

Nama : Restu Putri Mulyatiningsih

NIM : 12030123120025

Kelas : C

Mata Kuliah : Analisis dan Desain Sistem

Deskripsi Kasus: Sistem Informasi Akademik Universitas

Sebuah universitas ingin mengembangkan sistem informasi akademik (SIAKAD) untuk memudahkan pengelolaan data akademik dan interaksi antara mahasiswa, dosen, serta administrasi akademik. Sistem ini akan digunakan untuk mengelola informasi terkait pendaftaran mahasiswa, pendaftaran mata kuliah, penginputan nilai oleh dosen, dan pelaporan hasil akademik.

- **Fitur Utama Sistem:**

1. Pendaftaran Mahasiswa Baru: Mahasiswa baru harus mengisi formulir pendaftaran dengan informasi pribadi mereka. Data tersebut akan disimpan di sistem dan diverifikasi oleh admin.
2. Pengelolaan Data Mahasiswa: Admin dapat menambah, mengubah, atau menghapus data mahasiswa.
3. Pendaftaran Mata Kuliah: Mahasiswa dapat mendaftar mata kuliah setiap semester dengan melihat jadwal yang tersedia.
4. Input Nilai oleh Dosen: Dosen dapat menginput nilai mahasiswa untuk mata kuliah yang diajarkan.
5. Melihat Nilai: Mahasiswa dapat melihat nilai yang diberikan dosen untuk mata kuliah yang diambil.
6. Pelaporan Akademik: Admin dapat menghasilkan laporan akademik seperti IPK mahasiswa, statistik kelulusan, dan lain-lain.

- **Penjelasan Proses:**

1. Pendaftaran Mahasiswa:

- o Mahasiswa baru mengisi form pendaftaran online yang mencakup data pribadi (nama, alamat, nomor telepon, email, program studi).
- o Admin memverifikasi data dan memberikan akses akun kepada mahasiswa.

2. Pendaftaran Mata Kuliah:

- Mahasiswa memilih mata kuliah yang tersedia setiap semester berdasarkan jadwal dan kapasitas.
- Sistem memvalidasi apakah jadwal mata kuliah tidak bentrok dan mahasiswa sudah memenuhi prasyarat mata kuliah.

3. Penginputan Nilai:

- Dosen menginput nilai mahasiswa melalui antarmuka sistem yang mengelola setiap mata kuliah yang diajarkan.
- Sistem secara otomatis menghitung IPK berdasarkan nilai mata kuliah yang diberikan.

4. Laporan Akademik:

- Admin dapat mengunduh berbagai laporan akademik seperti daftar mahasiswa yang lulus, statistik IPK, dan laporan mata kuliah terpopuler.

• **Pihak yang Terlibat:**

1. Mahasiswa: Mengakses sistem untuk mendaftar mata kuliah, melihat nilai, dan mengelola informasi pribadi.
2. Dosen: Menginput nilai untuk mahasiswa pada mata kuliah yang diampu.
3. Admin: Mengelola data mahasiswa, mata kuliah, jadwal, serta menghasilkan laporan akademik.

• **Skenario Penggunaan:**

Skenario 1: Pendaftaran Mata Kuliah oleh Mahasiswa

1. Mahasiswa login ke sistem menggunakan akun yang telah diberikan.
2. Mahasiswa memilih mata kuliah dari daftar yang tersedia.
3. Sistem memeriksa apakah jadwal mata kuliah tidak bentrok.
4. Jika valid, pendaftaran berhasil; jika tidak, sistem memberikan pesan error.
5. Mahasiswa mendapatkan konfirmasi pendaftaran yang mencakup jadwal kuliah.

Skenario 2: Penginputan Nilai oleh Dosen

1. Dosen login ke sistem dan memilih mata kuliah yang diajarkan.
2. Dosen melihat daftar mahasiswa yang terdaftar dalam mata kuliah tersebut.
3. Dosen menginput nilai untuk masing-masing mahasiswa.
4. Setelah selesai, sistem menyimpan nilai dan mahasiswa dapat melihatnya.

Skenario 3: Pembuatan Laporan oleh Admin

1. Admin login ke sistem dan memilih opsi untuk membuat laporan akademik.
2. Admin memilih parameter laporan (misalnya, laporan IPK rata-rata per program studi).
3. Sistem menghasilkan laporan dan menyajikannya dalam format PDF atau Excel.

