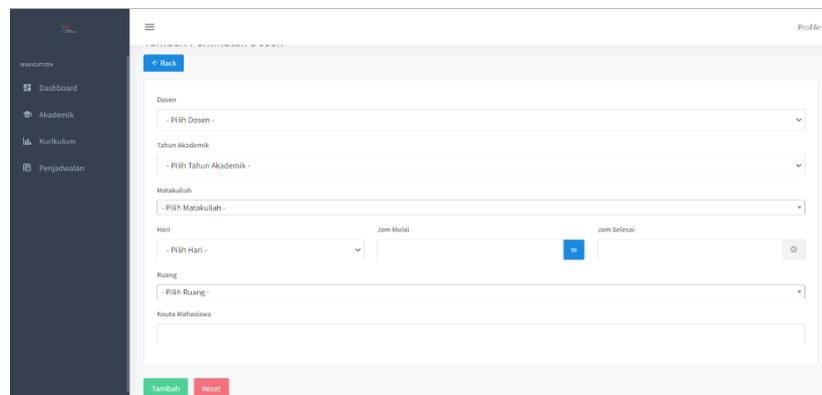
NO	Dosen	Tahun Akademik	Mata Kuliah	Hari	Jam Mulai - Selesai	Ruang
1	Ratna Saffri	20221	TIF - CPS101 - Bahasa Indonesia	Senin	13:00:00 - 15:00:00	A-104
2	a	20221	SIF - INS101 - Fondasi Pemrograman dan Struktur Data	Senin	07:00:00 - 10:30:00	A-104
3	Ratna Saffri	20221	TIF - CPS105 - Dasar Logika Matematika	Senin	13:05:00 - 14:30:00	A-104
4	Augury El Rayeb	20221	TIF - CPS103 - Bahasa Inggris	Senin	13:15:00 - 14:35:00	A-104
5	Ratna Saffri	20221	TIF - CPS103 - Bahasa Inggris	Jumat	13:00:00 - 15:00:00	A-105
6	Agus Sunandar	20221	TIF - CPS101 - Bahasa Indonesia	Jumat	13:00:00 - 15:30:00	A-301
7	Agus Sunandar	20221	TIF - CPS105 - Dasar Logika Matematika	Jumat	09:59:00 - 11:00:00	A-304

At the top right of the page is a "Tambah" button. The table has a search bar and pagination controls at the bottom.

Gambar 3.2.2 14 Halaman Utama Peminatan Dosen

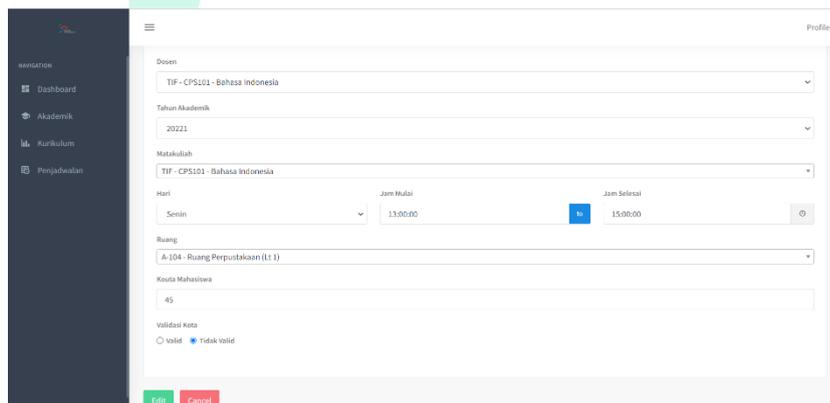
Gambar diatas merupakan halaman utama peminatan dosen, pada

halaman tersebut dapat mengklik tombol tambah, *edit*, dan hapus.



