**1. Deskripsi Umum Perangkat Lunak**

**1.1. Latar Belakang**

Pemindahan jurusan merupakan salah satu kebutuhan akademik yang sering terjadi di lingkungan perguruan tinggi, terutama ketika mahasiswa merasa bahwa program studi yang dijalaninya saat ini tidak sesuai dengan minat atau bakatnya. Di Program Studi Informatika, proses pengajuan pemindahan jurusan sebelumnya dilakukan dengan metode tradisional yang melibatkan pengisian berkas fisik dan pengumpulan dokumen secara manual. Prosedur ini sering kali memerlukan koordinasi dengan banyak pihak, seperti pihak akademik, dosen wali, dan administrasi, sehingga memakan waktu yang cukup lama dan rawan kesalahan. Selain itu, penyimpanan dokumen fisik yang terpisah-pisah meningkatkan resiko kehilangan atau kerusakan data, serta menghambat kelancaran pelacakan proses oleh mahasiswa maupun pihak administrasi. Proses manual ini juga kurang efisien dalam hal sumber daya manusia dan biaya, serta berpotensi menyebabkan ketidaktransparanan dalam pengelolaan permohonan. Melihat permasalahan tersebut, diperlukan sebuah sistem berbasis digital yang mampu menyederhanakan dan mempercepat seluruh proses. Dengan memanfaatkan teknologi website, diharapkan sistem ini dapat memberikan solusi yang lebih terstruktur dan terdokumentasi dengan baik, yang pada akhirnya dapat membantu mahasiswa dalam mewujudkan keinginan pindah jurusan dengan lebih cepat dan akurat. Sistem ini juga bertujuan untuk mendukung kelestarian lingkungan dengan mengurangi penggunaan kertas (*paperless*) sejalan dengan program kampus ramah lingkungan.

Secara keseluruhan, sistem ini diharapkan mampu meningkatkan efisiensi pelayanan, memberikan transparansi proses bagi mahasiswa, mengamankan data permohonan, serta meningkatkan aksesibilitas dengan fleksibilitas tinggi.

**1.2. SDLC (*Software Development Life Cycle*)**

Metode pengembangan yang di ambil yaitu *Waterfall*. Model *waterfall* (model air terjun) adalah salah satu metode pengembangan perangkat lunak yang linear dan berurutan. Dalam model ini, setiap fase dalam siklus pengembangan perangkat lunak dijalankan secara berurutan, dan masing-masing fase bergantung pada selesai atau diselesaikannya fase sebelumnya. Keuntungan dari model waterfall melibatkan pendekatan yang terstruktur dan mudah dimengerti. Setiap fase memiliki hasil yang jelas dan terdokumentasi dengan baik. Namun, model ini memiliki kelemahan, terutama ketidakmampuan untuk menangani perubahan kebutuhan pengguna setelah proyek dimulai. Jika terdapat perubahan kebutuhan di tahap lanjut, dapat sulit dan mahal untuk mengimplementasikannya.

**1.3. Paradigma Program**

Paradigma Paradigma pemrograman mengacu pada kerangka kerja konseptual atau cara berpikir yang digunakan dalam pengembangan perangkat lunak. Ini mencakup aturan, konsep, dan praktik-praktik yang membentuk dasar untuk merancang program computer, paradigma yang digunakan dalam program kami yaitu paradigma OOP. Paradigma berorientasi objek (*Object-Oriented Programming* atau OOP) adalah suatu pendekatan dalam pengembangan perangkat lunak yang didasarkan pada konsep "objek". Konsep ini mencakup kombinasi data dan metode yang beroperasi pada data tersebut.

**2.** **Deskripsi Global Perangkat Lunak**

**2.1. *Statement of Objective* Perangkat Lunak**

Sistem Pengajuan untuk Pemindahan Jurusan di Informatika berbasis *Website* menyediakan proses pengajuan, pengelolaan, dan status pemindahan di Program Studi Informatika, sehingga proses dapat berlangsung dengan lebih cepat, efisien, dan transparan, yang dapat digunakan oleh mahasiswa.

**2.2. Prespektif dan Fungsi Perangkat Lunak**

Terdapat beberapa fungsi yang ditawarkan oleh aplikasi ini, diantaranya adalah:

1. Pengajuan Permohonan Pindah Jurusan

Mahasiswa dapat mengajukan permohonan pindah jurusan secara *online*. Fitur ini memungkinkan mahasiswa untuk mengisi formulir permohonan pindah jurusan dan mengunggah dokumen-dokumen pendukung yang dibutuhkan.

1. Pengecekan Status Pengajuan

Mahasiswa bisa melihat status pengajuan mereka secara langsung melalui sistem. Ini memungkinkan mereka mengetahui apakah permohonannya sedang dalam proses verifikasi, evaluasi, atau sudah mendapatkan keputusan.

1. Persetujuan oleh Pihak Terkait

Mahasiswa bisa melihat status pengajuan mereka secara langsung melalui sistem. Ini memungkinkan mereka mengetahui apakah permohonannya sedang dalam proses verifikasi, evaluasi, atau sudah mendapatkan keputusan sekaligus Admin akan memberikan notifikasi terkait berkas yang diberikan ditolak/diterima.

**2.3. Profil dan Karakteristik Pengguna**

Pengguna utama sistem adalah mahasiswa yang ingin mengajukan permohonan pindah jurusan di Program Studi Informatika. Mahasiswa ini mungkin sedang menjalani semester awal atau lanjutan, dan memiliki berbagai alasan untuk mengajukan pindah jurusan, seperti ketidaksesuaian minat, kesulitan dalam mengikuti program studi, atau alasan lainnya.

Tabel 2.1 Profil dan Karakteristik Pengguna

|  |  |  |
| --- | --- | --- |
| **No** | **Karakteristik Pengguna** | **Aktivitas** |
| 1. | Admin | Pihak yang bertugas mengelola sistem dan memverifikasi permohonan pindah jurusan |
| 2. | Mahasiswa | Pihak yang ingin melakukan pengajuan pindah jurusan |

**2.4. Lingkungan Operasi**

Lingkungan kami dapat berfungsi pada spesifikasi berikut:

1. Windows
2. Web Server

**2.5. Batasan Perangkat Lunak / Sistem**

Adapun bebarapa batasan perangkat lunak dari Sistem Pengajuan untuk Pemindahan Jurusan di Informatika dianataranya:

1. Sistem ini hanya mendukung proses pengajuan dan pengelolaan permohonan pindah jurusan di Program Studi Informatika.
2. Harus terhubung dengan internet
3. Fitur pelacakan status hanya terbatas pada tahap-tahap standar, seperti "Pengajuan diterima", "Dalam proses verifikasi", "Disetujui", atau "Ditolak". Sistem tidak memberikan informasi detail mengenai alasan di balik setiap tahapan atau komentar dari pihak yang memverifikasi.

**2.6. Asumsi dan Dependensi**

Berikut ini ada beberapa manfaat atau tugas dari Asumsi:

1. Sistem ini dibuat untuk memudahkan bagi Mahasiswa yang ingin pindah jurusan.
2. Akses Internet yang stabil.
3. Pengguna sistem tahu cara menggunakan perangkat komputer atau ponsel untuk mengakses *website*.

Berikut ini ada beberapa manfaat atau tugas dari Dependensi:

1. Sistem memerlukan koneksi internet agar bisa diakses.
2. Sistem hanya bisa digunakan oleh akun yang telah terdaftar.

**3. Deskripsi Perangkat Lunak**

**3.1. Deskripsi Kebutuhan**

Berikut ini beberapa kebutuhan dan rincian perangkat lunak dari kebutuhan fungsional dan non fungsional dari website yang dibuat:

**3.1.1. Kebutuhan Fungsional**

Tabel 3.1 Kebutuhan Fungsional

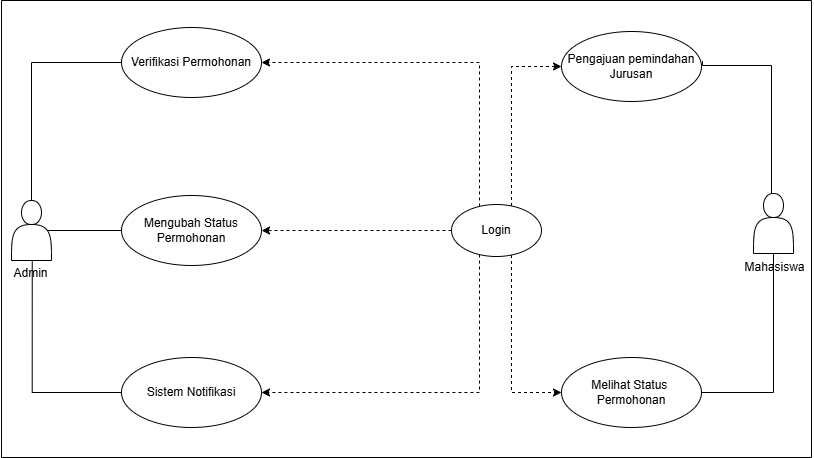
|  |  |  |
| --- | --- | --- |
| **Kode** | **Deskripsi** | **Sumber** |
| **F1** | **Pengajuan Permohonan Pindah Jurusan:**  Mahasiswa dapat mengajukan permohonan pemindahan jurusan dengan mengisi formulir dan mengunggah dokumen pendukung. | Mahasiswa |
| **F2** | **Persetujuan oleh Admin**  Admin dapat melihat dan memverifikasi permohonan yang diajukan oleh mahasiswa dan Admin akan memberikan notifikasi terkait berkasi yang diberikan | Admin |
| **F3** | **Pengecekan Status Pengajuan:**  Mahasiswa dapat melihat status permohonan. | Mahasiswa |