- **Manfaat Sistem:**

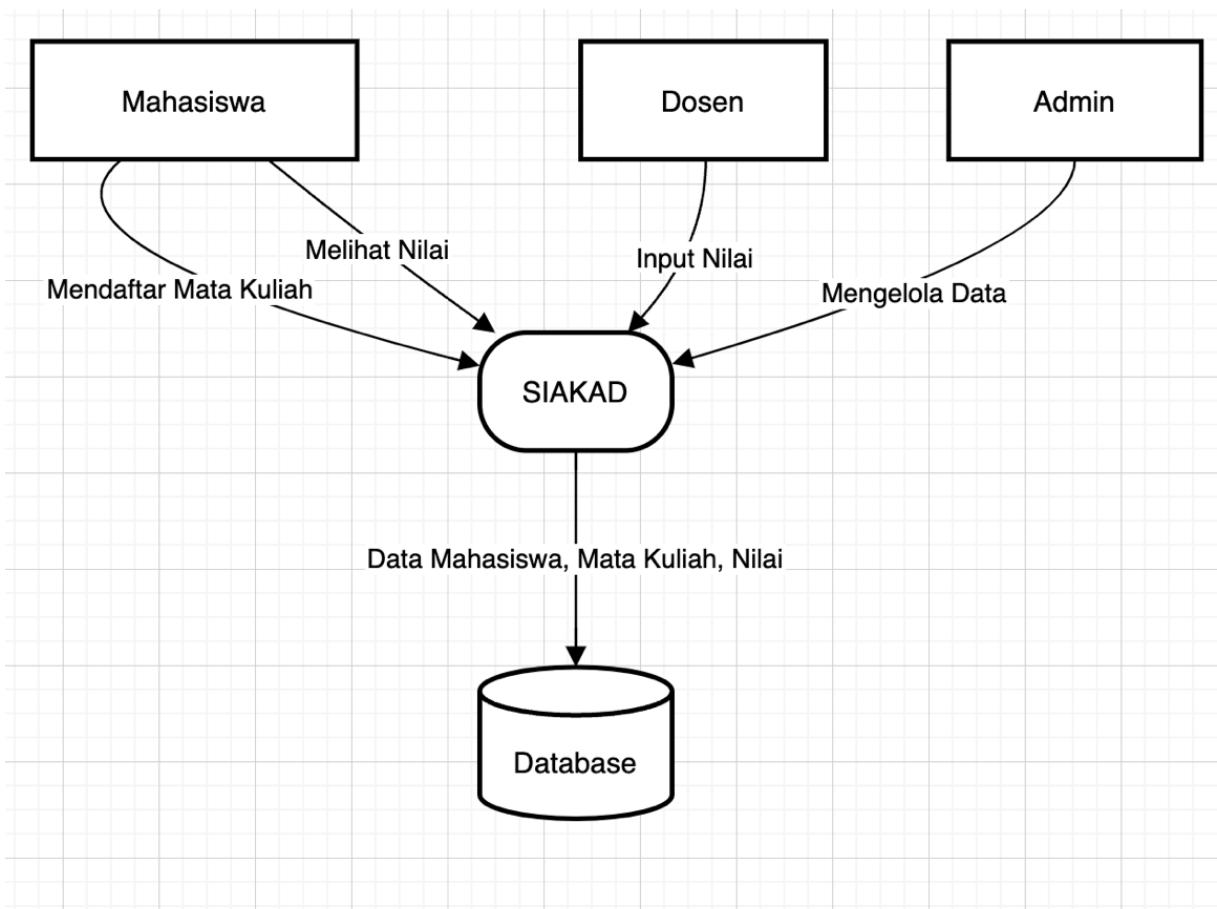
- Efisiensi: Proses pendaftaran dan pengelolaan data akademik menjadi lebih cepat dan terorganisir.
- Aksesibilitas: Mahasiswa, dosen, dan admin dapat mengakses sistem dari mana saja dan kapan saja.
- Akuntabilitas: Setiap perubahan data terjaga dalam sistem, sehingga dapat dilacak jika terjadi kesalahan.
- Transparansi: Mahasiswa dapat melihat nilai secara langsung dan memastikan keakuratan data akademik mereka.

Ini adalah contoh kasus sederhana dari sistem informasi akademik yang bisa diterapkan di sebuah universitas.

DFD Level 0 (Context Diagram)

Penjelasan DFD Level 0:

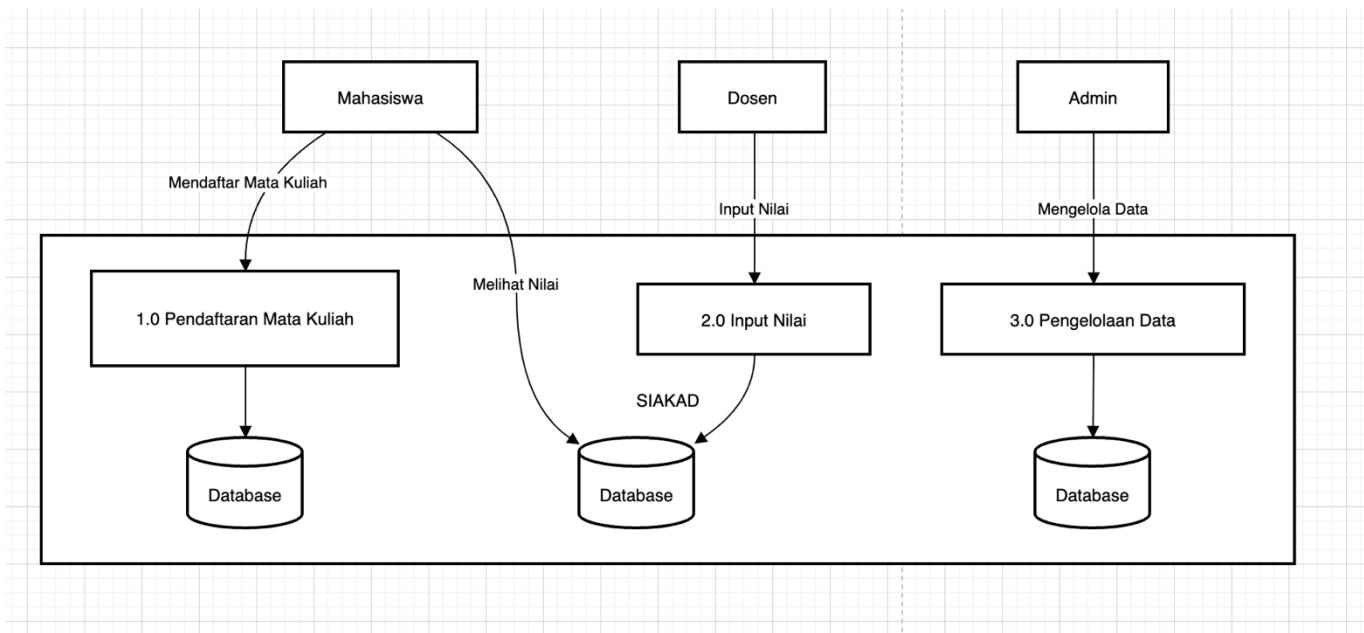
- Mahasiswa: Mendaftar mata kuliah dan melihat nilai.
- Dosen: Menginput nilai mahasiswa.
- Admin: Mengelola data di dalam sistem.
- SIAKAD: Sistem pusat yang berinteraksi dengan aktor eksternal dan menyimpan data dalam Database.



DFD Level 1 (Detail Proses Utama)

Penjelasan DFD Level 1:

