

## **BAB IV**

### **HASIL DAN ANALISIS PENELITIAN**

#### **4.1 Analisa Perancangan Sistem**

Analisis perancangan sistem yang saat ini beroperasi dan penilaian kebutuhan yang telah dijelaskan sebelumnya di Bab III adalah dasar dari desain sistem ini. Penulis menemukan solusi untuk masalah yang ditemukan dalam penelitian ini dengan merancang pengembangan modul *Helpdesk* dengan fitur konsultasi dan *live chat* pada aplikasi advokasi mahasiswa. Pada fitur konsultasi nantinya akan terdapat elemen-elemen yang dirancang untuk memberikan informasi tentang status dan kategori konsultasi yang sedang maupun sudah diajukan. Informasi yang dimaksud meliputi: status konsultasi seperti tertunda, disetujui, dalam perkembangan, sukses, dan ditolak. Selanjutnya, kategori daftar kasus konsultasi meliputi: administrasi akademik, administrasi blended, penelitian & inovasi, pengabdian masyarakat, kemahasiswaan & alumni, pengembangan karakter, pengembangan bahan ajar, pengembangan outcome based education, layanan pemustakaan, layanan teknis, layanan *digital* & sistem informasi, kolaborasi & kemitraan, humas & internasionalisasi, pengembangan pegawai, pengelolaan pegawai, operasional TIK, dan pemasaran & admisi. Tentunya, fitur konsultasi ini akan menggunakan desain yang menarik dan intuitif untuk memberikan pengalaman pengguna (*user experience*) yang optimal. Selain itu, pada fitur *live chat* nantinya akan terdapat beberapa elemen yang dirancang untuk memberikan informasi histori percakapan yang sedang maupun sudah dilakukan. Informasi yang dimaksud meliputi: Bukti percakapan, Kategori percakapan, Waktu percakapan, Status percakapan, dan Tracking percakapan. Pengembangan fitur ini bertujuan untuk memfasilitasi akses informasi yang lebih tepat guna tentang advokasi yang lebih mudah dipahami dan diakses oleh mahasiswa. Metode ini diharapkan dapat menyediakan cara terbaik untuk meningkatkan standar layanan yang mendukung upaya advokasi mahasiswa. Selain itu, sistem ini dikembangkan menggunakan teknologi berbasis web dengan bahasa pemrograman HTML, PHP, CSS, dan JavaScript. yang akan mempermudah pengguna dalam mengakses sistem

atau aplikasi tersebut. Oleh karena itu, Berikut adalah perbandingan antara sistem yang sedang beroperasi dengan sistem yang direncanakan untuk dikembangkan.

Tabel 4. 1 Tabel Perbandingan Sistem

Indikator	Sistem Lama	Sistem Baru
Konsultasi	Sistem konsultasi mahasiswa saat ini masih konvensional, mengharuskan pertemuan langsung yang tidak efisien dan menyulitkan. Masalah jarak, jadwal, dan transportasi menjadi hambatan. Oleh karena itu, perlu dikembangkan sistem konsultasi berbasis <i>web</i> untuk meningkatkan aksesibilitas, fleksibilitas, dan efisiensi. dan efisiensi dalam proses konsultasi.	Sistem konsultasi berbasis <i>web</i> memungkinkan mahasiswa untuk melakukan konsultasi secara online tanpa perlu ke lokasi fisik. Mereka dapat melacak status konsultasi secara real-time, mengelola konsultasi, dan mengakses detail dengan mudah. Ini memudahkan layanan konsultasi, mengatasi kendala waktu dan lokasi dari sistem konvensional.
<i>Live chat</i>	Sistem konsultasi lama mengharuskan mahasiswa untuk menggunakan platform eksternal agar bisa berkomunikasi dengan pihak terkait, yang menyebabkan privasi dalam pengajuan konsultasi menjadi kurang terjaga.	Dengan sistem <i>live chat</i> baru, mahasiswa dapat langsung berkomunikasi dalam sistem secara real-time dengan pihak terkait. Hal ini memudahkan proses konsultasi dengan menyediakan akses langsung dan interaksi instan tanpa harus bergantung pada platform eksternal.

## 4.2 Perancangan Diagram Sistem Usulan

Pendekatan OOAD (*Object Oriented Analysis and Design*) yang dikombinasikan dengan UML (*Unified Modeling Language*) adalah metode standar yang digunakan oleh penulis untuk pengembangan diagram sistem. Industri pengembangan perangkat lunak menggunakan metodologi OOAD dan UML secara ekstensif untuk memberikan pengembang perangkat lunak struktur kerangka kerja yang terstruktur dan modular. Di awal proses desain, teknik OOAD sangat berguna untuk menciptakan pengalaman pengguna sistem ini. Para penulis akan menggunakan berbagai diagram, seperti diagram aktivitas, diagram urutan, diagram kelas, diagram kasus penggunaan, dan deskripsi kasus penggunaan, untuk menunjukkan bagaimana sistem dan pengguna berinteraksi.

### 4.2.1 Diagram Kasus Pengguna (*Use Case Diagram*)

Diagram kasus pengguna sangat krusial dalam mengilustrasikan fungsi inti dari sistem yang akan dikembangkan, terutama untuk modul *Helpdesk* dengan fitur konsultasi dan *live chat* dalam aplikasi advokasi mahasiswa. Dalam konteks ini, baik sistem maupun mahasiswa berperan aktif. Diagram *Use Case* memiliki peran penting dalam memahami dan merancang fitur-fitur ini dalam aplikasi advokasi mahasiswa dengan melibatkan pihak-pihak terkait secara visual.

Proses pengembangan modul *Helpdesk* dimulai dengan menggunakan Diagram *Use Case*, yang kemudian dievaluasi dan di *review* oleh mahasiswa yang terlibat dalam proses ini. Umpan balik dari mahasiswa diperlukan untuk memastikan bahwa diagram tersebut sesuai dengan mekanisme serta komunikasi yang berlangsung di lingkungan advokasi mahasiswa. Langkah ini sangat penting untuk mengidentifikasi serta mengatasi kelemahan yang mungkin ada dalam sistem yang sedang berjalan saat ini.

Hasil Analisa Kebutuhan *User*:




- a. Mahasiswa dapat melakukan pengajuan konsultasi
- b. Mahasiswa dapat melihat konsultasi yang sudah diajukan
- c. Mahasiswa dapat mengedit konsultasi yang diajukan
- d. Mahasiswa dapat cek status konsultasi
- e. Admin dapat melakukan validasi status konsultasi
- f. Mahasiswa dapat mengirim pesan terhadap admin terkait
- g. Admin terkait dapat membalas pesan terhadap mahasiswa
- h. Pesan *live chat* terkirim secara real time