**3.1.2. Kebutuhan Non Fungsional**

Tabel 3.2 Kebutuhan Non Fungsional

|  |  |  |
| --- | --- | --- |
| **Kode** | **Quality** | **Deskripsi** |
| **N1** | *Ergonomy*. | Aplikasi ini menampilkan desain yang menarik agar peminjam dapat dengan nyaman menggunakan aplikasi ini. |
| **N2** | *Avaliability* | Aplikasi ini memiliki *availability* 24 jam |

**3.2. Pemodelan Analisis**  
**3.2.1. *Use Case* Diagram**

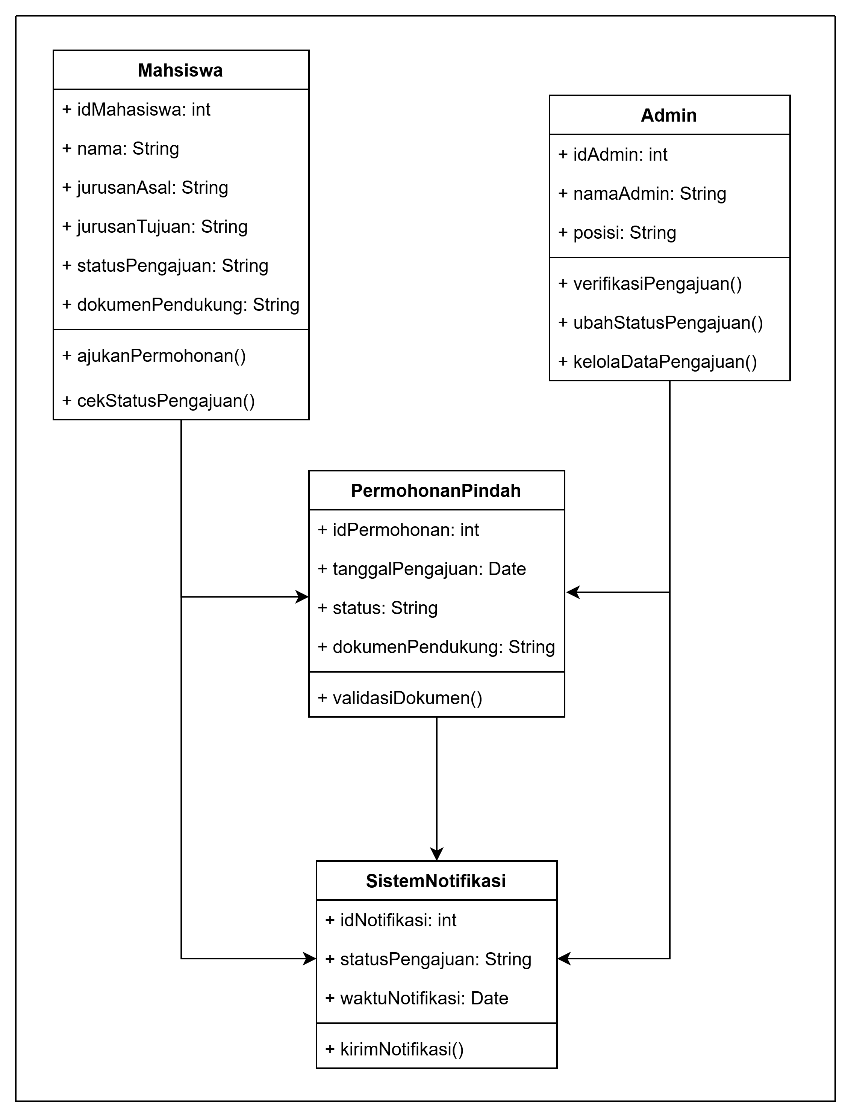
****

Gambar 3.1 *Use Case* Diagram

**Penjelasan:**

Untuk *Use Case* diatas dijelaskan bahwa Mahasiswa dapat mengajukan permohonan dengan mengisi formulir dan mengunggah dokumen, serta melihat status permohonan tersebut. Sedangkan Admin memiliki tugas memverifikasi permohonan mahasiswa dan mengubah statusnya menjadi sebagaimana semestinya setelah verifikasi. Dilanjutkan Sistem Notifikasi kemudian akan memberikan pemberitahuan kepada mahasiswa jika ada perubahan status permohonan.

**3.2.2. *Class* Diagram**

******

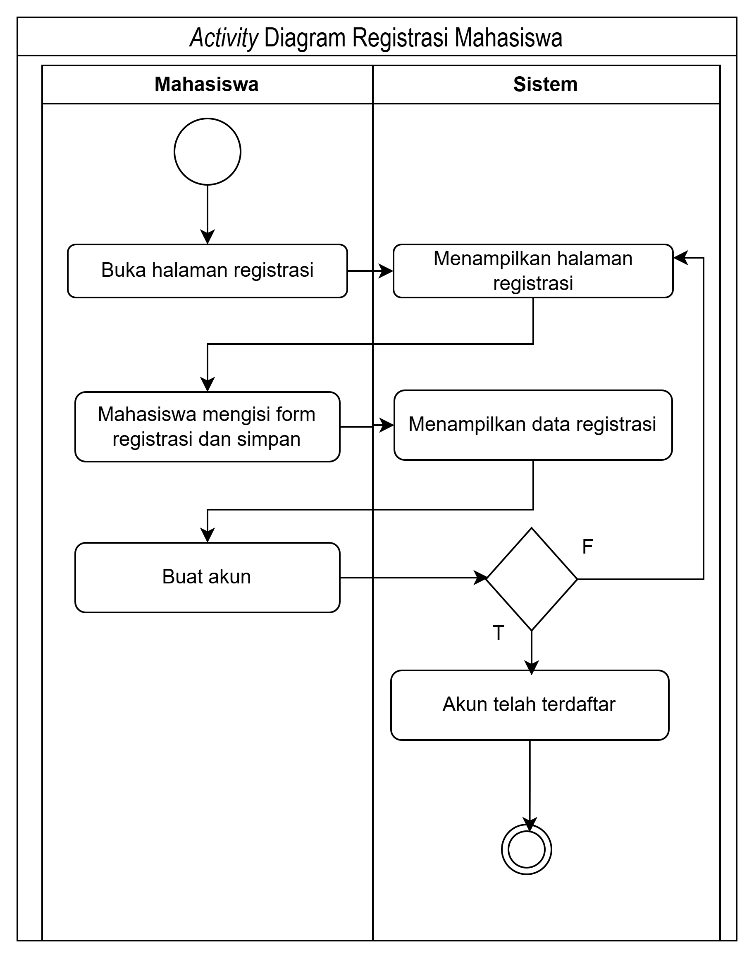
Gambar 3.2 *Class* Diagram

**Penjelasan:**

Struktur diatas menjelaskan bahwa memiliki representasi visual dari *class-class* dan hubungan satu sama lain, class diagram membantu dalam memahami struktur suatu system dan bagaimana *class-class* membantu dan berinteraksi satu sama lain.