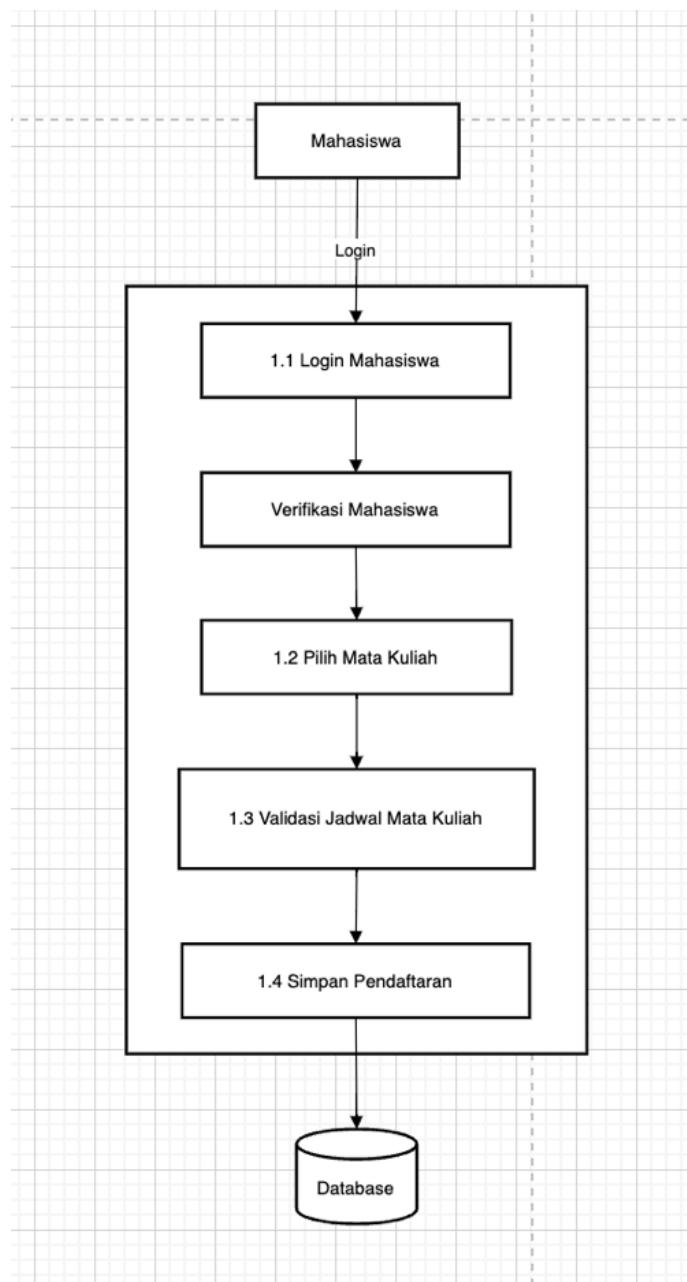
- Proses 1.0 Pendaftaran Mata Kuliah: Mahasiswa mendaftar mata kuliah ke sistem, data disimpan di database.
- Proses 2.0 Input Nilai: Dosen menginput nilai mahasiswa dan data nilai disimpan di database.
- Proses 3.0 Pengelolaan Data: Admin mengelola data mahasiswa, dosen, mata kuliah, dan nilai yang ada di sistem.



DFD Level 2: Detail Proses Pendaftaran Mata Kuliah

Penjelasan DFD Level 2 (Proses Pendaftaran Mata Kuliah):

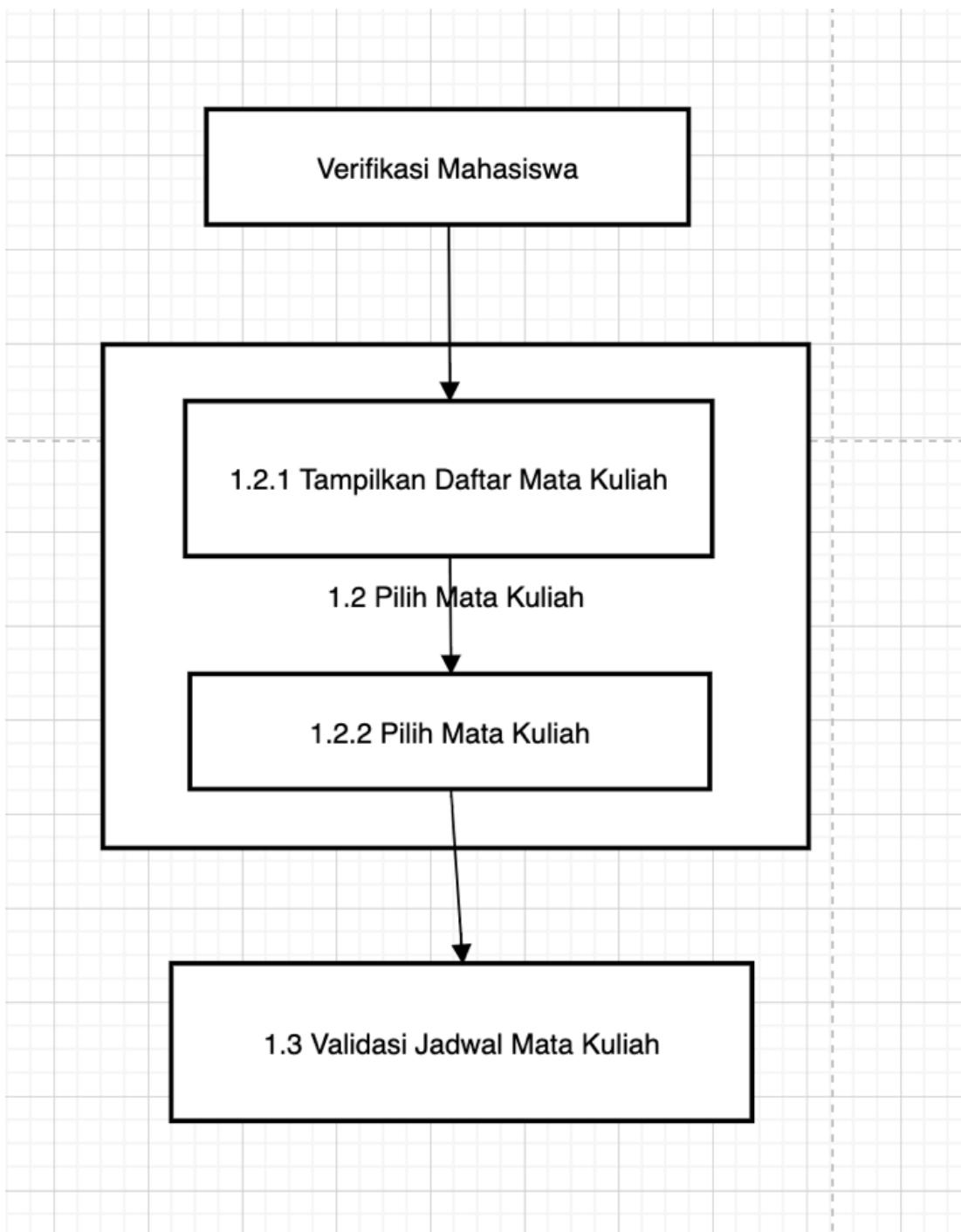
- Proses 1.1 Login Mahasiswa: Mahasiswa login ke sistem dan diverifikasi.
- Proses 1.2 Pilih Mata Kuliah: Mahasiswa memilih mata kuliah yang diinginkan.
- Proses 1.3 Validasi Jadwal Mata Kuliah: Sistem memvalidasi jadwal mata kuliah agar tidak bentrok.
- Proses 1.4 Simpan Pendaftaran: Jika validasi berhasil, data pendaftaran disimpan di database.