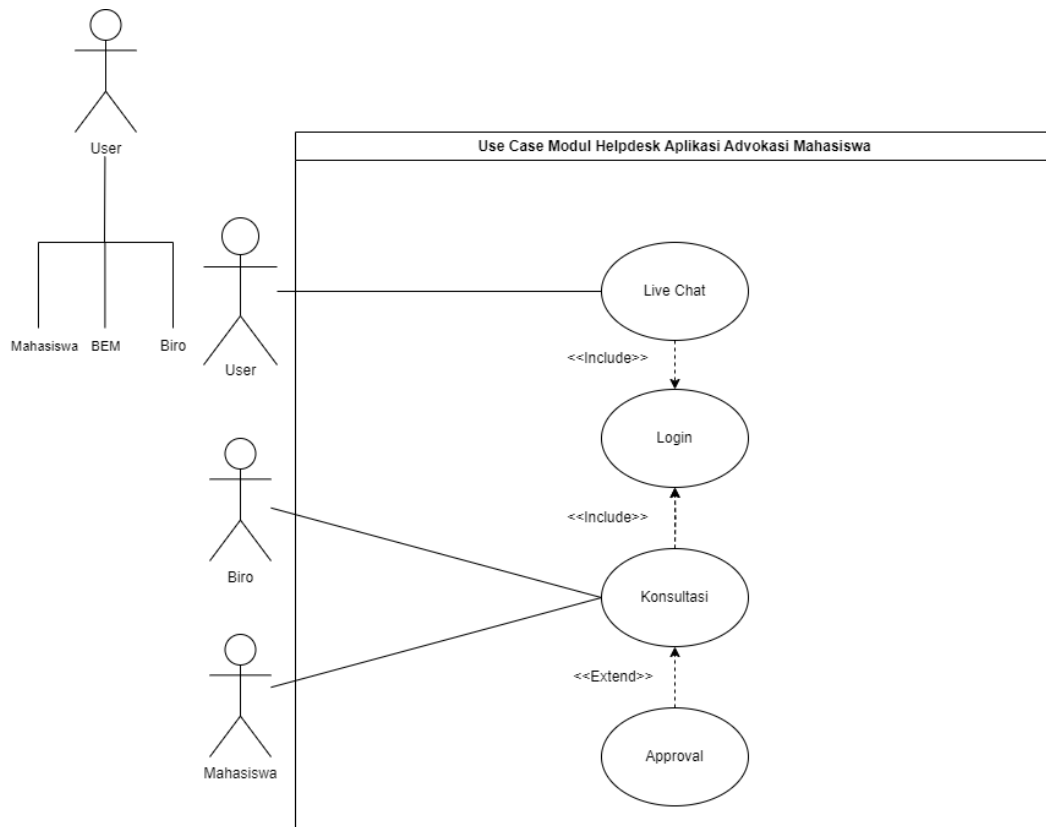
Hasil Identifikasi dari Elisitasi Final

- a. Mahasiswa dapat mengajukan konsultasi
- b. Mahasiswa dapat mengirimkan pesan secara *real time*

Daftar Task (Usecase) yang harus dibuat:

Tabel 4. 2 Daftar task (Use Case) yang harus dibuat

No	Notasi	Penjelasan
1		Aktor yang mendapatkan role sebagai <i>user</i> yang ingin dikembangkan yaitu Mahasiswa, BEM dan <i>Unit</i> .
4		<i>Use Case</i> yang didalamnya terdapat fitur konsultasi yang ditujukan untuk mahasiswa mengajukan konsultasi sebelum melaporkan aduan advokasi.
5		<i>Use Case</i> yang didalamnya terdapat fitur <i>live chat</i> yang ditujukan untuk mahasiswa agar bisa berinteraksi secara langsung.



Gambar 4. 1 Use Case Modul Helpdesk

Berikut merupakan penjelasan dari setiap *Use Case* pada modul *Helpdesk* aplikasi advokasi mahasiswa pada gambar :

a. *Konsultasi*

Proses *user* dapat melakukan pengajuan konsultasi, melihat daftar konsultasi, merubah data konsultasi. Kemudian, admin terkait dapat melakukan validasi.

b. *Live chat*

Proses *user* dapat melakukan interaksi secara *real time*, melihat riwayat pesan, melihat status pesan melalui fitur *live chat*.

c. *Login*

Proses *user* memasukan email dan password mereka untuk mengakses kedalam aplikasi

#### 4.2.2 Use Case Description

Sebuah dokumen yang dikenal sebagai deskripsi kasus penggunaan membahas secara mendalam tentang bagaimana sebuah sistem atau aplikasi berperilaku dari sudut pandang administrator dan pengguna yang terlibat. Hasil dari deskripsi kasus penggunaan yang dibuat tercantum di bawah ini.

Tabel 4. 3 Use Case Description Login

<b>Use Case</b>	<i>Login</i>	
<b>Trigger</b>	<i>User</i> ingin mengakses kedalam aplikasi advokasi mahasiswa	
<b>Actors</b>	<i>User</i> (Mahasiswa)	
<b>Pre-condition</b>	<i>User</i> harus memiliki akun yang terdaftar dalam sistem advokasi mahasiswa.	
<b>Normal Course</b>	Actor	system
	<ol style="list-style-type: none"> <li>1. <i>User</i> melakukan <i>input</i> data <i>login</i> akun yang valid, mencakup:             <ol style="list-style-type: none"> <li>a. Alamat email yang terdaftar dalam sistem. Password yang sesuai dengan akun yang sudah di daftarkan</li> </ol> </li> <li>2. <i>User</i> melakukan klik pada tombol <i>Login</i></li> </ol>	<ol style="list-style-type: none"> <li>1. Sistem memverifikasi informasi <i>login</i> yang dimasukkan oleh <i>user</i>.</li> <li>2. Sistem memeriksa dan membandingkan informasi <i>login</i> dengan data yang ada di basis data akun yang disimpan.</li> <li>3. Sistem memberikan izin akses ke akun yang sesuai.</li> <li>4. Sistem memberikan akses lengkap ke semua fitur dan fungsionalitas yang diizinkan untuk akun tersebut.</li> </ol>
<b>Post-condition</b>	Jika <i>login</i> berhasil, <i>user</i> akan memiliki akses lengkap ke semua fitur yang tersedia dalam sistem.	
<b>Alternative Course</b>	Jika informasi yang dimasukkan <i>user</i> tidak valid atau tidak sesuai, maka: <ul style="list-style-type: none"> <li>• Sistem akan menolak akses dan menampilkan notifikasi atau pesan kesalahan.</li> <li>• <i>User</i> akan diberikan kesempatan untuk mencoba <i>login</i> kembali dengan informasi yang benar.</li> </ul>	

Tabel 4. 4 Use Case Description Konsultasi

<b>Use Case</b>	Konsultasi	
<b>Trigger</b>	<i>User</i> ingin mengakses fitur konsultasi	
<b>Actors</b>	<i>User</i> (Mahasiswa)	
<b>Pre-condition</b>	<i>User</i> harus sudah melakukan <i>login</i> kedalam aplikasi advokasi mahasiswa.	
<b>Normal Course</b>	Actor	system
	<ol style="list-style-type: none"> <li>1. <i>User</i> melakukan akses fitur konsultasi</li> <li>2. <i>User</i> mengklik tombol “tambah”</li> <li>3. <i>User</i> mengisi <i>form</i> dan klik <i>submit</i></li> </ol>	<ol style="list-style-type: none"> <li>1. Sistem akan menampilkan halaman konsultasi</li> <li>2. Sistem akan menampilkan form detail konsultasi</li> <li>3. Data terinput kedalam halaman konsultasi</li> <li>4. Menampilkan data yang di input</li> </ol>
<b>Post-condition</b>	Jika <i>user</i> sudah melakukan pengajuan konsultasi maka data yang diajukan akan tampil.	
<b>Alternative Course</b>	<p>Jika Mahasiswa ingin melakukan Edit Konsultasi, maka:</p> <ul style="list-style-type: none"> <li>• <i>User</i> mengklik tombol logo “edit”</li> <li>• Maka sistem akan menampilkan form edit data konsultasi</li> <li>• <i>User</i> mengubah isi <i>form</i> dan klik <i>submit</i></li> <li>• Data berhasil berubah kedalam halaman konsultasi</li> </ul>	

Tabel 4. 5 Use Case Description Approval

<b>Use Case</b>	Mengubah Status Pengajuan Mahasiswa	
<b>Trigger</b>	Mahasiswa melakukan pengajuan	
<b>Actors</b>	Biro	
<b>Pre-condition</b>	Biro sudah melakukan login ke sistem dan memiliki hak akses untuk mengubah status pengajuan	
<b>Normal Course</b>	Actor	system
	<ol style="list-style-type: none"> <li>1. Biro mengakses fitur konsultasi</li> <li>2. Biro memilih pengajuan yang ingin diubah statusnya</li> <li>3. Biro memberikan putusan status pengajuan mahasiswa</li> </ol>	<ol style="list-style-type: none"> <li>1. Sistem menampilkan halaman konsultasi</li> <li>2. Sistem menampilkan data yang diajukan oleh mahasiswa</li> <li>3. Sistem mengubah status data dan menampilkan konfirmasi bahwa status telah berhasil diubah</li> </ol>
<b>Post-condition</b>	Status pengajuan mahasiswa berhasil diubah dan tersimpan dalam system.	
<b>Alternative Course</b>		