**3.2.3. *Activity* Diagram**

**1. Registrasi Mahasiswa**

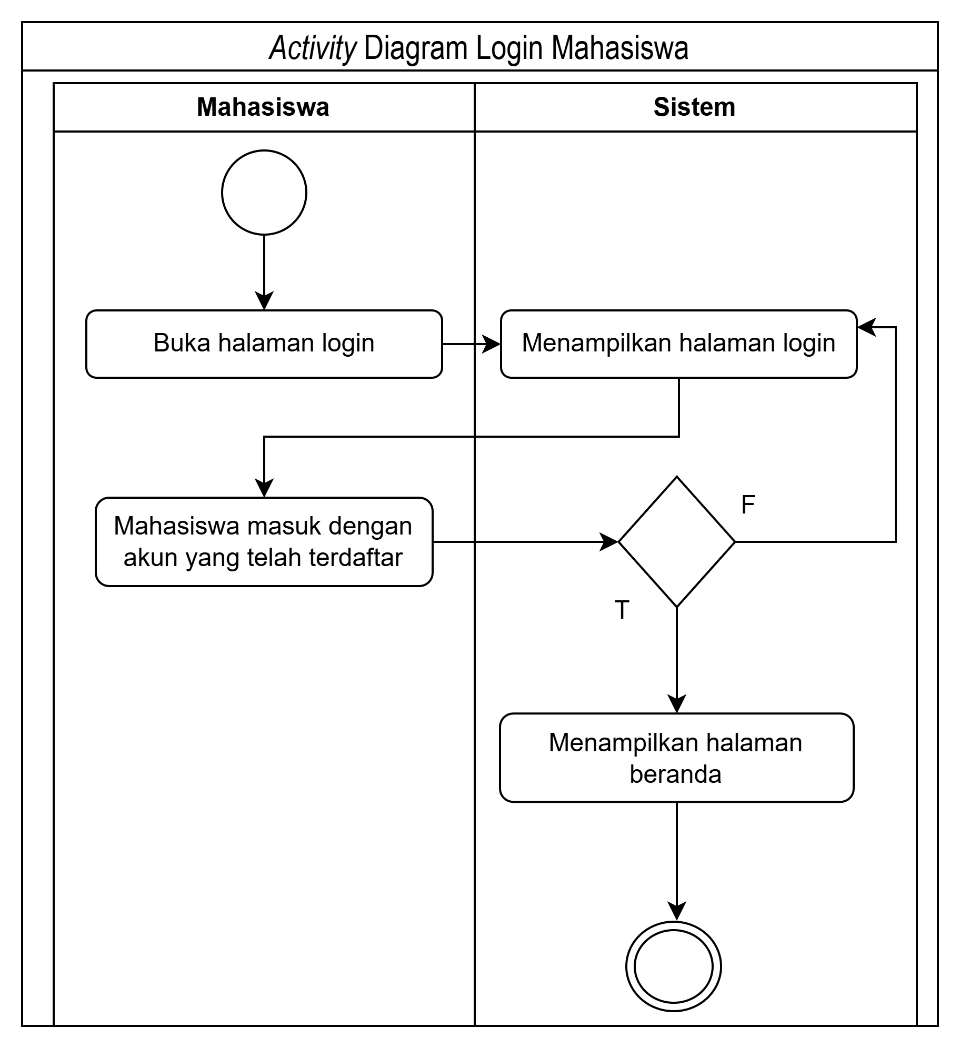
****

Gambar 3.3 *Activity* DiagramRegistrasi Mahasiswa

**Penjelasan:**

Diagram di atas menggambarkan alur registrasi akun mahasiswa di dalam sistem. Mahasiswa pertama-tama membuka halaman registrasi, yang kemudian ditampilkan oleh sistem. Setelah itu, mahasiswa mengisi form registrasi dan menyimpannya. Sistem menampilkan kembali data registrasi yang diisi untuk memastikan kebenaran data. Selanjutnya, sistem melakukan pengecekan, jika data tidak sesuai, alur kembali ke tampilan *form* registrasi, jika sesuai, sistem membuat *username* dan *password*. Terakhir, sistem mengonfirmasi bahwa akun mahasiswa telah berhasil terdaftar.

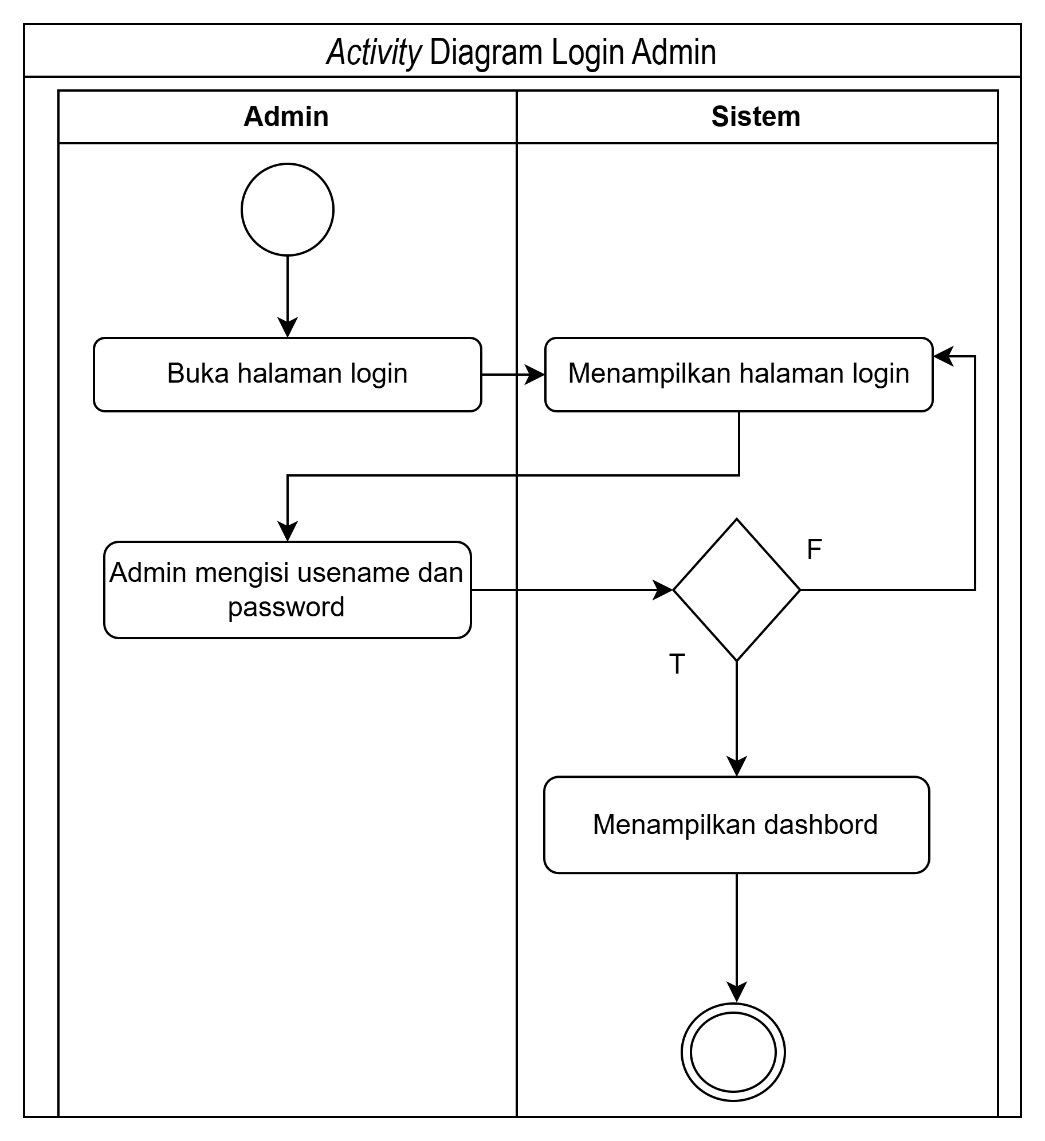
**2. *Login* Mahasiswa**

****

Gambar 3.4 *Activity* Diagram *Login* Mahasiswa

**Penjelasan:**

Pada proses diatas menandahkan *login* sebagai Mahasiswa dimana Mahasiswa *login* lalu sistem akan menampilkan halaman *login* dan dilanjutkan dengan Mahasiswa mulai mengisi akun (*Password* dan *Username*) yang telah di daftarkan pada *form* registrasi Mahasiswa tadi, lalu dilanjutkan sistem akan menampilkan halaman *login* dari *website*.