DFD Level 3: Detail Proses Pilih Mata Kuliah

Penjelasan DFD Level 3 (Proses Pilih Mata Kuliah):

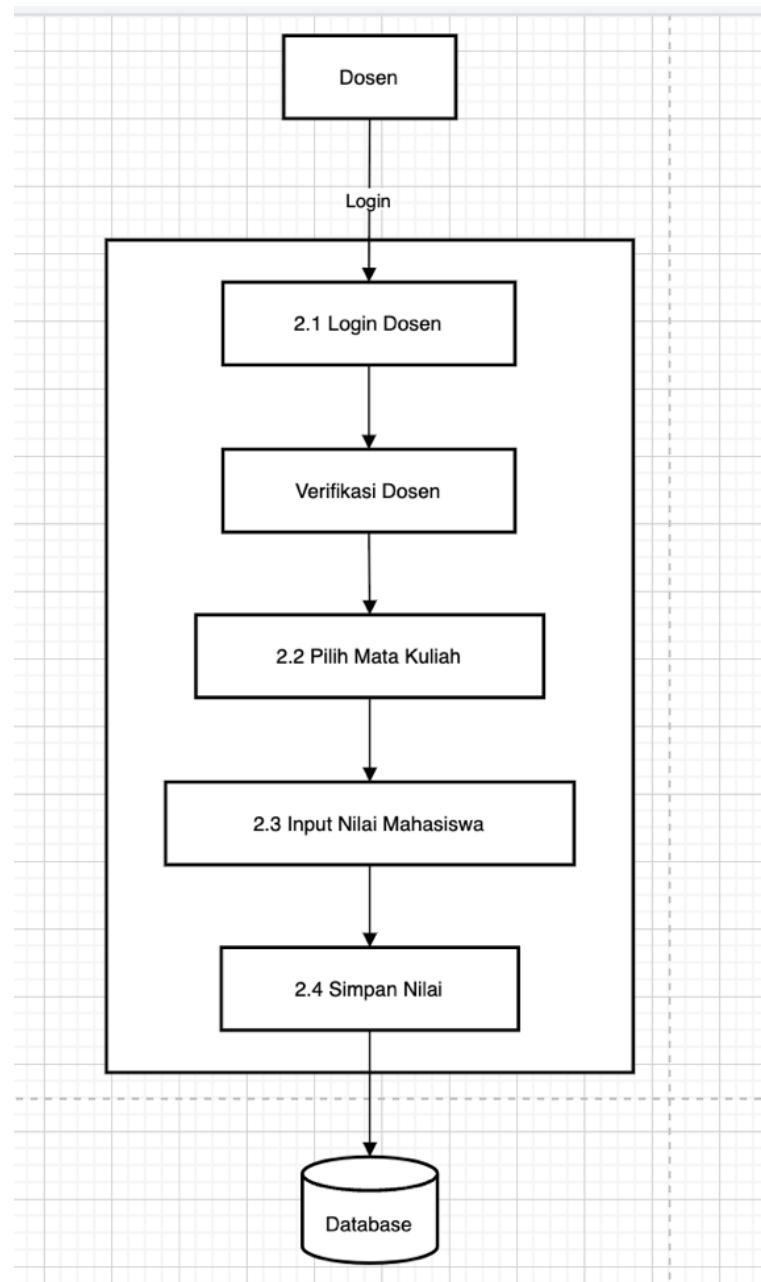
- Proses 1.2.1 Tampilkan Daftar Mata Kuliah: Sistem menampilkan daftar mata kuliah yang tersedia untuk dipilih mahasiswa.
- Proses 1.2.2 Pilih Mata Kuliah: Mahasiswa memilih mata kuliah dari daftar yang ditampilkan.



DFD Level 2: Detail Proses Input Nilai

Penjelasan DFD Level 2 (Proses Input Nilai):