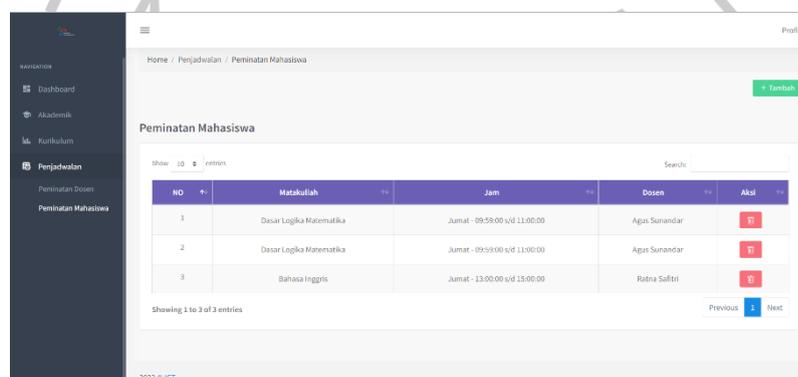
Gambar 3.2.2 15 Implementasi Tambah Peminatan Dosen

Gambar diatas merupakan halaman tambah peminatan dosen, pada halaman tersebut dapat menambahkan data dosen, tahun akademik, matakuliah, hari, jam mulai, jam selesai, ruang.



Gambar 3.2.2 16 Implementasi *Edit* Peminatan Dosen

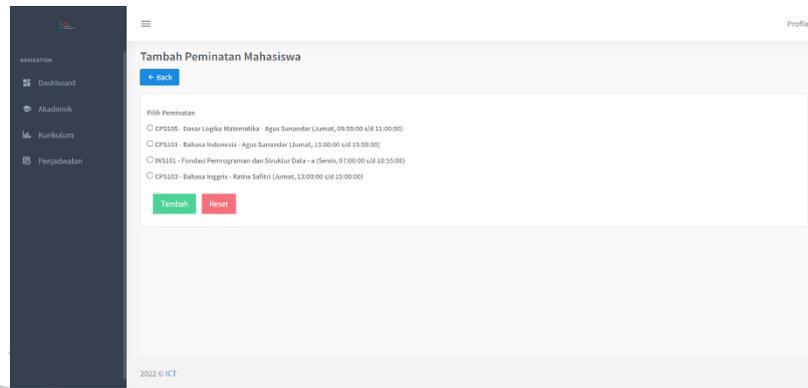
Gambar diatas merupakan halaman *edit* peminatan dosen, pada halaman tersebut dapat mengubah data dosen, tahun akademik, matakuliah, hari, jam mulai, jam selesai, ruang.



NO	Matakuliah	Jam	Dosen	Aksi
1	Dasar Logika Matematika	Jumat - 09:55:00 s/d 11:00:00	Agus Sunandar	[Edit]
2	Dasar Logika Matematika	Jumat - 09:55:00 s/d 11:00:00	Agus Sunandar	[Edit]
3	Bahasa Inggris	Jumat - 13:00:00 s/d 15:00:00	Ratna Salfiti	[Edit]

Gambar 3.2.2 17 Implementasi Peminatan Mahasiswa

Gambar diatas merupakan halaman utama peminatan mahasiswa, pada halaman tersebut dapat mengklik tombol tambah dan hapus.



Gambar 3.2.2 18 Implementasi Tambah Peminatan Mahasiswa

Gambar diatas merupakan halaman tambah peminatan mahasiswa, pada halaman tersebut data yang sudah valid pada peminatan dosen dapat muncul pada halaman tersebut.