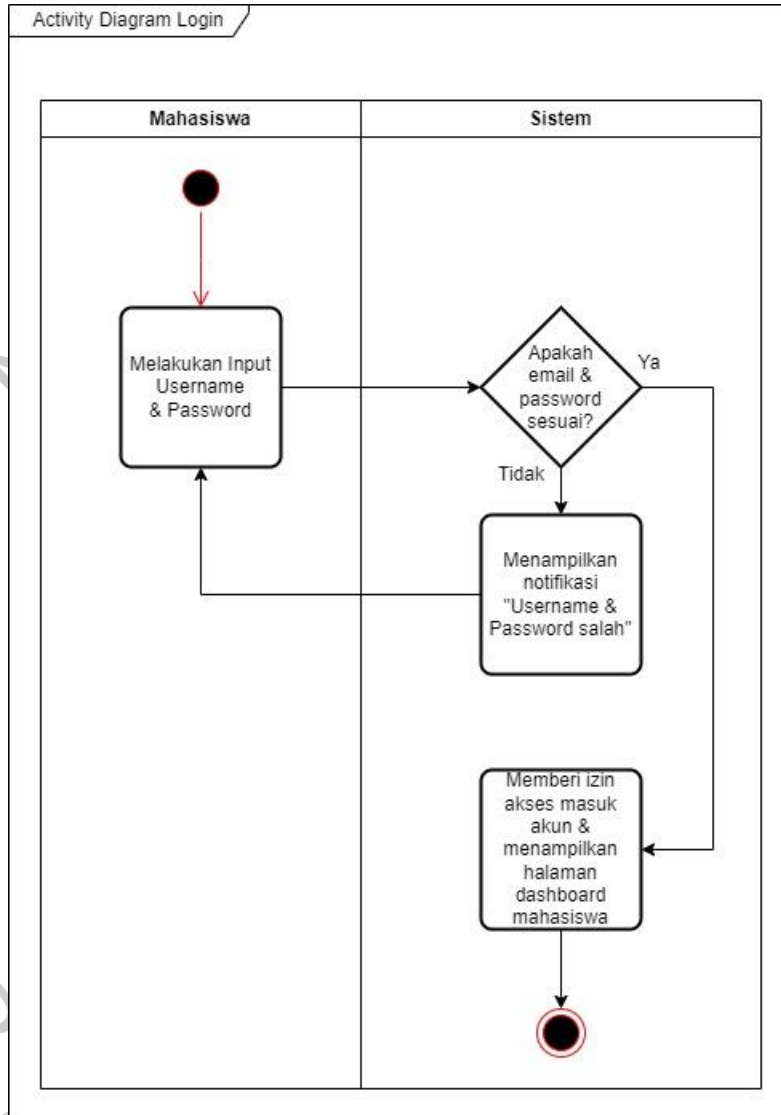
Tabel 4. 6 Use Case Description user live chat

<b>Use Case</b>	<i>User live chat</i>	
<b>Trigger</b>	<i>User ingin berinteraksi secara real time</i>	
<b>Actors</b>	<i>User (Mahasiswa)</i>	
<b>Pre-condition</b>	<i>User harus sudah melakukan login kedalam aplikasi advokasi mahasiswa.</i>	
<b>Normal Course</b>	Actor	system
	<ol style="list-style-type: none"> <li>4. <i>User mengakses fitur live chat</i></li> <li>5. <i>User bisa memilih ingin berinteraksi dengan admin terkait</i></li> <li>6. <i>User mengirim pesan kepada admin terkait</i></li> </ol>	<ol style="list-style-type: none"> <li>4. <i>Sistem akan menampilkan halaman live chat</i></li> <li>5. <i>Sistem akan menampilkan halaman chat dengan admin terkait</i></li> <li>6. <i>Sistem akan menampilkan pesan yang dikirim user kepada admin terkait</i></li> </ol>
<b>Post-condition</b>	<i>Pesan yang dikirimkan oleh user tampil di halaman chat admin</i>	
<b>Alternative Course</b>	-	

#### 4.2.3 Diagram Aktivitas (*Diagram Activity*)

Salah satu jenis diagram UML yang sering digunakan untuk menunjukkan proses atau aktivitas di dalam sistem adalah diagram aktivitas. Grafik ini menunjukkan keputusan yang dibuat selama proses tertentu, urutan pembuatan keputusan tersebut, dan transfer kontrol dari satu aktivitas ke aktivitas lainnya. Diagram aktivitas membantu dalam memahami bagaimana sebuah sistem atau proses berfungsi dengan memanfaatkan simbol-simbol grafis seperti simbol pengontrol, garis penghubung, dan

aktivitas. Alur kerja dan proses bisnis juga dapat dirancang, dianalisis, didokumentasikan, dan dikomunikasikan kepada pemangku kepentingan yang berbeda dengan bantuan mereka.

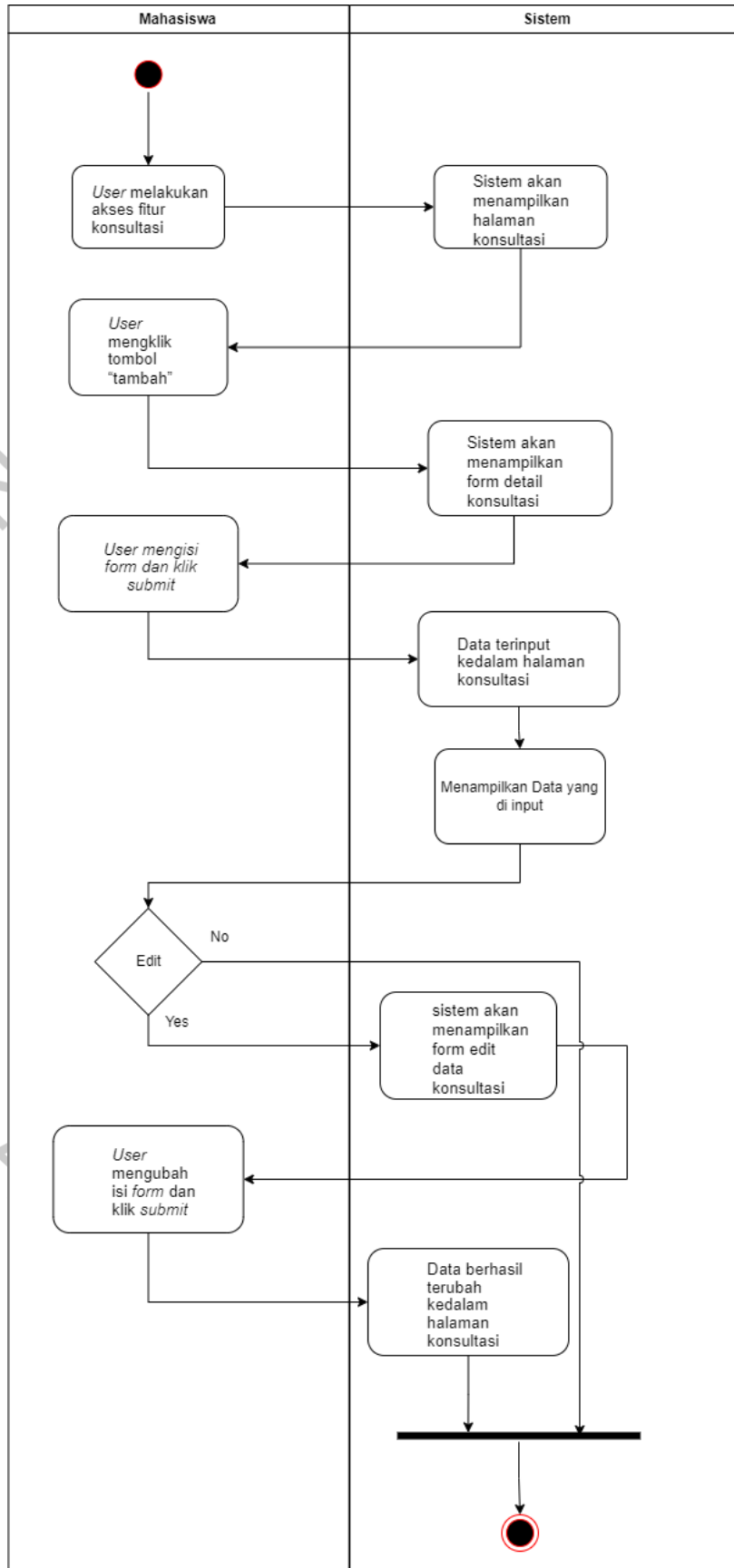


Gambar 4. 2 Activity diagram Login

Gambar di atas adalah Diagram Aktivitas yang menggambarkan proses login pada Aplikasi Advokasi Mahasiswa. Diagram ini mengilustrasikan urutan langkah-langkah yang diambil oleh pengguna saat masuk ke sistem. Proses dimulai dengan pengguna memasukkan data login (email dan kata sandi). Sistem kemudian memeriksa validitas informasi tersebut. Jika informasi benar, pengguna akan berhasil login dan dialihkan ke halaman dashboard mahasiswa. Tetapi, jika informasi tidak sah, sistem akan menolak login dan memberikan notifikasi

kesalahan, sehingga pengguna dapat memperbaiki data *login* sebelum mencoba lagi.