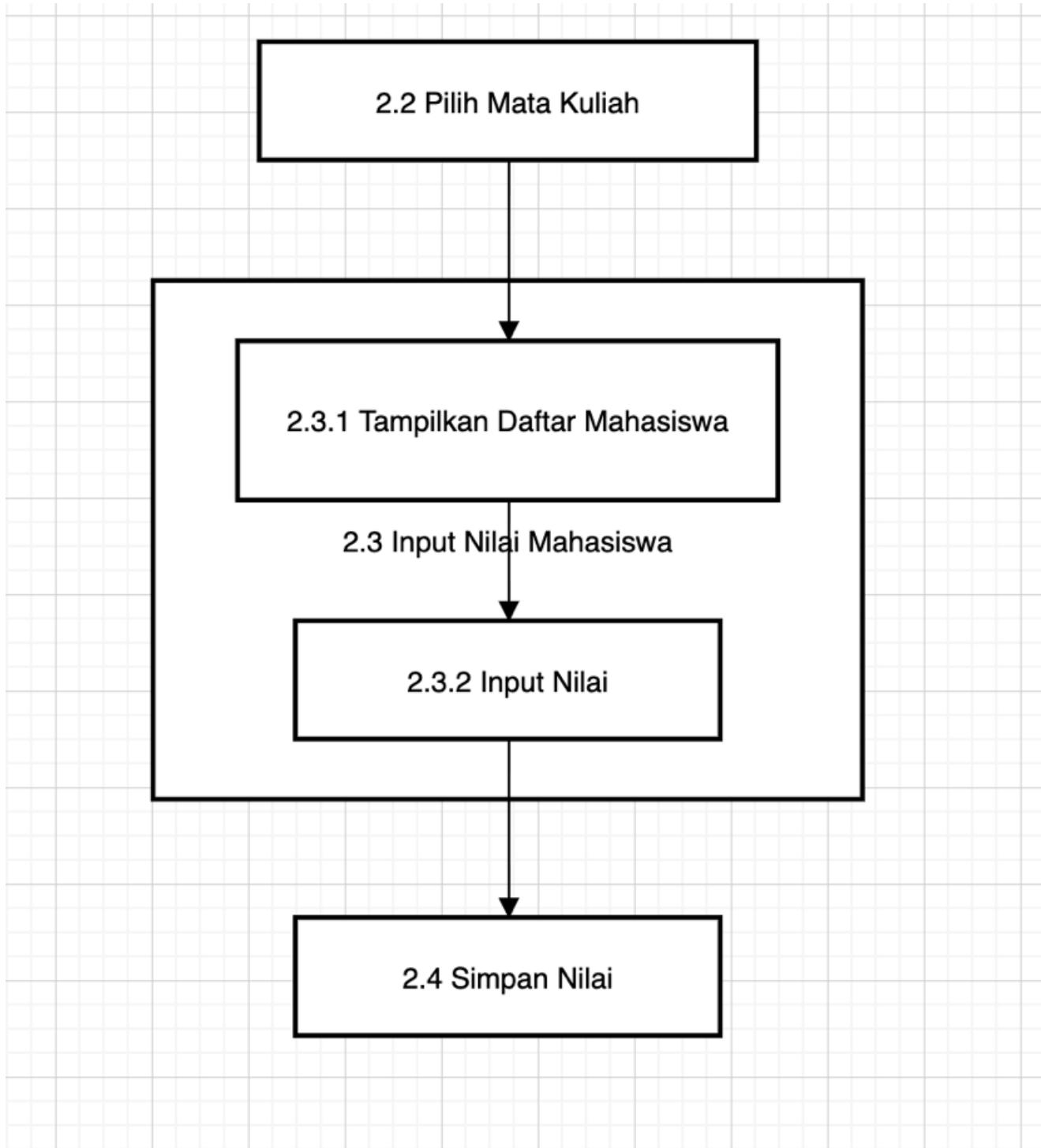
- Proses 2.1 Login Dosen: Dosen login ke sistem dan diverifikasi.
- Proses 2.2 Pilih Mata Kuliah: Dosen memilih mata kuliah yang diajarkan.
- Proses 2.3 Input Nilai Mahasiswa: Dosen memasukkan nilai untuk mahasiswa.
- Proses 2.4 Simpan Nilai: Data nilai disimpan ke dalam database.



DFD Level 3: Detail Proses Input Nilai Mahasiswa

Penjelasan DFD Level 3 (Proses Input Nilai Mahasiswa):

- Proses 2.3.1 Tampilkan Daftar Mahasiswa: Sistem menampilkan daftar mahasiswa yang terdaftar dalam mata kuliah.
- Proses 2.3.2 Input Nilai: Dosen memasukkan nilai untuk setiap mahasiswa dari daftar yang ditampilkan.



Entity-Relationship Diagram (ERD) SIAKAD

Penjelasan ERD:

1. Entitas:

- MAHASISWA: Menyimpan informasi mahasiswa, seperti ID, nama, email, password, jurusan, dan angkatan.
- DOSEN: Menyimpan informasi dosen, seperti ID, nama, email, password, dan departemen.
- MATA_KULIAH: Menyimpan informasi tentang mata kuliah, termasuk ID, nama mata kuliah, SKS, semester, dan jurusan.
- NILAI: Menyimpan nilai yang diperoleh oleh mahasiswa dalam mata kuliah, termasuk ID nilai, ID mahasiswa, ID mata kuliah, dan nilai.
- ADMIN: Menyimpan informasi tentang admin sistem, seperti ID, nama, email, dan password.

2. Relasi:

- MAHASISWA ||--o{ NILAI: Mahasiswa dapat memiliki banyak nilai untuk berbagai mata kuliah.
- DOSEN ||--o{ MATA_KULIAH: Dosen dapat mengajar banyak mata kuliah.
- MATA_KULIAH ||--o{ NILAI: Mata kuliah dapat memiliki banyak nilai yang diberikan kepada mahasiswa.
- ADMIN ||--o{ MAHASISWA: Admin dapat mengelola banyak mahasiswa.
- ADMIN ||--o{ MATA_KULIAH: Admin dapat mengelola banyak mata kuliah.
- ADMIN ||--o{ NILAI: Admin dapat mengelola banyak nilai.

Kesimpulan:

ERD ini memberikan gambaran menyeluruh tentang struktur data dalam Sistem Informasi Akademik (SIAKAD). Ini menunjukkan entitas utama yang terlibat dalam sistem serta hubungan antar entitas tersebut, yang dapat membantu dalam pengembangan dan implementasi sistem.

2.2 Pilih Mata Kuliah

2.3.1 Tampilkan Daftar Mahasiswa

2.3 Input Nilai Mahasiswa

2.3.2 Input Nilai

2.4 Simpan Nilai

DOSEN		
int	id_dosen	Primary Key
string	nama	
string	email	
string	password	
string	departemen	

ADMIN		
int	id_admin	Primary Key
string	nama	
string	email	
string	password	

MAHASISWA		
int	id_mahasiswa	Primary Key
string	nama	
string	email	
string	password	
string	jurusan	
string	angkatan	

MATA_KULIAH		
int	id_mata_kuliah	Primary Key
string	nama_mata_kuliah	
string	sks	
string	semester	
string	jurusan	

NILAI		
int	id_nilai	Primary Key
int	id_mahasiswa	Foreign Key
int	id_mata_kuliah	Foreign Key
float	nilai	