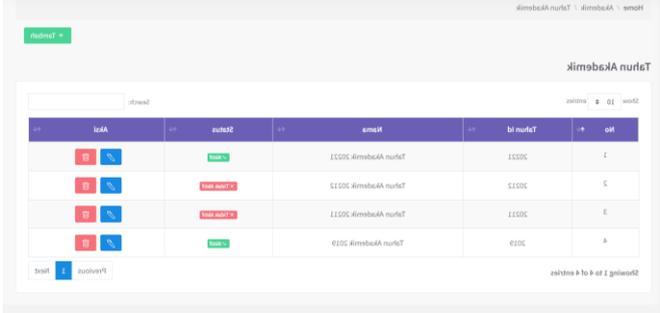
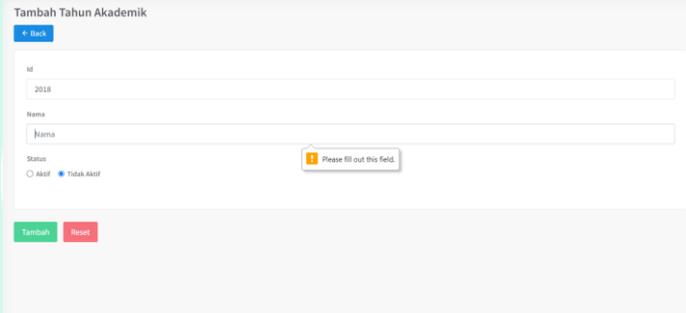
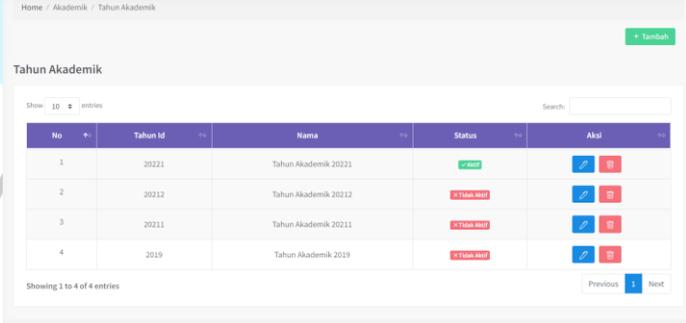
2. Pengujian dengan Metode *Black Box*

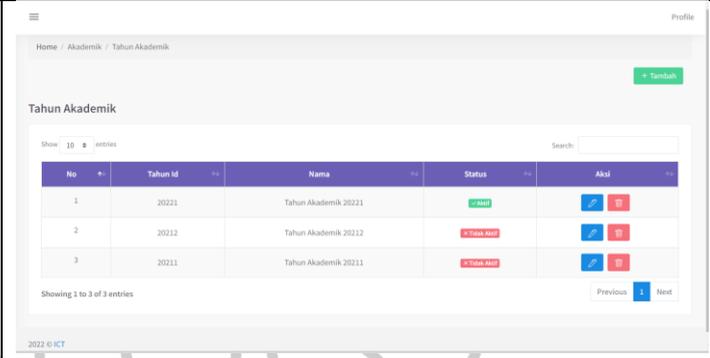
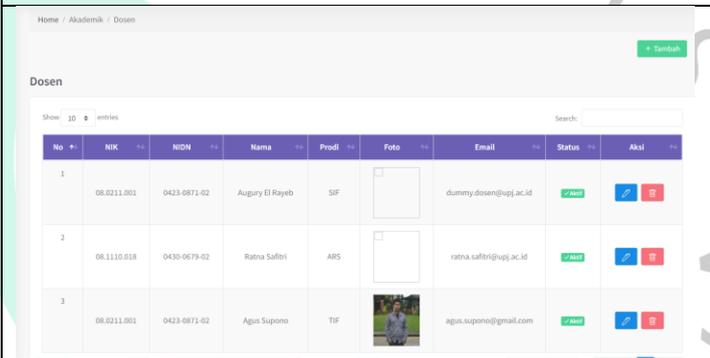
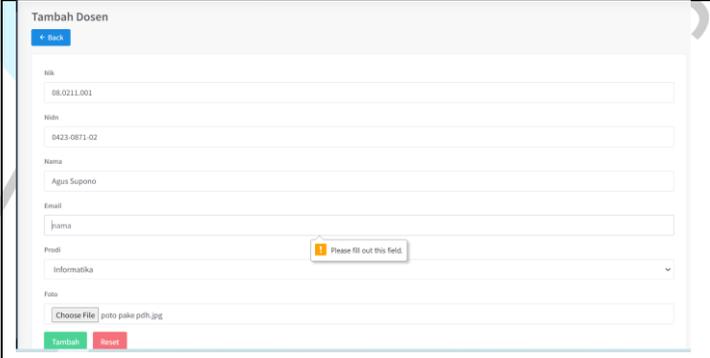
Black-box atau *behavioral testing* adalah pengujian yang dilakukan untuk mengamati hasil input dan output perangkat lunak tanpa mengetahui struktur kode perangkat lunak. Tes ini dijalankan pada akhir pembuatan perangkat lunak untuk memastikan bahwa perangkat lunak bekerja dengan benar.