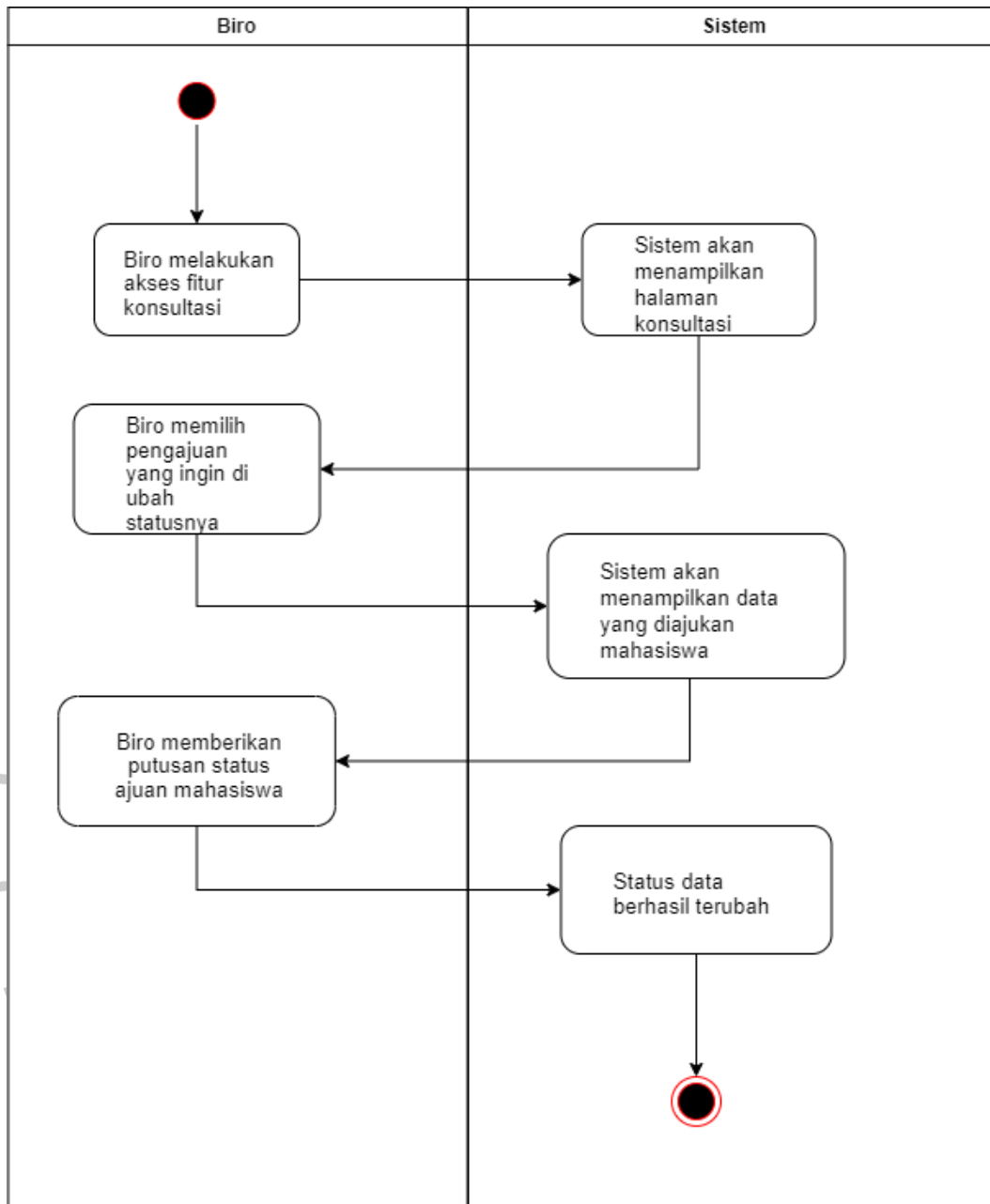
Gambar 4. 3 Activity diagram Konsultasi

Gambar di atas adalah Diagram Aktivitas yang menunjukkan proses mahasiswa untuk mengakses halaman konsultasi pada Aplikasi Advokasi Mahasiswa. Setelah berhasil login, mahasiswa diarahkan ke halaman dashboard. Dari dashboard, mahasiswa dapat memilih opsi untuk mengakses halaman konsultasi, yang memungkinkan mereka untuk melihat atau memulai sesi konsultasi dengan pihak terkait.

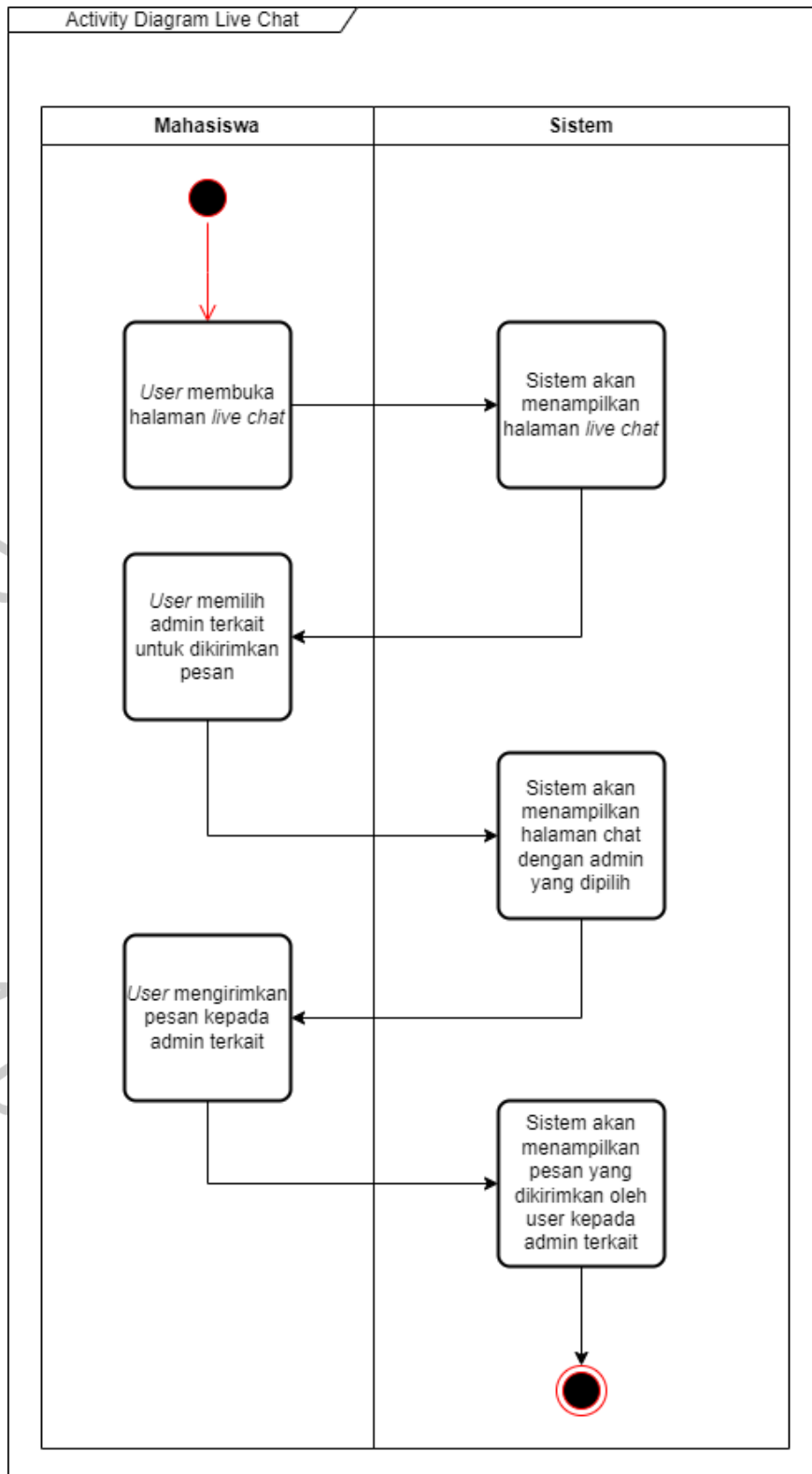
Prosesnya dimulai ketika mahasiswa melakukan akses fitur konsultasi. Pada tahap ini, mahasiswa memilih fitur konsultasi yang tersedia di dashboard aplikasi. Setelah itu, sistem akan menampilkan halaman konsultasi, di mana mahasiswa dapat melihat opsi-opsi terkait sesi konsultasi yang ada.