mengelola

mengajar

mengelola

mengelola

mendapatkan

memiliki

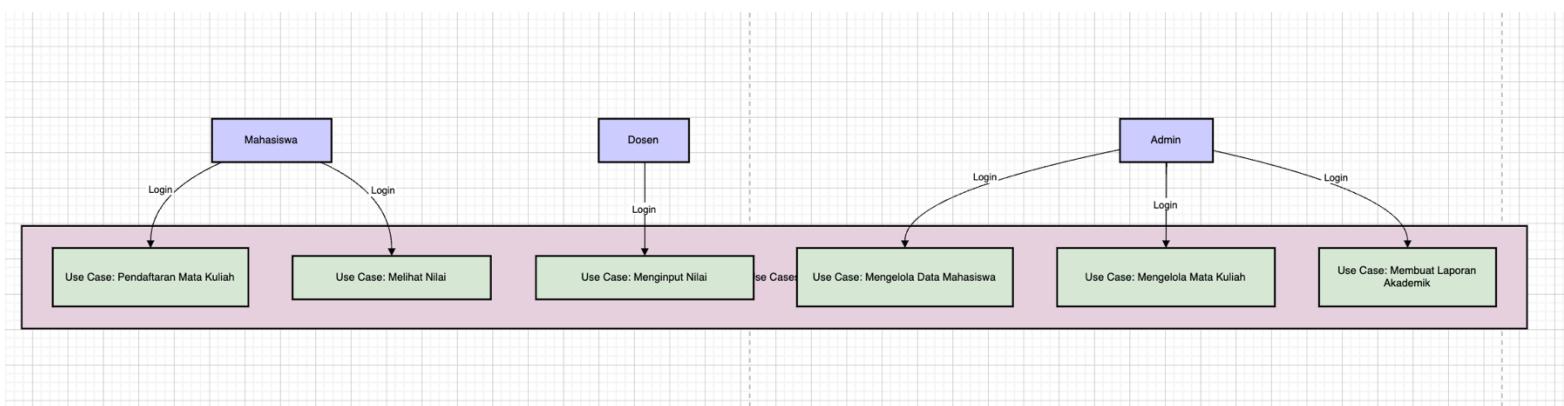
Sistem Informasi Akademik (SIAKAD) dalam bentuk berbagai diagram UML

1. Use Case Diagram

Penjelasan:

- Use Case Diagram menggambarkan interaksi antara aktor (pengguna sistem) dan fungsionalitas sistem (use case). Dalam diagram ini, terdapat tiga aktor: Mahasiswa, Dosen, dan Admin, yang masing-masing berinteraksi dengan sistem untuk melakukan berbagai aktivitas.
 - Mahasiswa: Aktor yang dapat melakukan pendaftaran mata kuliah dan melihat nilai.
 - Dosen: Bertugas untuk menginput nilai bagi mahasiswa pada mata kuliah yang diajarkan.
 - Admin: Mengelola data akademik, termasuk mengelola data mahasiswa, mengelola mata kuliah, serta membuat laporan akademik seperti IPK rata-rata atau statistik kelulusan.

Diagram ini memberikan gambaran umum tentang fungsionalitas yang diakses oleh masing-masing aktor dalam sistem SIAKAD.

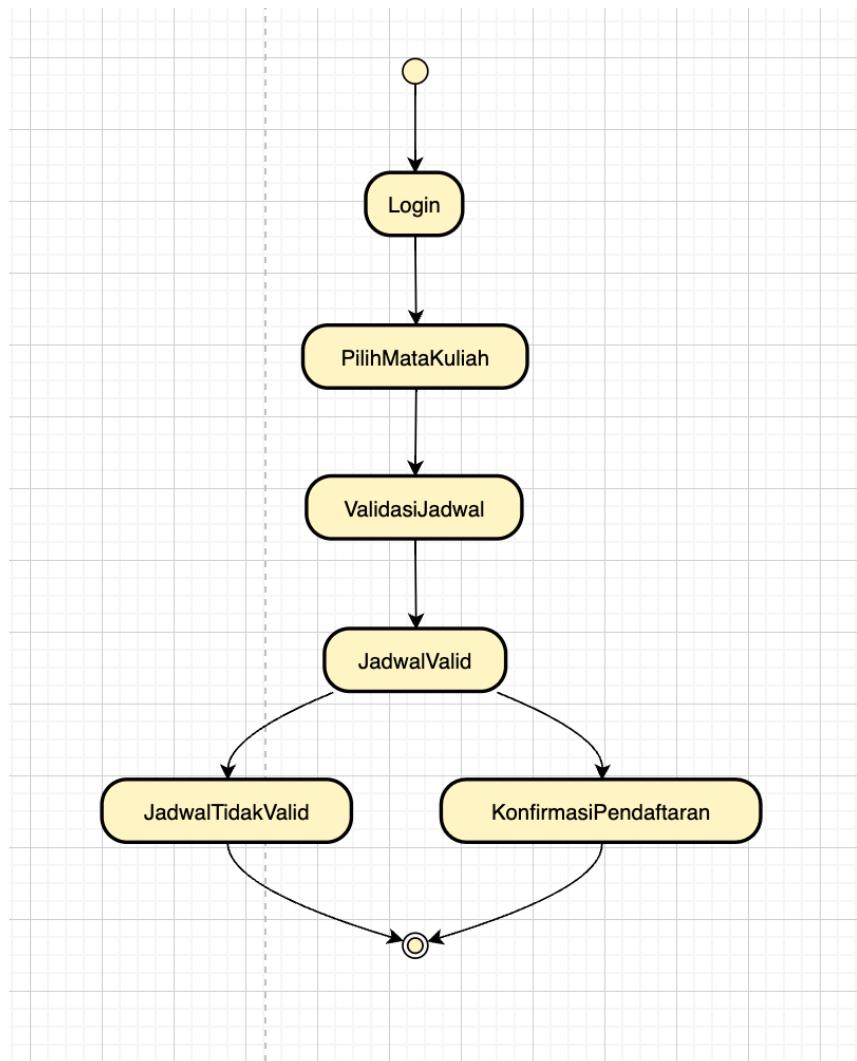


2. Activity Diagram: Pendaftaran Mata Kuliah oleh Mahasiswa

Penjelasan:

- Activity Diagram menjelaskan alur aktivitas yang dilakukan mahasiswa saat melakukan pendaftaran mata kuliah.
 1. Mahasiswa login ke dalam sistem.
 2. Mahasiswa memilih mata kuliah dari daftar yang tersedia (PilihMataKuliah).
 3. Sistem akan melakukan validasi jadwal untuk memastikan bahwa mata kuliah yang dipilih tidak bentrok dengan jadwal lainnya.
 4. Jika jadwal valid, sistem akan mengirimkan konfirmasi pendaftaran.
 5. Jika tidak valid, mahasiswa menerima pesan error dan tidak dapat melanjutkan pendaftaran.

Diagram ini memperlihatkan proses yang dijalani mahasiswa dari awal hingga selesai melakukan pendaftaran mata kuliah.