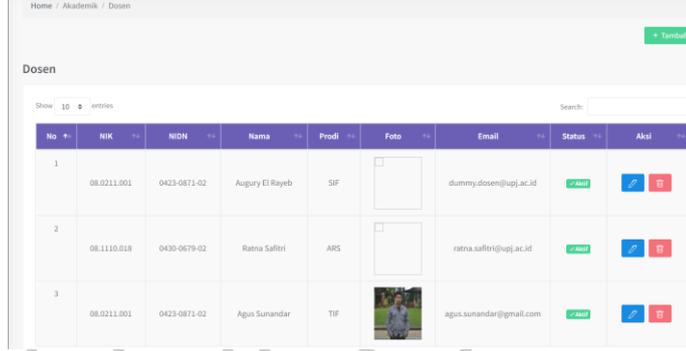
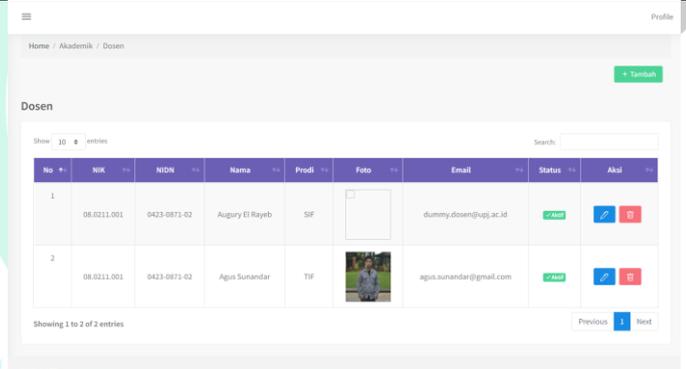
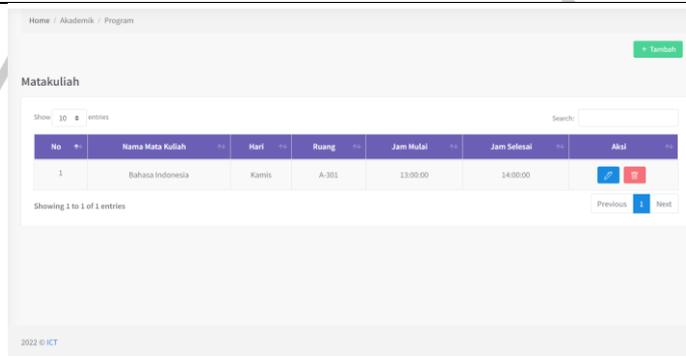
Berikut merupakan black box testing pada aplikasi sistem evaluasi akademik :