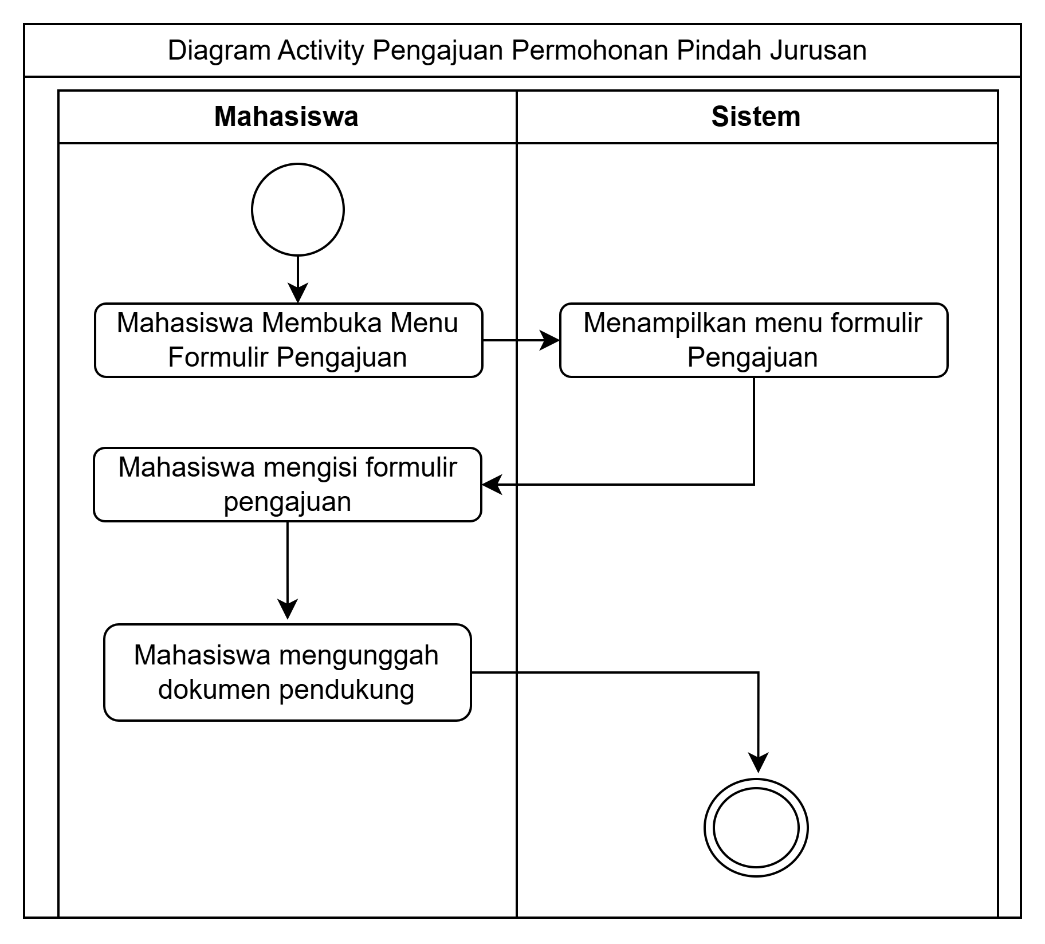
**3. *Login* Admin**

Gambar 3.5 *Activity* Diagram *Login* Admin

**Penjelasan:**

Pada proses diatas menandahkan *login* sebagai Admin dimana Admin *login* lalu sistem akan menampilkan halaman *login* dan dilanjutkan dengan Admin mulai mengisi akun (*Password* dan *Username*) yang telah di daftarkan pada *form* registrasi Admin tadi, lalu dilanjutkan sistem akan menampilkan halaman *login* dari *website*.

**4. Pengajuan Permohonan Pindah Jurusan**

****

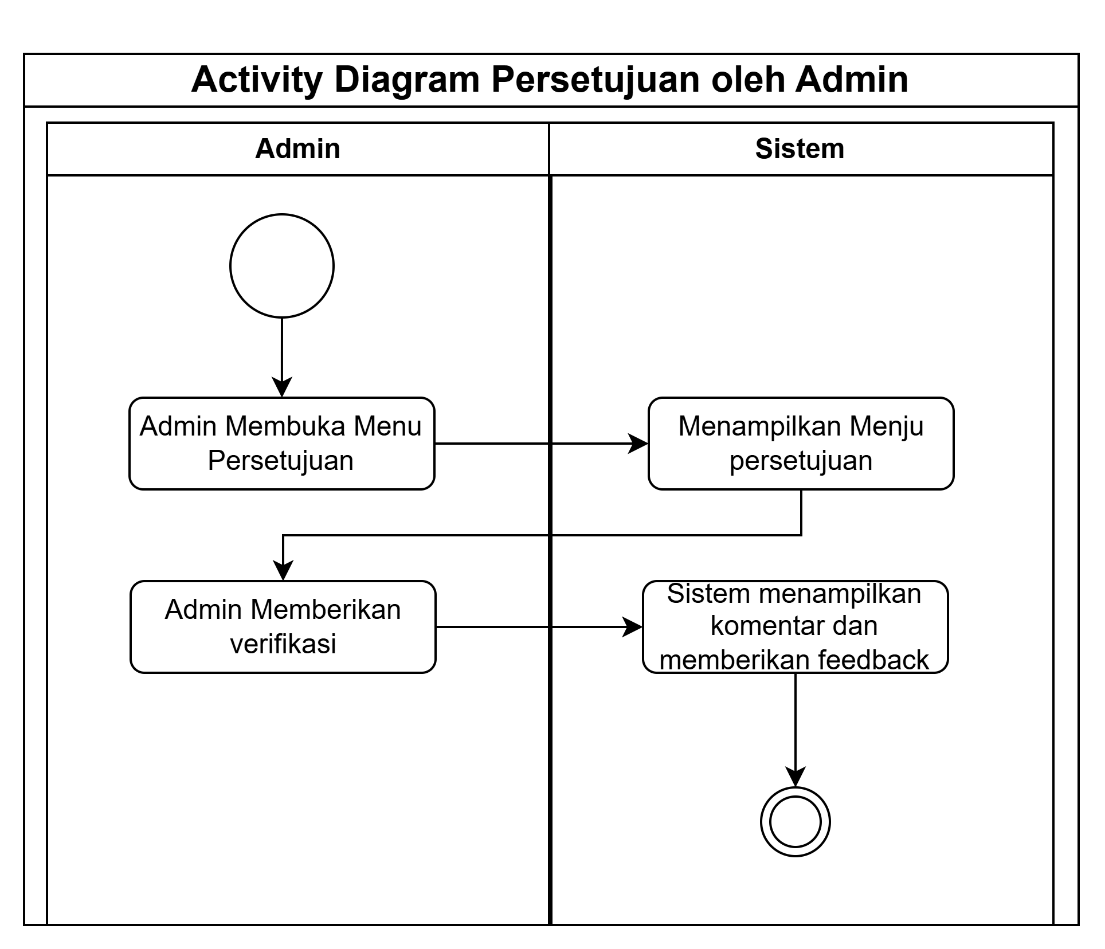


Gambar 3.6 *Activity* DiagramPengajuan Permohonan Pindah Jurusan

**Penjelasan:**

Diagram diatas dimulai dengan Mahasiswa yang membuka menu formulir pengajuan untuk mengajukan pindah jurusan, setelah itu sistem menampilkan menu formulir pengajuan dan mahasiswa mengisi formulir pengajuan tersebut, dan mahasiswa juga dapat mengunggah dokumen pendukung