Selanjutnya, mahasiswa mengklik tombol "tambah" untuk memulai sesi konsultasi baru. Setelah tombol ini diklik, sistem akan menampilkan form detail konsultasi. Form ini berisi berbagai kolom yang perlu diisi oleh mahasiswa terkait informasi konsultasi yang diinginkan.

Mahasiswa kemudian mengisi form tersebut dengan detail informasi yang diperlukan dan mengklik tombol "submit". Setelah mahasiswa mengirimkan form tersebut, Data yang telah dimasukkan akan disimpan di halaman tersebut. konsultasi. Sistem akan menampilkan data yang telah diinput oleh mahasiswa, sehingga mahasiswa dapat melihat informasi konsultasi yang baru saja dibuat.



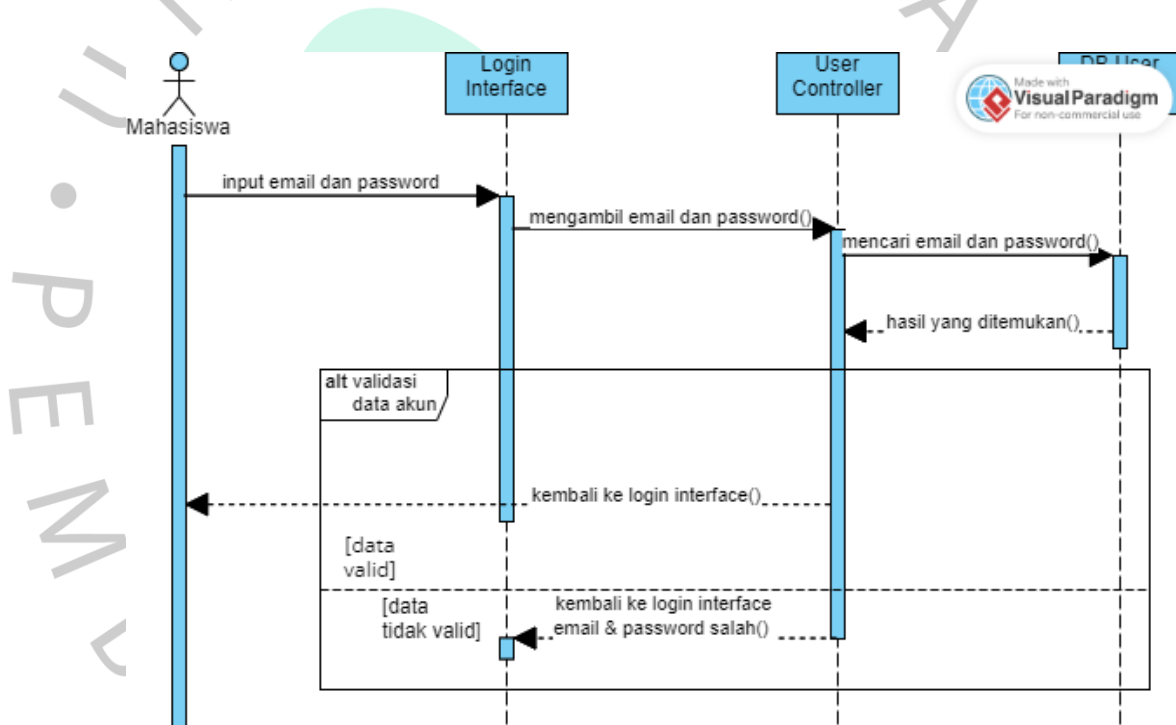
Gambar 4. 4 Acticity Diagram Approval



Gambar 4. 5 Activity diagram Live chat Mahasiswa

#### 4.2.4 Sequence Diagram

Sequence Diagram digunakan untuk mengilustrasikan bagaimana pengguna berinteraksi dengan sistem dalam Aplikasi Advokasi Mahasiswa. Diagram ini menunjukkan urutan pesan yang dikirim antara objek selama proses konsultasi dan *live chat*. Misalnya, dalam fitur konsultasi, Sequence Diagram memperlihatkan langkah-langkah dari mahasiswa mengajukan konsultasi, verifikasi oleh sistem, hingga pemberitahuan status kepada mahasiswa. Untuk fitur *live chat*, diagram ini menampilkan proses mulai dari mahasiswa memulai sesi chat hingga komunikasi real-time dengan pihak terkait.

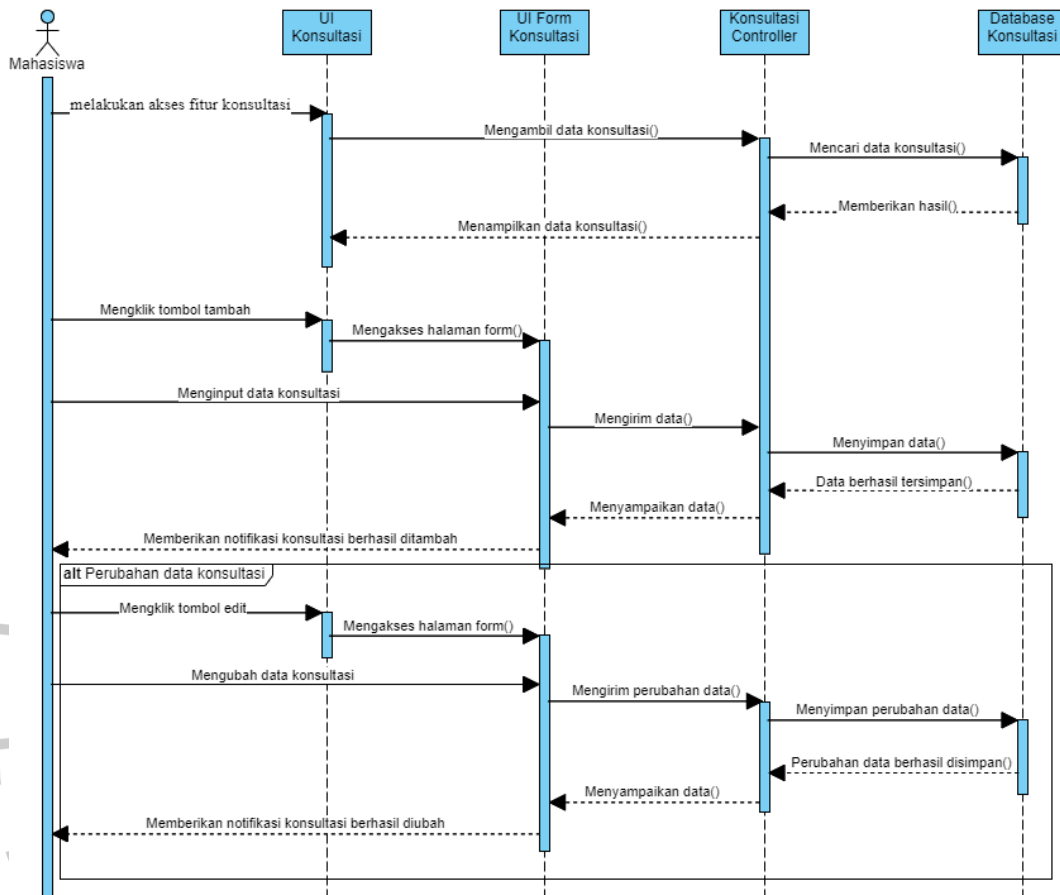


Gambar 4. 6 Sequence Diagram Login

Mahasiswa memasukkan email dan kata sandi di antarmuka *login*. Sistem kemudian mengambil informasi *login* yang telah dimasukkan dan memverifikasinya dengan mencari data yang sesuai di dalam *database*. Jika *database* menemukan data yang cocok, sistem akan mengonfirmasi keabsahan informasi tersebut dan mengarahkan mahasiswa ke halaman *dashboard*. Namun, jika data tidak ditemukan di dalam *database*, sistem akan mengirimkan pesan gagal ke antarmuka *login*, memberitahu mahasiswa bahwa proses *login* tidak berhasil. Diagram ini mencerminkan urutan langkah-langkah interaksi antara mahasiswa dan