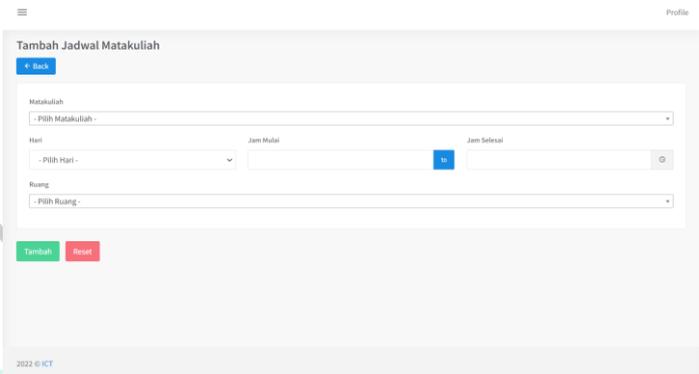
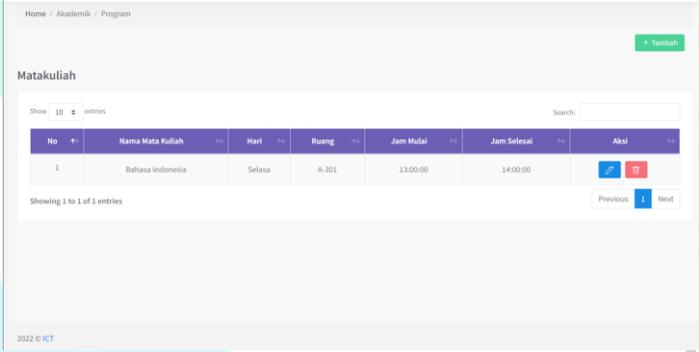
Tabel 3.2.2 1 Pengujian dengan Metode *Black Box*

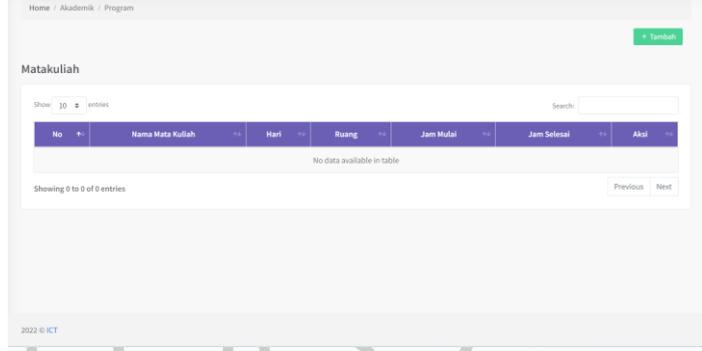
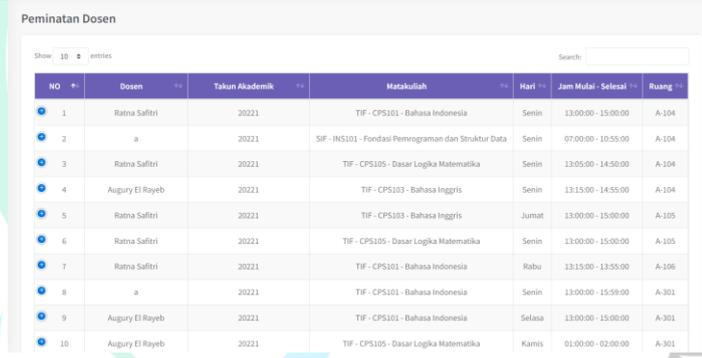
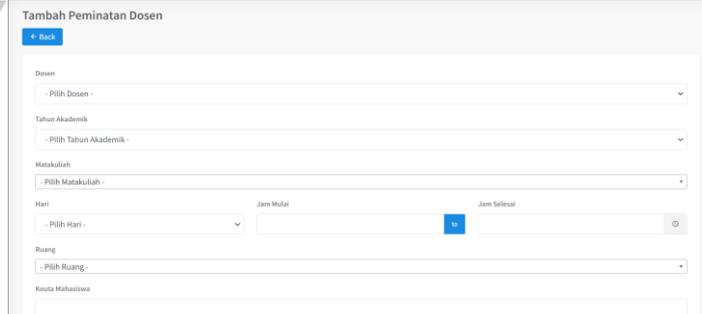
No	Skenario Pengujian	Hasil yang diharapkan
1	<i>Input</i> tahun akademik	Jika tahun id, nama, status di <i>input</i> maka data akan muncul di halaman utama tahun akademik.
		Hasil Pengujian