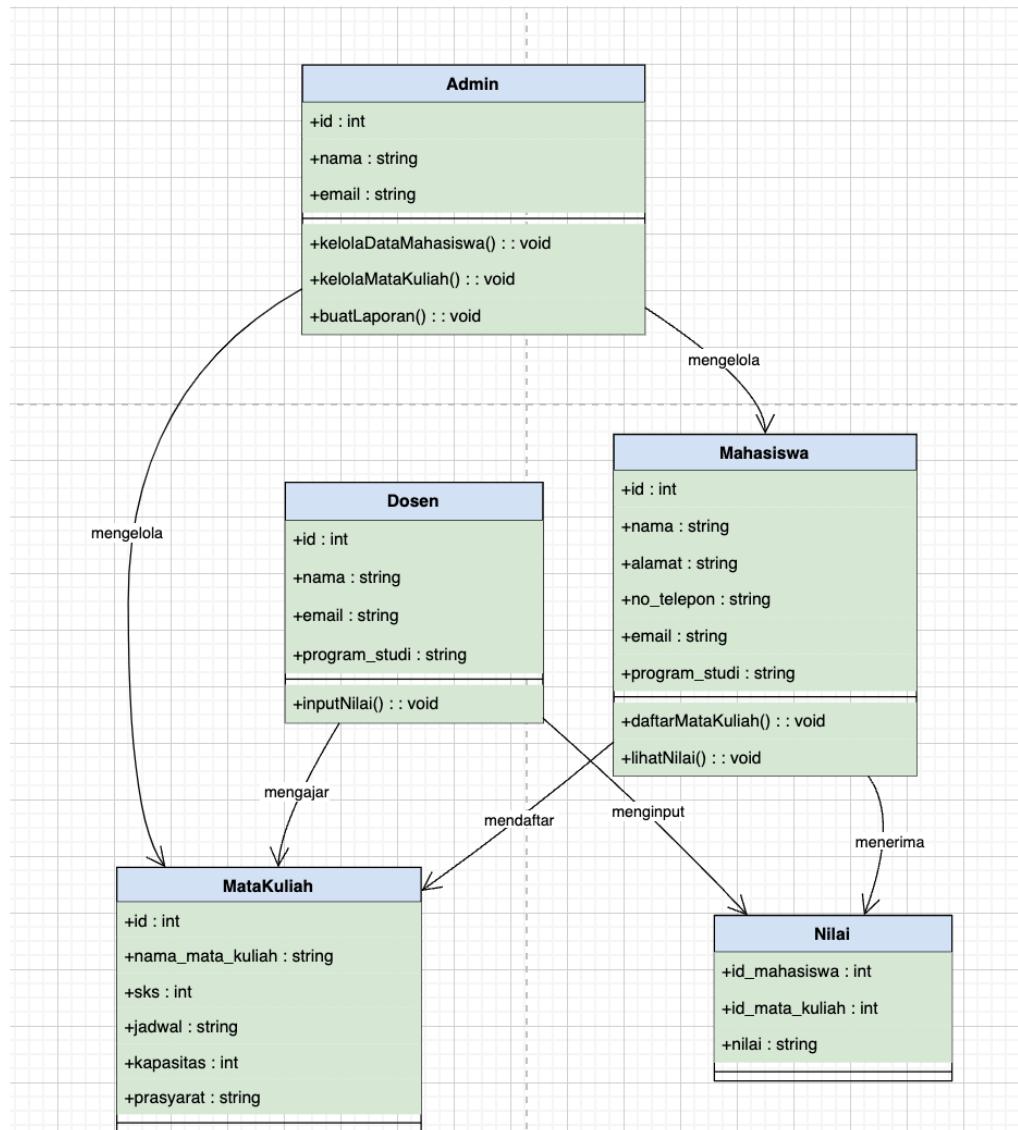


3. Class Diagram

Penjelasan:

- Class Diagram menggambarkan struktur statis dari sistem, termasuk kelas-kelas yang terlibat dan hubungannya.
 - Mahasiswa: Mewakili data dan aktivitas yang bisa dilakukan oleh mahasiswa, seperti mendaftar mata kuliah dan melihat nilai.
 - Dosen: Mewakili dosen yang menginput nilai mahasiswa.
 - Admin: Mengelola data mahasiswa, mata kuliah, dan membuat laporan akademik.
 - MataKuliah: Kelas ini berisi data mata kuliah seperti nama, SKS, jadwal, kapasitas, dan prasyarat.
 - Nilai: Berfungsi untuk menyimpan informasi nilai yang diberikan oleh dosen kepada mahasiswa.

Kelas-kelas ini saling berhubungan, misalnya, Mahasiswa mendaftar pada MataKuliah, Dosen menginput Nilai, dan Admin mengelola semua informasi.