**5. Persetujuan oleh Admin**

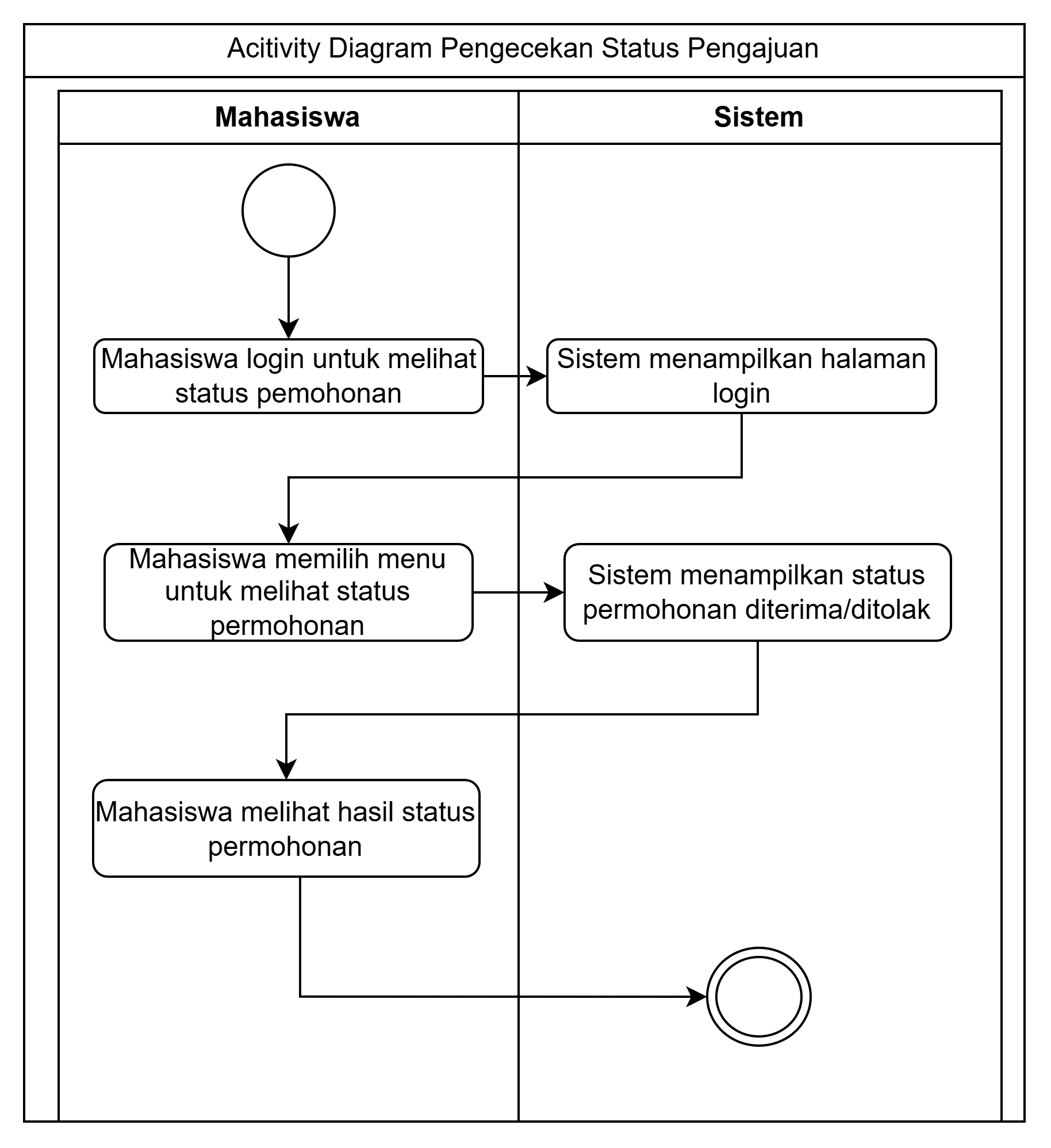
****

Gambar 3.7 *Activity* DiagramPersetujuan oleh Admin

**Penjelasan:**

Diagram di atas menunjukkan proses persetujuan permohonan oleh admin. Mahasiswa mengajukan permohonan, kemudian sistem menyimpan data permohonan tersebut. Admin masuk ke sistem, memilih permohonan yang akan diverifikasi, melakukan verifikasi, lalu memberikan *feedback* ke mahasiswa. Sistem ini memberi notifikasi kepada mahasiswa, dan mahasiswa bisa melihat hasil permohonannya. Diagram ini mengilustrasikan alur kerja antara mahasiswa, sistem, dan admin dalam pengelolaan persetujuan permohonan.

**6. Pengecekan Status Pengajuan**

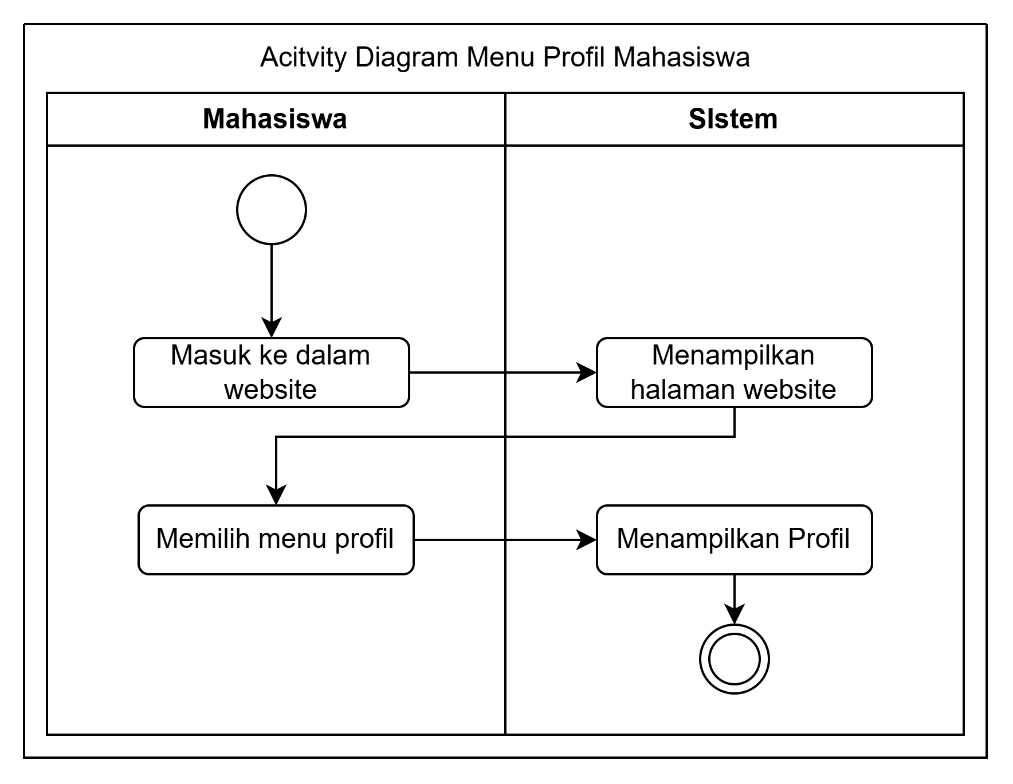


Gambar 3.8 *Activity* DiagramPengecekan Status Pengajuan

**Penjelasan:**

Diagram diatas dimulai dengan Mahasiswa harus *login* terlebih dahulu, lalu sistem akan menampilkan halaman login lalu mahasiswa memilih menu untuk melihat status permohonan, dilanjutkan sistem menampilkan status permohonannya, lalu mahasiswa melihat hasil dari status permohonan yang diberikan oleh sistem.

**7. Profil Mahasiswa**

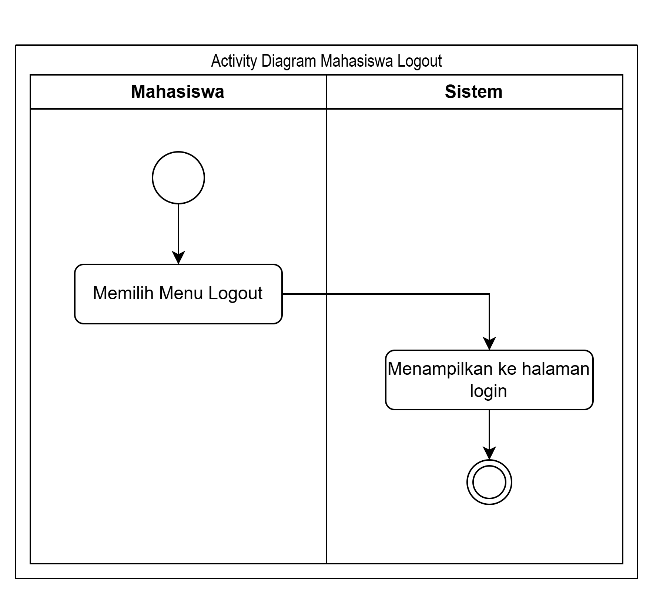
****

Gambar 3.9 *Activity* DiagramProfil Mahasiswa

**Penjelasan:**

Diagram diatas menjelaskan bahwa Mahasiswa memilih menu porfil lalu nanti sistem akan menamilkan halaman profil dari mahasiswa.

**8. Mahasiswa Logout**

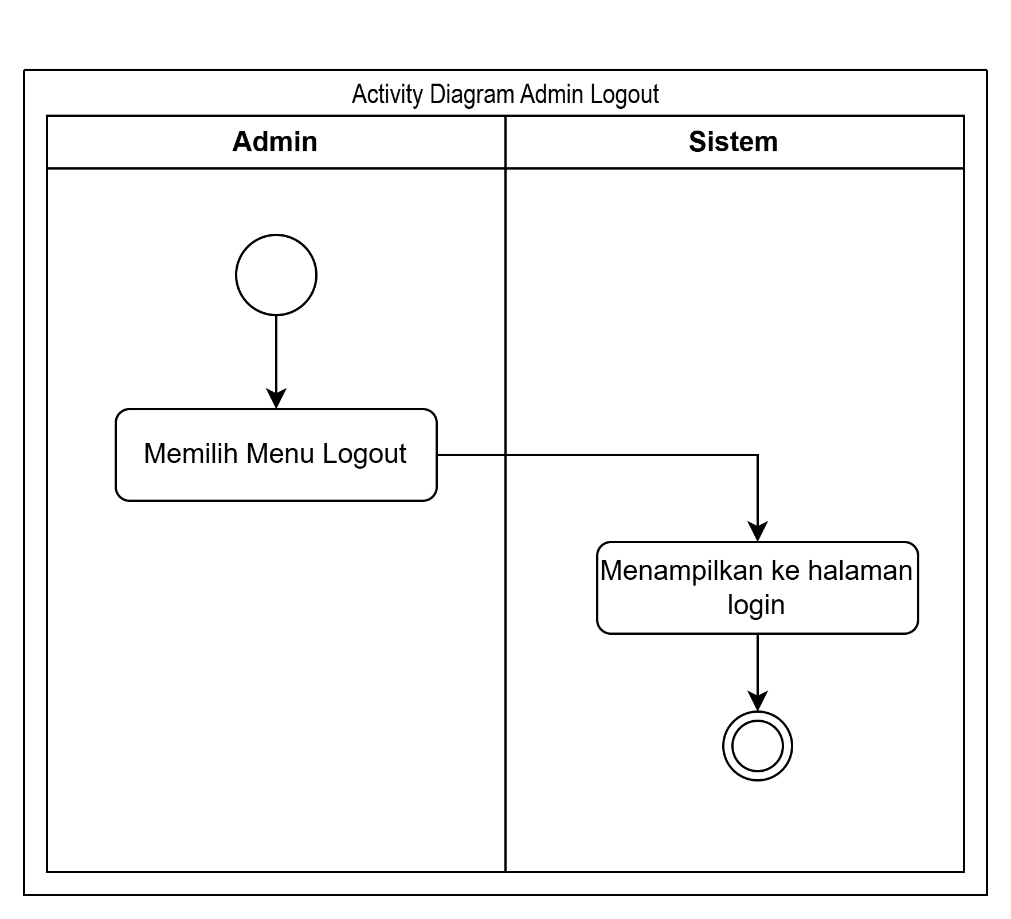
****

Gambar 3.10 *Activity* DiagramMahasiswa *Logout*

**Penjelasan:**

Mahasiswa mengklik logout lalu system menampilkan Kembali ke tampilan login awal.