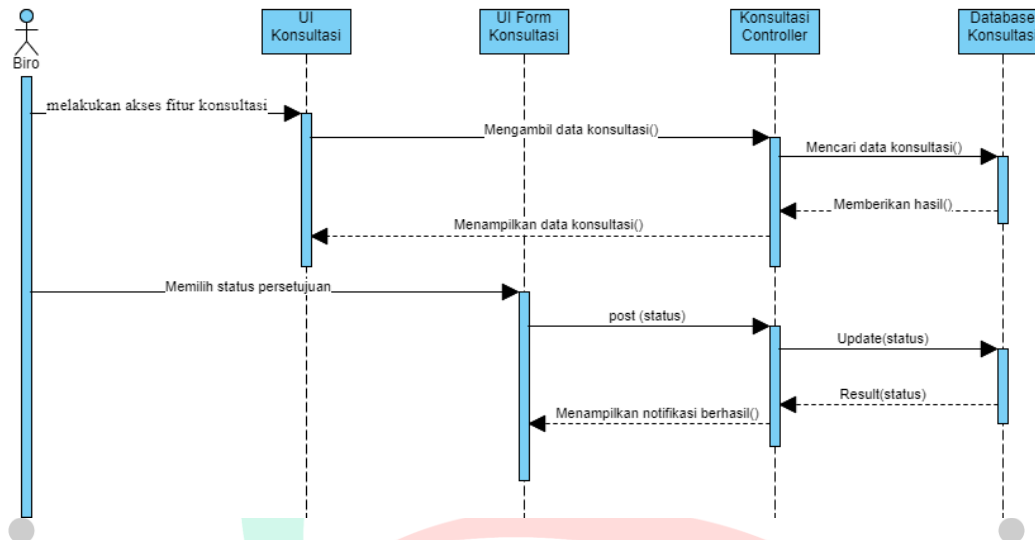
sistem selama proses verifikasi *login* serta bagaimana sistem merespons berdasarkan hasil pencarian data di dalam *database*.



Gambar 4. 7 Sequence Diagram konsultasi

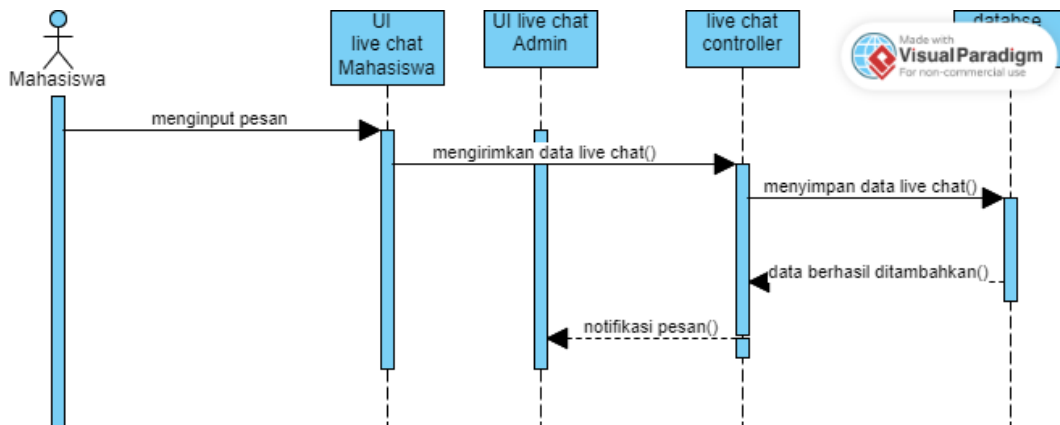
Mahasiswa mengakses halaman konsultasi pada aplikasi advokasi mahasiswa. Antarmuka pengguna mengirim permintaan ke controller, yang mengambil data dari *database* konsultasi dan menampilkannya kembali di antarmuka pengguna, sehingga mahasiswa dapat melihat detail konsultasi mereka dengan mudah. Mahasiswa kemudian memasukkan detail ajuan konsultasi melalui antarmuka aplikasi, yang mengirim data ke controller untuk disimpan di *database*. Setelah data berhasil disimpan, pesan konfirmasi dikirim kembali ke antarmuka pengguna, dan mahasiswa menerima pemberitahuan bahwa ajuan konsultasi mereka telah berhasil disimpan. Jika mahasiswa mengedit data konsultasi, data perubahan dikirim ke controller, disimpan di *database*, dan pesan konfirmasi

dikirim kembali ke antarmuka pengguna, memastikan bahwa informasi yang telah diperbarui tercatat dengan benar.



Gambar 4. 8 Sequence Diagram Approval

Diagram sequence tersebut menggambarkan proses persetujuan (approval) dalam fitur konsultasi oleh aktor "Biro". Proses dimulai ketika Biro mengakses fitur konsultasi pada antarmuka pengguna (UI) Konsultasi. Antarmuka ini kemudian mengambil data konsultasi dengan mengirimkan permintaan ke form UI Konsultasi, yang meminta data dari Konsultasi Controller. Konsultasi Controller mencari data konsultasi yang relevan di dalam *database* konsultasi dan mengirimkan hasilnya kembali ke form UI Konsultasi untuk ditampilkan pada UI Konsultasi. Setelah data ditampilkan, Biro memilih status persetujuan yang sesuai, yang kemudian dikirim dari form UI Konsultasi ke Konsultasi Controller melalui metode "post". Konsultasi Controller memperbarui status persetujuan di dalam *database* konsultasi. Setelah status diperbarui, *database* mengembalikan hasil pembaruan ke Konsultasi Controller, yang kemudian mengirimkan notifikasi ke form UI Konsultasi bahwa pembaruan status berhasil, dan akhirnya notifikasi ini ditampilkan di UI Konsultasi untuk Biro. Proses ini memastikan bahwa semua langkah dari pengambilan data hingga pembaruan status dilakukan secara sistematis dan terkoordinasi.