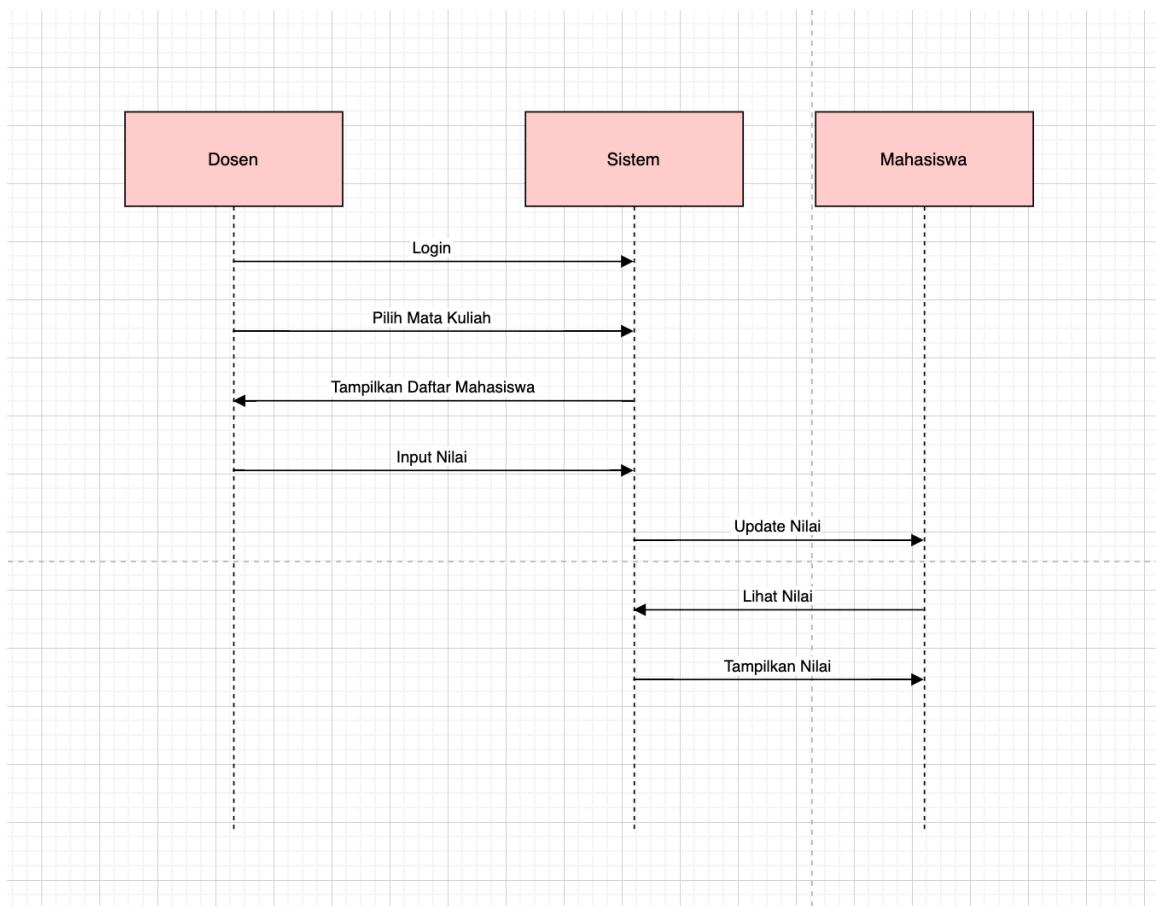


4. Sequence Diagram: Penginputan Nilai oleh Dosen

Penjelasan:

- Sequence Diagram menggambarkan urutan interaksi antar objek dalam suatu skenario waktu. Dalam hal ini, alur penginputan nilai oleh dosen.
1. Dosen login ke sistem.
 2. Dosen memilih mata kuliah yang diajarkan.
 3. Sistem menampilkan daftar mahasiswa yang terdaftar dalam mata kuliah tersebut.
 4. Dosen menginput nilai masing-masing mahasiswa.
 5. Nilai yang diinput kemudian disimpan di sistem, dan mahasiswa bisa melihat nilai yang telah diinput oleh dosen.

Diagram ini menjelaskan proses penginputan nilai dari perspektif waktu dan interaksi antara aktor dan sistem.

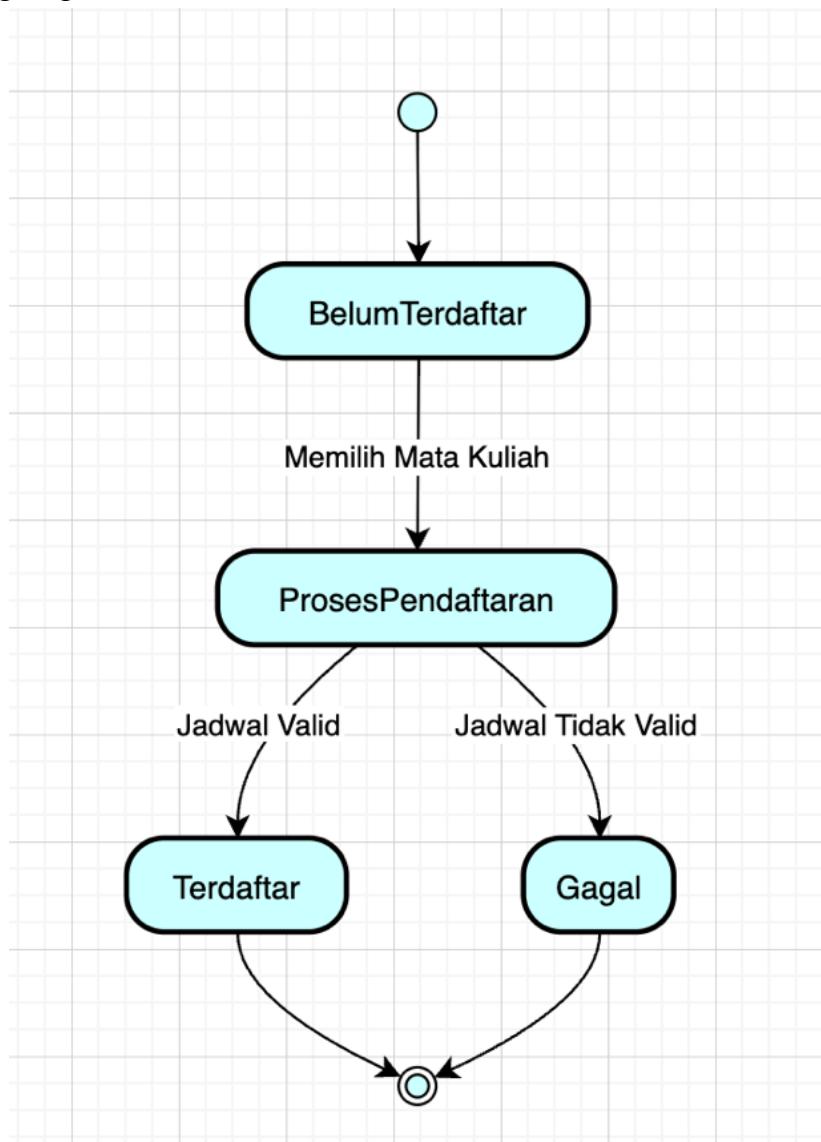


5. State Diagram: Status Pendaftaran Mata Kuliah

Penjelasan:

- State Diagram menjelaskan perubahan status sistem, khususnya terkait dengan proses pendaftaran mata kuliah.
 1. Awalnya, mahasiswa berada dalam status Belum Terdaftar.
 2. Saat mahasiswa memilih mata kuliah, sistem masuk ke dalam status Proses Pendaftaran.
 3. Jika jadwal valid dan mahasiswa memenuhi syarat, pendaftaran berhasil dan status berubah menjadi Terdaftar.
 4. Jika jadwal tidak valid atau syarat tidak terpenuhi, pendaftaran gagal dan sistem kembali ke status Belum Terdaftar.

Diagram ini menggambarkan perubahan status dalam sistem selama proses pendaftaran mata kuliah berlangsung.



Secara keseluruhan, diagram-diagram UML ini memberikan representasi visual dari bagaimana Sistem Informasi Akademik (SIAKAD) berfungsi, baik dari sisi fungsionalitas (Use Case Diagram), alur proses (Activity Diagram), hubungan antar entitas (Class Diagram), urutan interaksi (Sequence Diagram), hingga perubahan status sistem (State Diagram).