**9. Admin Logout**

****

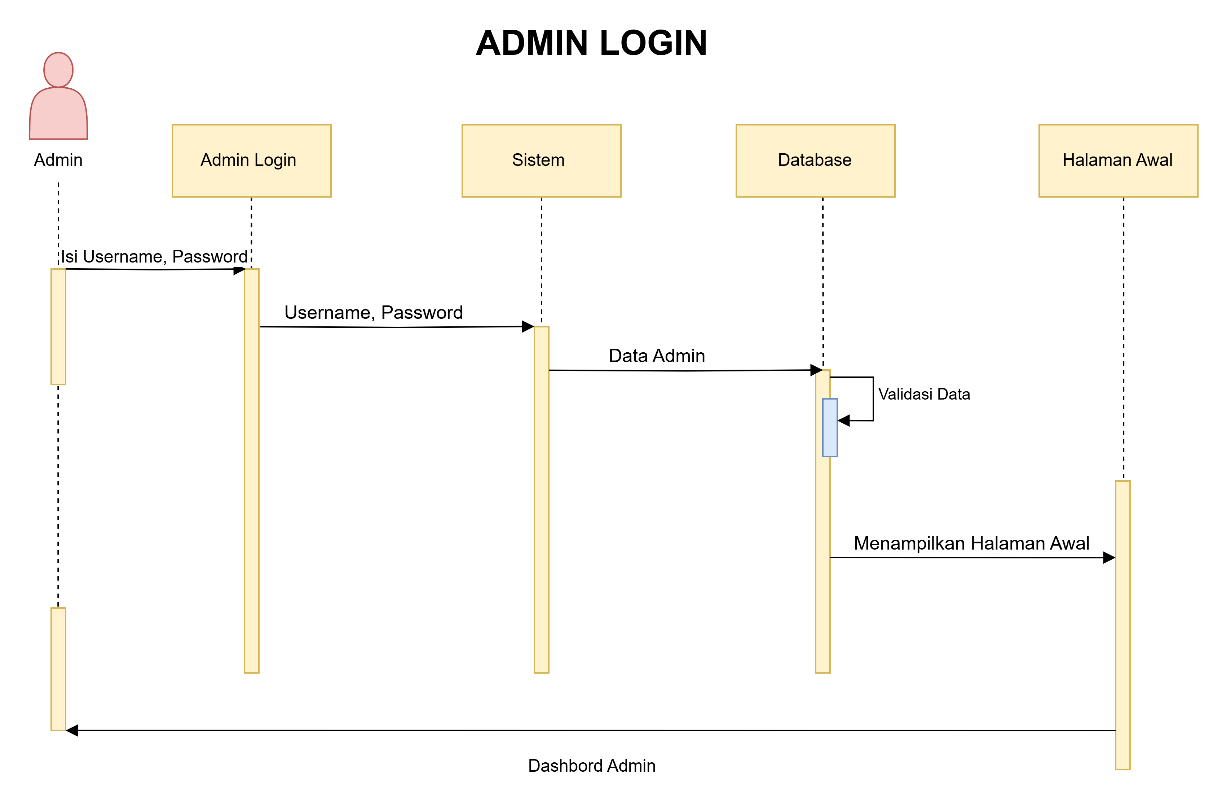
Gambar 3.11 *Activity* DiagramAdmin *Logout*

**Penjelasan:**

Admin mengklik *logout* lalu system menampilkan Kembali ke tampilan halaman login awal.

**3.2.4. *Sequence* Diagram**

**1. Admin *Login***

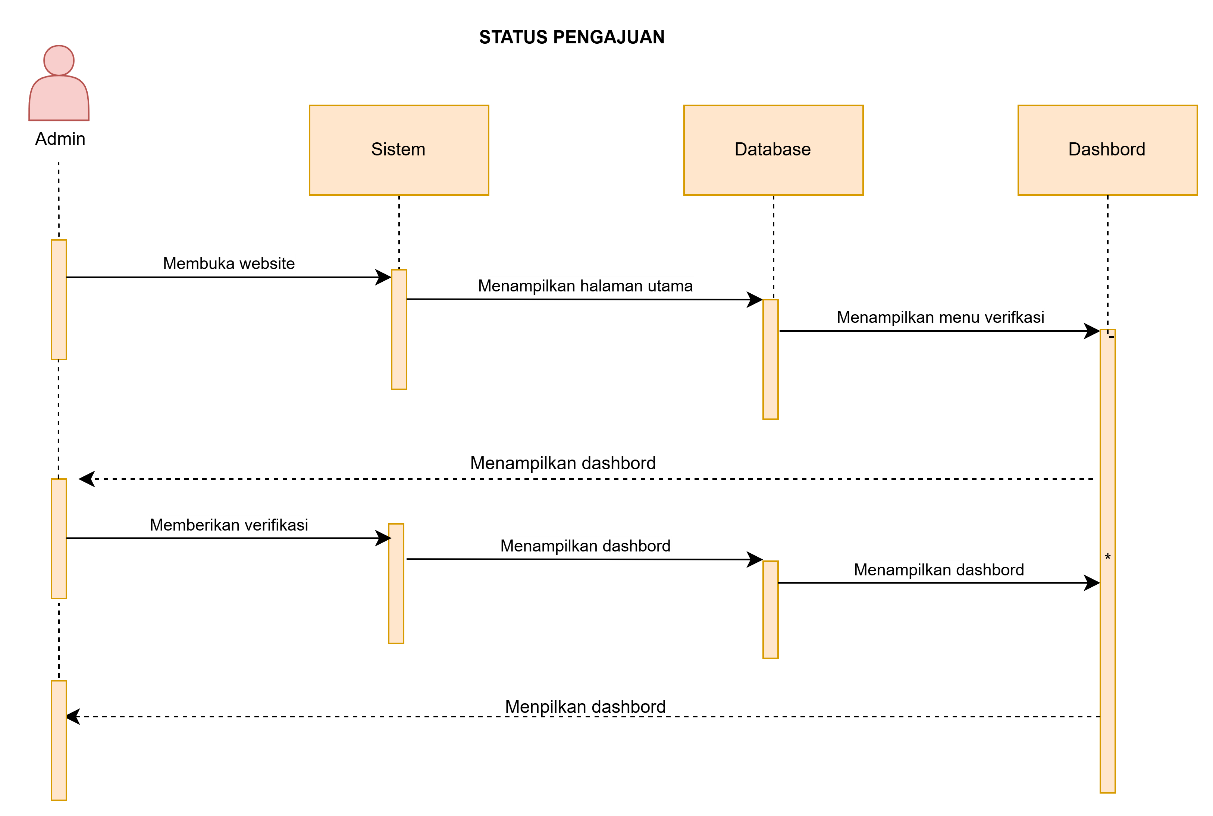
****

Gambar 3.12 *Sequence*DiagramAdmin *Login*

**Penjelasan:**

Menjelaskan *Login* Admin pada sistem. *Database* *Dashboard* akan memverivikasi di database sesuai *username* dan *password* yang dimasukkan Admin. Jika sudah sesuai dengan yang di *database*, maka sistem akan mengecek status admin, dimana setiap *user* memiliki halaman utama masing-masing. Jika tidak ditemukan kecocokan *username* dan *password* yang dimasukkan, maka sistem akan memberi peringatan *username* dan *password* salah. Jika sudah benar maka *database* akan memvalidasi data dan ditampilkan ke layar.