Gambar 4. 9 Sequence Diagram Live chat Mahasiswa

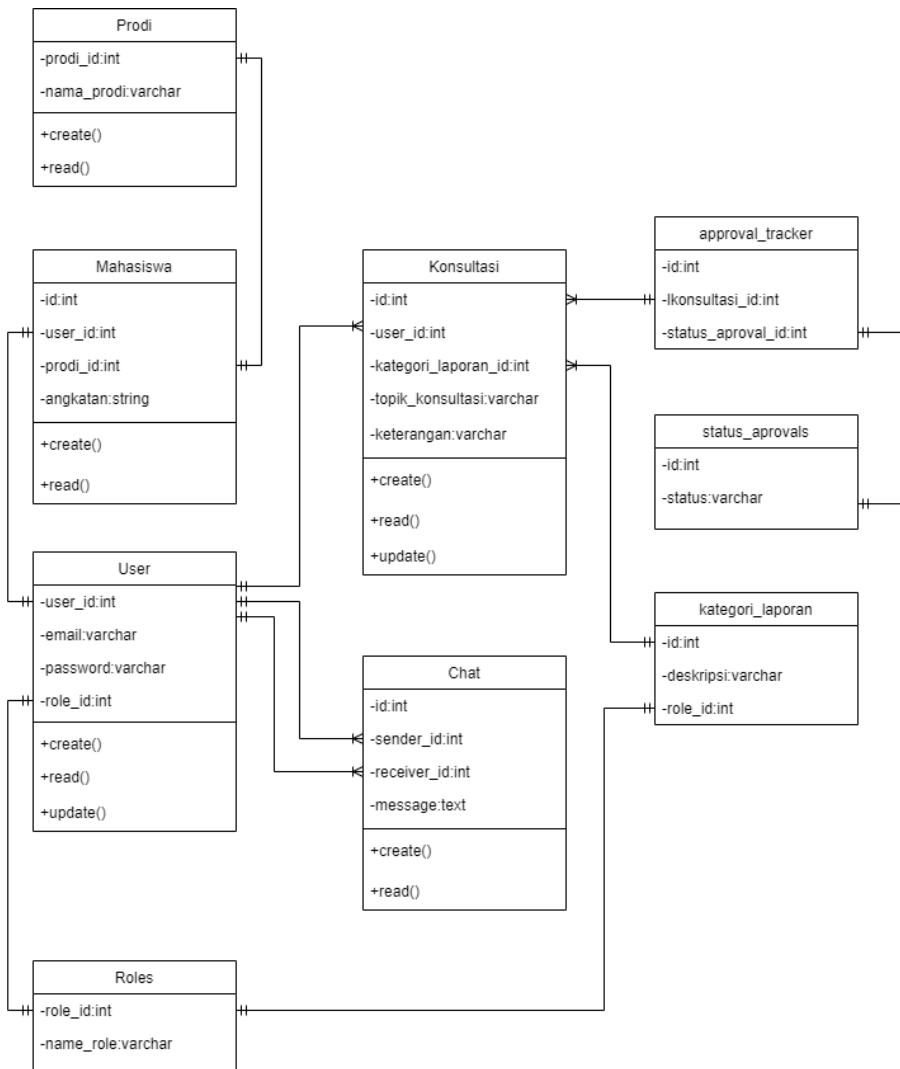
Sequence diagram untuk fitur *live chat* dalam aplikasi advokasi mahasiswa dimulai ketika mahasiswa memasukkan pesan melalui antarmuka *live chat*. Antarmuka *live chat* (UI) mengirimkan data pesan tersebut ke pengontrol (controller) dalam sistem. Pengontrol kemudian bertanggung jawab untuk menyimpan pesan ke dalam *database live chat*. Setelah data pesan berhasil ditambahkan ke dalam *database*, pengontrol mengirim notifikasi ke antarmuka *live chat* admin untuk memberitahu bahwa ada pesan baru yang perlu ditanggapi. Dengan demikian, Diagram ini menampilkan urutan interaksi antara mahasiswa, sistem, dan administrator dalam proses tersebut. *live chat* untuk meningkatkan komunikasi dan responsivitas dalam aplikasi advokasi mahasiswa.

#### 4.2.5 Class Diagram

*Class Diagram* dalam modul *Helpdesk* pada Aplikasi Advokasi Mahasiswa berperan penting dalam merancang struktur sistem untuk fitur konsultasi dan *live chat*. Diagram ini memvisualisasikan kelas-kelas utama beserta atribut dan metode yang relevan.

Diagram ini membantu dalam memahami hubungan antar kelas dan memastikan bahwa semua aspek fungsionalitas terkait konsultasi dan *live chat*

terdefinisi dengan baik, memfasilitasi pengembangan dan pemeliharaan sistem yang efisien.



Gambar 4. 10 Class Diagram Modul Helpdesk

#### 4.2.6 Spesifikasi Database

Spesifikasi *database* adalah dokumen yang memberikan detail lengkap tentang struktur fisik data dalam suatu sistem atau aplikasi. Dokumen ini tidak hanya menjelaskan cara penyimpanan data, tetapi juga menguraikan metode yang digunakan untuk menyimpan data dalam basis data perangkat lunak. Melalui spesifikasi *database*, kita dapat memperoleh

pemahaman yang mendalam mengenai pengaturan dan akses data dalam lingkungan basis data perangkat lunak.

a) *Table Prodi*

*Tabel 4. 7 Spesifikasi Database Table Prodi*

Nama Field	Type	Size	Keterangan
id	int	20	Kode prodi
nama_prodi	varchar	255	Nama program studi

Tabel Program Studi dirancang untuk menyimpan dan menampilkan informasi yang mencatat semua detail terkait Program Studi di sebuah lembaga pendidikan. Di dalam tabel ini, id berperan sebagai kunci utama yang memberikan identifikasi eksklusif untuk setiap entitas Program Studi yang ada. Dengan menggunakan id sebagai pengenal unik, sistem dapat menyimpan dan mengelola data yang terstruktur tentang Program Studi, termasuk deskripsi, kurikulum, dosen pengajar, dan informasi penting lainnya yang relevan untuk administrasi dan operasional lembaga pendidikan tersebut.

b) *Table Mahasiswa*

*Tabel 4. 8 Spesifikasi Database Table Mahasiswa*

Nama Field	Type	Size	Keterangan
nim	Int	20	Kode mahasiswa
<i>user_id</i>	int	20	Kode <i>user</i>
prodi_id	int	20	Kode prodi
angkatan	varchar	255	Angkatan Mahasiswa

Di dalam struktur basis data, terdapat sebuah tabel yang dikenal sebagai "mahasiswa" yang digunakan untuk menyimpan informasi lengkap mengenai setiap mahasiswa di sebuah lembaga pendidikan. *Primary key* pada tabel ini menggunakan id yang menggunakan nomor induk mahasiswa,

yang berperan sebagai identifier unik untuk setiap entitas mahasiswa yang terdaftar. "nim" digunakan untuk membedakan satu mahasiswa dengan mahasiswa lainnya dalam basis data.

Ada dua kunci asing tambahan di tabel "siswa". "user\_id" adalah kunci asing yang menghubungkan tabel "siswa" dan "pengguna" dalam database. Informasi pengguna atau akun yang ditautkan ke setiap siswa ditunjukkan dalam kolom "user\_id". Hal ini memungkinkan untuk mengintegrasikan data siswa dengan data pengguna lain yang tersimpan di dalam sistem.

"prodi\_id" kunci asing ini menghubungkan tabel "mahasiswa" dengan tabel "program studi" (prodi). "prodi\_id" mengidentifikasi program studi yang diikuti oleh mahasiswa tertentu dalam lembaga pendidikan tersebut. Informasi ini membantu dalam pelacakan dan pengelolaan data terkait program studi yang relevan bagi setiap mahasiswa.

Dengan menggunakan kunci utama dan kunci asing ini, tabel "mahasiswa" dapat menyimpan dan mengintegrasikan informasi yang diperlukan secara efisien, memfasilitasi manajemen data yang terstruktur dan terhubung antara mahasiswa, akun pengguna, dan program studi dalam lembaga pendidikan.

c) *Table User*

Tabel 4. 9 Spesifikasi Database Table User

Nama Field	Type	Size	Keterangan
<i>user_id</i>	varchar	10	kode <i>user</i>
email	varchar	255	email <i>user</i>
password	varchar	255	password akun
role_id	int	20	koderole

Dalam desain struktur *database* yang sedang direncanakan, terdapat sebuah tabel yang dinamakan "*user*" yang bertujuan untuk menyimpan informasi lengkap tentang pengguna dalam sistem aplikasi. Tabel ini merupakan bagian krusial dari basis data yang dirancang untuk mengelola data pengguna secara efisien dan terstruktur. Kunci utamanya adalah "*user\_id*", yang berfungsi sebagai pengenal unik yang membedakan setiap entitas pengguna yang terdaftar dalam sistem. Kunci utama ini memungkinkan sistem untuk dengan tepat mengelola operasi pencarian, pembaruan, dan penghapusan data pengguna tanpa ambiguitas.

Selain kunci utama, tabel "*user*" juga mengandung satu kunci asing (*Foreign key*): "*role\_id*" kunci asing ini mengaitkan tabel "*user*" dengan tabel "*role*" dalam basis data. Kolom "*role\_id*" mengacu pada tabel yang berisi informasi tentang peran atau *role* yang dimiliki oleh setiap pengguna dalam sistem. Pengguna dapat memiliki satu atau lebih peran yang memengaruhi hak akses dan fungsi yang mereka miliki dalam aplikasi.