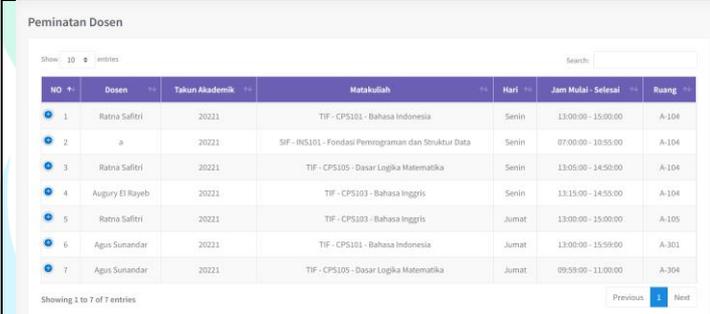
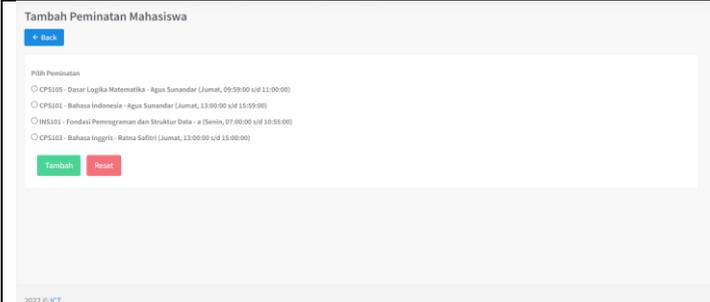
		 <p>Kesimpulan : data yang di <i>input</i> muncul di halaman utama</p>
2	<i>Input</i> tahun akademik	<p>Jika tahun id, nama, status salah satu tidak di <i>input</i> maka akan muncul peringatan untuk mengisi data.</p> <p>Hasil pengujian</p>  <p>Kesimpulan : peringatan berhasil muncul jika salah satu data tidak diisi.</p>
3	<i>Edit</i> tahun akademik	<p>Jika tahun id, nama, status di <i>edit</i> maka data akan berubah di halaman utama tahun akademik.</p> <p>Hasil pengujian</p>  <p>Kesimpulan : data yang di <i>edit</i> berubah di halaman utama tahun akademik</p>
4.	Hapus data tahun akademik	<p>Jika data di hapus maka data akan hilang dari halaman utama tahun akademik.</p> <p>Hasil Pengujian</p>