**2. Persetujuan Status Pengajuan**

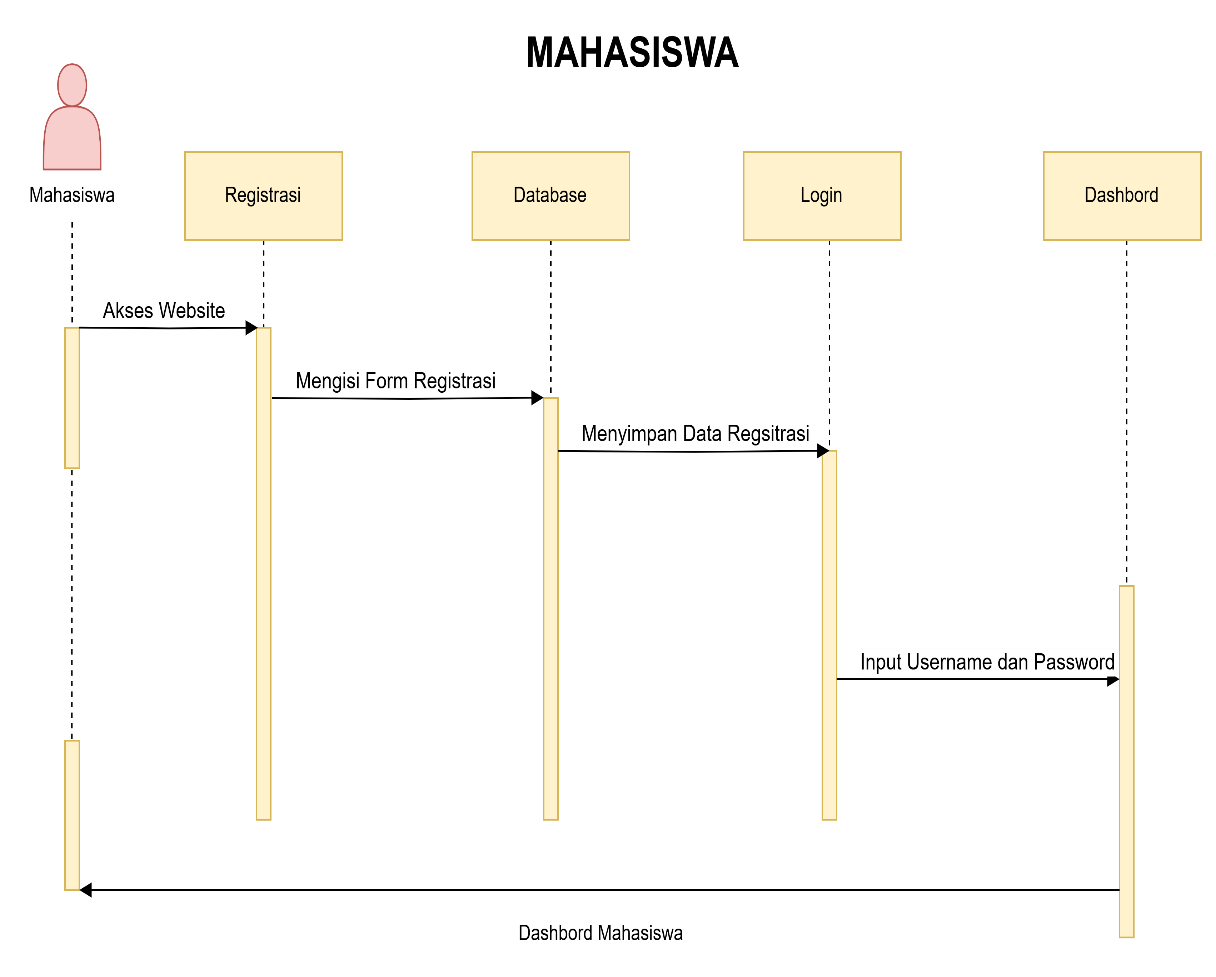
****

Gambar 3.13 *Sequence*DiagramStatus Pengajuan

**Penjelasan:**

Diagram di atas menggambarkan alur interaksi admin dengan sistem dalam mengakses halaman utama, melakukan verifikasi, dan menampilkan dashboard. Awalnya, admin membuka *website*, dan sistem menampilkan halaman utama. Setelah itu, admin menuju menu verifikasi, dan sistem menampilkannya dengan mengambil data dari database. Admin memberikan verifikasi, dan sistem mengonfirmasi verifikasi tersebut dengan database, lalu menampilkan dashboard yang diambil dari database ke admin.

**3. Mahasiswa Registrasi**

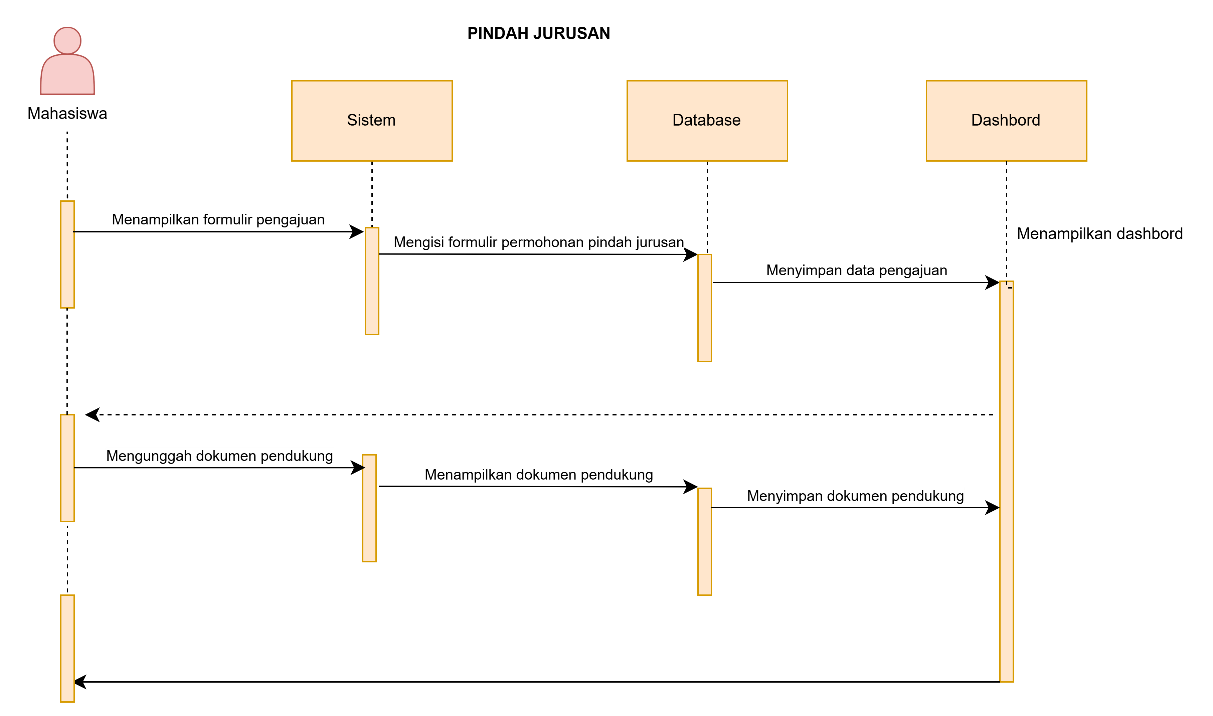
****

Gambar 3.14 *Sequence*DiagramMahasiswa

**Penjelasan:**

Diagram sequence di atas menggambarkan alur proses registrasi dan *login* pengguna di sebuah sistem. Pertama, pengguna mengakses *website* dan diarahkan ke halaman registrasi. Di halaman registrasi, pengguna mengisi form registrasi dengan data yang diperlukan. Setelah *form* registrasi di isi, data tersebut disimpan ke dalam database. Setelah data berhasil disimpan, pengguna kemudian menuju halaman *login*, di mana mereka diminta untuk menginput *username* dan *password* yang telah didaftarkan. Jika proses login berhasil, sistem menampilkan *dashboard* sebagai halaman utama bagi pengguna untuk melanjutkan aktivitas mereka di dalam sistem.