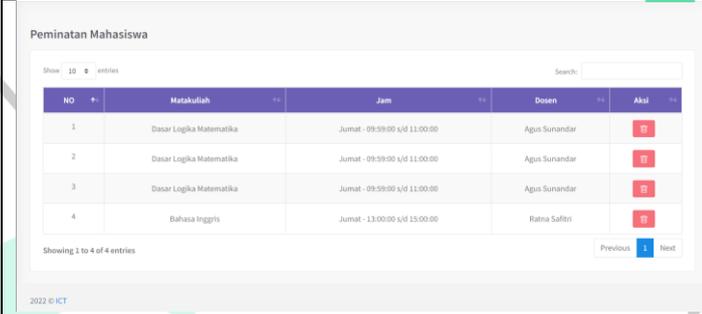
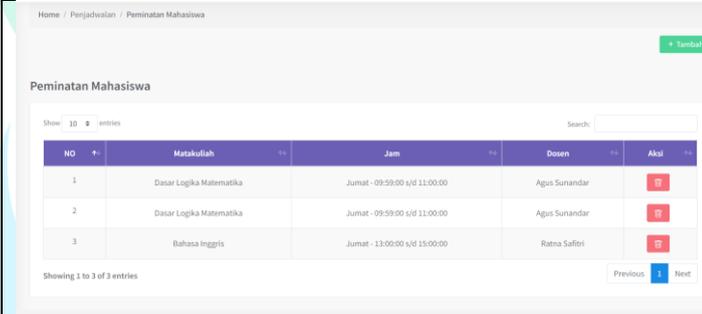
		 <p>Kesimpulan : data yang dihapus hilang dari halaman utama</p>
5	<i>Input dosen</i>	<p>Jika nik, nidn, nama, email, prodi, foto di <i>input</i> maka data akan muncul pada halaman utama dosen.</p> <p>Hasil Pengujian</p>  <p>Kesimpulan : data yang di <i>input</i> muncul di halaman utama</p>
6	<i>Input dosen</i>	<p>Jika nik, nidn, nama, email, prodi, foto salah satu tidak di input maka akan muncul peringatan untuk mengisi data.</p> <p>Hasil Pengujian</p>  <p>Kesimpulan : Peringatan berhasil muncul jika salah satu data tidak diisi.</p>
7	<i>Edit dosen</i>	<p>Jika nik, nidn, nama, email, prodi, foto di <i>edit</i> maka data akan berubah di halaman utama dosen.</p> <p>Hasil Pengujian</p>

		 <p>Kesimpulan : data yang di <i>edit</i> berubah di halaman utama dosen</p>
8	Hapus data dosen	<p>Jika data dihapus maka data akan hilang dari halaman utama dosen.</p> <p>Hasil Pengujian</p>  <p>Kesimpulan : data yang dihapus hilang dari halaman utama dosen.</p>
9	<i>Input</i> matakuliah	<p>Jika matakuliah, hari, jam mulai, jam selesai, ruang di <i>input</i> maka data akan muncul pada halaman utama matakuliah.</p> <p>Hasil Pengujian</p>  <p>Kesimpulan : data yang di input berhasil muncul dihalaman utama.</p>

10	<i>Input</i> matakuliah	Jika matakuliah, hari, jam mulai, jam selesai, ruang salah satu tidak di <i>input</i> maka halaman tidak berubah.
		<p>Hasil Pengujian</p>  <p>Kesimpulan : halaman tambah jadwal matakuliah tidak berubah</p>
11	<i>Edit</i> matakuliah	<p>Jika matakuliah, hari, jam mulai, jam selesai, ruang di <i>edit</i> maka data akan berubah pada halaman utama matakuliah.</p> <p>Hasil Pengujian</p>  <p>Kesimpulan : data yang di edit berubah di halaman utama matakuliah.</p>
12	Hapus matakuliah	<p>Jika data di hapus maka data akan hilang dari halaman utama matakuliah.</p> <p>Hasil Pengujian</p>

		 <p>Kesimpulan : data yang di hapus hilang dari halaman utama matakuliah.</p>
13	Input peminatan dosen	<p>Jika dosen, tahun akademik, matakuliah, hari, jam mulai, jam selesai, ruang, kuota mahasiswa di input maka data akan muncul pada halaman utama peminatan dosen.</p> <p>Hasil Pengujian</p>  <p>Kesimpulan : data yang di <i>input</i> muncul di halaman utama peminatan dosen</p>
14	Input peminatan dosen	<p>Jika dosen, tahun akademik, matakuliah, hari, jam mulai, jam selesai, ruang, kuota mahasiswa salah satu tidak di <i>input</i> maka halaman tambah peminatan dosen tidak berubah.</p> <p>Hasil Pengujian</p>  <p>Kesimpulan : halaman tambah peminatan dosen tidak</p>