**4. Pengajuan Pindah Jurusan**

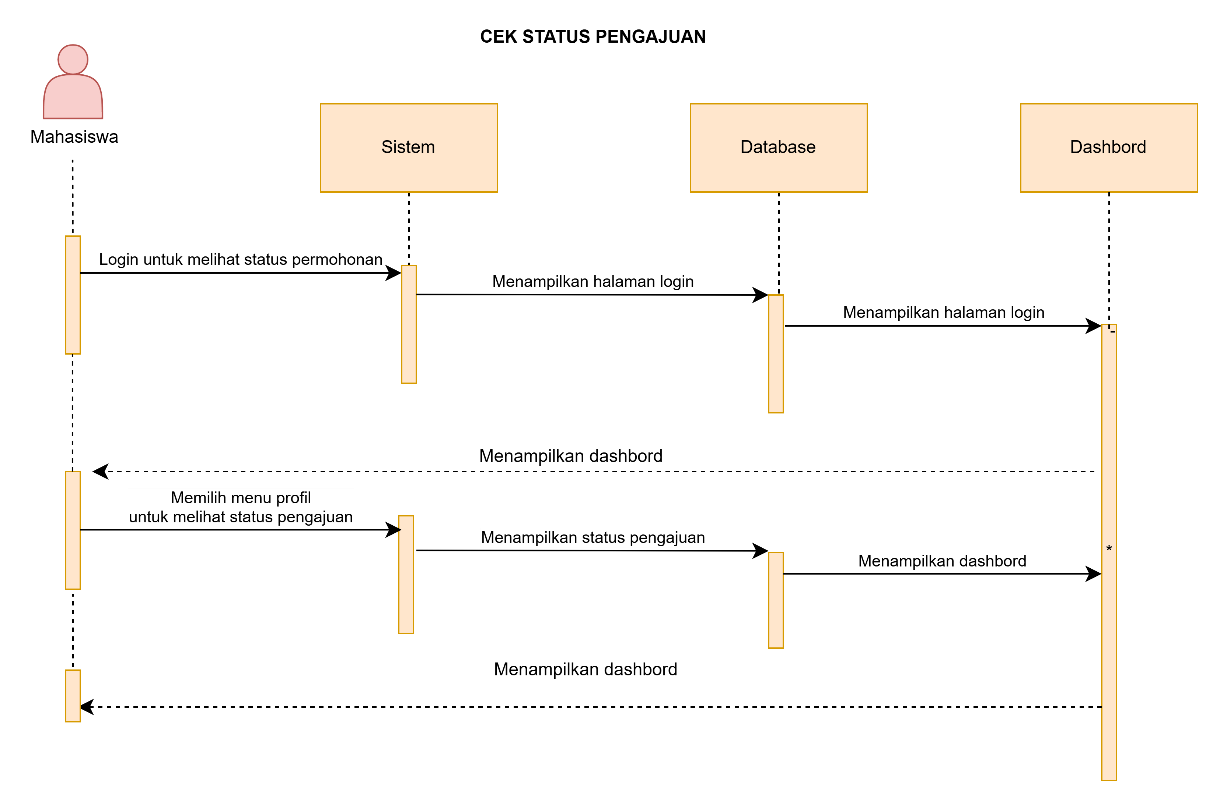
****

Gambar 3.15 *Sequence*DiagramPengajuan Pindah Jurusan

**Penjelasan:**

Mahasiswa mengisi formulir permohonan pindah jurusan melalui antarmuka yang disediakan oleh Sistem. Sistem kemudian menampilkan formulir pengajuan tersebut dan selanjutnya menyimpan data pengajuan kedalam *database*. Setelah data pengajuan berhasil disimpan, pengguna mengunggah dokumen pendukung yang diperlukan untuk proses pindah jurusan. Sistem akan menampilkan konfirmasi unggahan dokumen pendukung tersebut. Kemudian, dokumen-dokumen tersebut juga disimpan dalam *database*.

**5. Pengecekan Status Pengajuan**

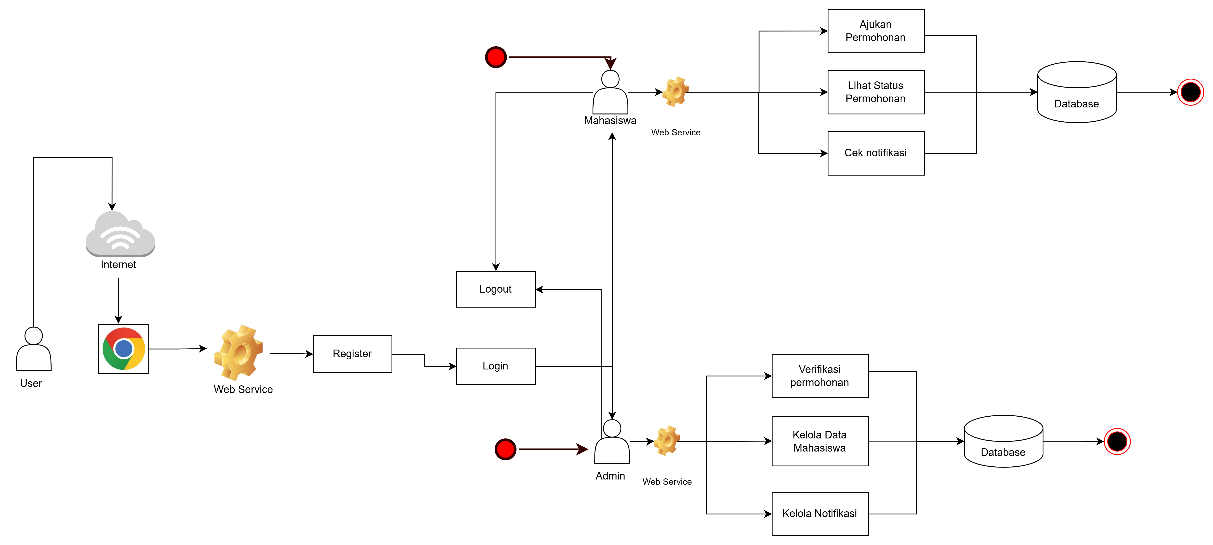
****

Gambar 3.16 Sequence Diagram Pengajuan Pindah Jurusan

**Penjelasan:**

Dimulai dengan Mahasiswa melakukan *login* untuk mengakses status permohonan mereka. Sistem kemudian menampilkan halaman *login*, dan database memastikan kredensial pengguna yang valid sebelum menampilkan halaman login pada dashboard. Setelah *login* berhasil, pengguna memilih menu profil untuk melihat status pengajuan mereka. Sistem kemudian memproses permintaan tersebut, mengakses database untuk menampilkan data status pengajuan, dan menampilkan status tersebut pada dashboard untuk dilihat pengguna.

**3.2.5. ArsitekturDiagram**

****

Gambar 3.17 Arsitektur Diagram

**Penejelasan:**

Diagram ini menggambarkan alur kerja sistem aplikasi berbasis web untuk pengelolaan permohonan mahasiswa. Pengguna mengakses aplikasi melalui browser yang terhubung ke server melalui internet. Di sisi server, terdapat layanan web yang mengelola proses registrasi dan login pengguna. Setelah pengguna berhasil *login*, mereka dapat mengakses fitur-fitur utama, seperti mengajukan permohonan, melihat status permohonan, dan memeriksa notifikasi. Semua data permohonan dan notifikasi pengguna disimpan di dalam *database*.

**4. Kebutuhan Antar Muka Eksternal**

1. **Antarmuka Pengguna**

Antarmuka ini akan dibuat dalam bentuk konsep desain atau sketsa, meliputi tampilan untuk mahasiswa dan admin, yang memudahkan proses pengajuan, verifikasi, dan notifikasi.

1. **Antarmuka Perangkat Lunak**

Kebutuhan perangkat keras yang dibutuhkan dalam menggunakan sistem ini adalah menggunakan hosting untuk *server web* yang membantu dalam pengeloaan server secara *real time*.

1. **Antarmuka Perangkat Lunak**

Aplikasi kami berbasis web yang dapat diakses jika terhubung dengan Internet dan dapat dibuka dengan web *browser* apapun. Jadi tidak diperlukan *browser* khusus.

1. **Antarmuka Komunikasi**

Antarmuka Komunikasi pada aplikasi kami menggunakan aplikasi web, karena perangkat ini digunakan mendukung model sistem *client-server* dan dapat digunakan baik dalam jaringan internet hingga intranet berbasis protokol *transmission control protocol/ internet protocol*.

**5. Estimasi Biaya Sistem**

**5.1. Rincian gaji per hari**

Tabel 5.1. Gaji Per Hari

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Hari | *System Analyst* | *Frontend* Dev | *Backend* Dev | *Database* |
| 1 Hari | Rp2.000.000 | Rp900.000 | Rp1.000.000 | Rp800.000 |

**5.2. Estimasi Detail**

Tabel 5.2 Estimasi Detail

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Modul | Waktu Kerja | Role | Gaji |
| Perancangan SDLC | 14 days | *System Analyst* | Rp28.000.000 |
| Desain Antarmuka | 10 days | *System Analyst* | Rp20.000.000 |
| *User Module*  *View Module* | 14 *days* | *Frontend Dev* | Rp12.600.000 |
| *pengembang Backend* | 18 *days* | *Backend Dev* | Rp18.000.000 |
| *Database* dan infrastruktur | 10 *days* | *Database Analyst* | Rp8.000.000 |
| **Total** | **66 *days*** |  | **Rp86.600.000** |

**5.3. Gaji Total Developer**

Tabel 5.3 Gaji Total Developer

|  |  |  |  |
| --- | --- | --- | --- |
| Tugas | | Total | Gaji Total |
| Backend Dev | | 18 hari | Rp18.000.000 |
| Frontend Dev | | 14 hari | Rp12.600.00 |
| Database Analyst | | 10 hari | Rp8.000.000 |
| System Analyst | | 24 hari | Rp48.000.000 |
|  | **Total** | | **Rp86.600.000** |