		berubah
15	<i>Edit</i> peminatan dosen	<p>Jika dosen, tahun akademik, matakuliah, hari, jam mulai, jam selesai, ruang, kuota mahasiswa di <i>edit</i> maka data akan berubah pada halaman utama peminatan dosen.</p> <p>Hasil Pengujian</p>  <p>Kesimpulan : data yang di <i>edit</i> berubah di halaman utama peminatan dosen.</p>
16	Hapus peminatan dosen	<p>Jika data di hapus maka data akan hilang dari halaman utama peminatan dosen.</p> <p>Hasil Pengujian</p>  <p>Kesimpulan : data yang di hapus hilang dari halaman peminatan dosen.</p>
17.	Validasi peminatan dosen	<p>Jika data pada peminatan dosen sudah divalidasi maka akan muncul pada halaman tambah peminatan mahasiswa</p> <p>Hasil Pengujian</p> 

		Kesimpulan : data yang sudah divalidasi muncul di halaman tambah peminatan mahasiswa
18	<i>Input</i> peminatan mahasiswa	<p>Jika data peminatan dosen sudah valid kemudian di <i>input</i> dari halaman tambah peminatan mahasiswa maka data akan muncul pada halaman utama peminatan mahasiswa.</p> <p>Hasil Pengujian</p>  <p>Kesimpulan : data yang di input muncul di halaman utama peminatan mahasiswa.</p>
19	Hapus peminatan mahasiswa	<p>Jika data di hapus maka data akan hilang dari halaman utama peminatan mahasiswa</p> <p>Hasil Pengujian</p>  <p>Kesimpulan : data yang di hapus hilang dari halaman utama peminatan mahasiswa.</p>

3.3 Kendala Yang Dihadapi

Berikut adalah kendala yang dihadapi dalam membuat aplikasi berbasis website ini.

1. Framework yang digunakan untuk membuat aplikasi ini adalah codeigniter 3, sehingga praktikan perlu mempelajari framework tersebut. Kesulitan tersebut disebabkan pada saat perkuliahan praktikan mempelajari codeigniter 4, sehingga praktikan membutuhkan waktu untuk beradaptasi

dalam membuat aplikasi ini.

2. Pada menu penjadwalan terdapat kesulitan terkait waktu dan ruang yang diinput tidak boleh sama. Jadi apabila waktu dan ruang yang diinput itu sama maka data tidak akan terinput.

3.4 Cara Mengatasi Kendala

Berikut cara mengatasi kendala yang dihadapi dalam membuat aplikasi ini.

1. Mencari informasi mengenai crud pada codeigniter 3 di youtube dan google. Kemudian membaca dokumentasi codeignite 3 di google.
2. Mengatasi kendala pada menu penjadwalan dengan bertanya kepada mentor mengenai tahapan atau proses apa yang harus dipahami, kemudian mencari di google dan youtube.
3. Meningkatkan komunikasi dengan mentor dan rekan tim sehingga dapat saling membantu apabila terdapat kesulitan.

3.5 Pembelajaran yang Diperoleh dari Kerja profesi

Selama mengikuti kerja profesi di Universitas Pembangunan Jaya bagian departemen ICT praktikan mendapatkan beberapa pembelajaran yang berguna sebagai bekal untuk di dunia kerja nantinya. Berikut ini adalah pembelajaran yang diperoleh dari kerja profesi.

1. Tanggung jawab terhadap tugas yang diberikan oleh mentor.
2. Disiplin terhadap waktu yang diberikan, praktikan harus mengikuti *time line* yang diberikan mentor mengenai tugas yang harus dikerjakan.
3. Mencari solusi terhadap tugas yang diberikan, praktikan harus bisa mencari solusi apabila terdapat tugas yang dimengerti praktikan.
4. Komunikasi antar tim, praktikan harus bisa berkomunikasi dengan tim agar dapat menjalin hubungan dengan baik dan mencari solusi dari pekerjaan yang tidak dimengerti.
5. Praktikan harus bisa beradaptasi terhadap lingkungan tempat praktikan bekerja.