Penggunaan kunci asing ini memfasilitasi integrasi yang efisien antara data pengguna dengan informasi peran atau *role* mereka. Hal ini sangat penting untuk manajemen yang efektif terhadap hak akses pengguna serta pengaturan fungsi dalam sistem. Dengan demikian, tabel "*user*" tidak hanya menyediakan struktur data yang terorganisir dengan baik, tetapi juga mendukung fungsionalitas aplikasi yang kompleks dan memastikan pengelolaan data yang konsisten dan terpadu dalam lingkungan pengembangan perangkat lunak.

d) *Table Roles*

Tabel 4. 10 Spesifikasi Database Table Roles

Nama Field	Type	Size	Keterangan
Role_id	int	20	Kode role
nama_role	varchar	255	Role yang diberikan

*Table roles* didesain untuk menjadi entitas penyimpanan informasi mengenai peran dari setiap aktor dalam sebuah sistem atau aplikasi. Kunci utama Kolom "*role\_id*" dalam tabel ini berfungsi sebagai atribut unik yang digunakan untuk mengidentifikasi setiap entitas secara individu peran yang terdaftar di dalamnya. Dengan menggunakan id sebagai *primary key*, setiap entitas peran dapat dibedakan dengan jelas dalam basis data. Hal ini memungkinkan sistem untuk mengatur dan mengelola peran-peran tersebut secara efisien, sehingga setiap aktor dapat diberikan hak akses dan fungsi yang sesuai dengan perannya dalam sistem tersebut.

e) *Table Konsultasi*

Tabel 4. 11 Spesifikasi Database Table Konsultasi

Nama Field	Type	Size	Keterangan
konsultasi_id	Int	20	Kode konsultasi
<i>user_id</i>	int	20	Kode <i>user</i>
Kategori_laporan_id	int	20	Kode kategori laporan
Topik_konsultasi_id	varchar	255	Angkatan Mahasiswa
keterangan	varchar	255	Keterangan konsultasi

Tabel *database* di atas, yang memiliki *primary key* "*konsultasi\_id*", didesain untuk menyimpan informasi terkait konsultasi dalam sebuah sistem. *Primary key* "*konsultasi\_id*" berperan sebagai kode unik untuk setiap entitas konsultasi yang tercatat dalam tabel. Selain itu, tabel ini juga



mencakup field-field lainnya seperti "user\_id" untuk menyimpan kode user yang terlibat dalam konsultasi, "Kategori\_laporan\_id" untuk menunjukkan kode kategori laporan terkait, "Topik\_konsultasi\_id" yang berfungsi sebagai angkatan mahasiswa yang terlibat dalam konsultasi, dan "keterangan" untuk menyimpan deskripsi atau keterangan tambahan terkait konsultasi tersebut. Setiap field memiliki tipe data yang sesuai, seperti int untuk "user\_id" dan "Kategori\_laporan\_id", serta varchar untuk "Topik\_konsultasi\_id" dan "keterangan", dengan ukuran tertentu untuk masing-masing field guna menampung informasi yang diperlukan. Dengan struktur ini, sistem dapat menyimpan dan mengelola data konsultasi secara efektif, menggunakan "konsultasi\_id" sebagai referensi utama untuk identifikasi dan pengelolaan data konsultasi.

●f) *Table* Kategori Laporan

*Tabel 4. 12 Spesifikasi Database Table Kategori Laporan*

Nama Field	Type	Size	Keterangan
kategori_id	int	20	Kode kategori laporan
deskripsi	varchar	255	Deskripsi laporan
role_id	int	20	Kode role

*Tabel* "kategori\_laporan" adalah bagian dari struktur basis data yang dirancang untuk menyimpan informasi tentang berbagai kategori laporan dalam suatu sistem. *Primary key* dalam tabel ini adalah "kategori\_id," yang berfungsi sebagai identifikasi unik untuk setiap entitas kategori laporan yang terdaftar. *Field* "deskripsi" merupakan kolom varchar dengan panjang maksimum 255 karakter, yang digunakan untuk menyimpan deskripsi atau keterangan lebih lanjut mengenai setiap kategori laporan. Selain itu, terdapat field "role\_id" yang berupa integer dengan panjang 20, berfungsi sebagai kunci asing (*foreign key*) yang mengaitkan kategori laporan dengan entitas peran (role) terkait dalam sistem basis data. Dengan struktur ini, tabel "kategori\_laporan" menyediakan kerangka kerja yang terstruktur untuk

mengorganisir dan mengelola berbagai jenis kategori laporan serta informasi tambahan yang terkait.

g) *Table Approval Tracker*

*Tabel 4. 13 Spesifikasi Database Table Approval Tracker*

Nama Field	Type	Size	Keterangan
approval_id	int	20	Kode approval
laporan_pengaduan_id	Int	20	Kode laporan
Status_approval_id	int	20	Kode status

Tabel "approval tracker" adalah bagian dari struktur basis data yang dirancang untuk memantau proses persetujuan terhadap laporan pengaduan dalam sebuah sistem. Kunci utama dari tabel ini adalah "approval\_id," yang digunakan sebagai pengenal unik untuk setiap entitas. persetujuan yang tercatat. Tabel ini mencatat setiap langkah dalam proses persetujuan, mulai dari permintaan persetujuan hingga status akhirnya. Informasi yang disimpan mencakup "laporan\_pengaduan\_id", yang menyimpan kode unik laporan pengaduan yang sedang diajukan untuk persetujuan, serta "status\_approval\_id" yang menunjukkan status terkini dari proses persetujuan tersebut. Dengan demikian, tabel ini memberikan transparansi dan pengelolaan yang efisien terhadap semua persetujuan yang terjadi dalam sistem.

h) *Table Status Approval*

*Tabel 4. 14 Spesifikasi Database Table Status Approval*

Nama Field	Type	Size	Keterangan
approval_id	int	20	Kode approval
Status	varchar	255	Status persetujuan laporan

Tabel "status\_approval" adalah bagian dari basis data yang digunakan untuk melacak status persetujuan laporan. Tabel ini terdiri dari

tiga kolom utama: "approval\_id" sebagai *primary key* untuk identifikasi unik, "laporan\_pengaduan\_id" sebagai kode identifikasi laporan, dan "Status" untuk menunjukkan status persetujuan laporan. Kolom "approval\_id" digunakan sebagai kunci utama yang unik untuk setiap entitas persetujuan dalam tabel. Ini memungkinkan sistem untuk mengelola dengan efisien setiap proses persetujuan laporan, sementara kolom "Status" menyediakan informasi langsung mengenai status saat ini dari proses persetujuan tersebut. Tabel ini penting untuk memantau dan mengelola alur kerja persetujuan secara efektif dalam aplikasi atau sistem manajemen laporan.

i) *Table Chat*

Tabel 4. 15 Spesifikasi Database Table Chat

Nama Field	Type	Size	Keterangan
chat_id	Int	20	Kode chat
sender_id	int	20	Kode sender
receiver_id	int	20	Kode receiver
message	text		Pesan yang akan dikirimkan

Tabel "chat" adalah struktur basis data yang digunakan untuk menyimpan percakapan langsung antara pengguna aplikasi advokasi mahasiswa. *Primary key* dari tabel ini adalah "chat\_id," yang berperan sebagai identifikasi unik untuk setiap entitas percakapan. Selain itu, terdapat kolom "sender\_id" dan "receiver\_id" yang menyimpan kode pengirim dan penerima pesan. Pesan yang dikirimkan disimpan dalam kolom "message" dengan tipe data teks. *Database* ini memungkinkan pengguna untuk berkomunikasi secara langsung dalam sistem, mendukung interaksi real-time antara mahasiswa dan admin atau pihak terkait lainnya.

### **4.3 Perancangan Antar Muka Pengguna**

Antarmuka pengguna (UI) adalah bagian penting dari koneksi langsung antara pengguna dan sistem. Karena menguraikan bagaimana sistem akan secara efektif terlibat dan berkomunikasi dengan pengguna, langkah desain UI ini sangat penting. Tujuan utama dari proses desain UI, yang dianggap sebagai bentuk seni, adalah untuk menyediakan antarmuka pengguna yang intuitif dan menginspirasi. Tujuan desainer pada tahap ini adalah untuk membuat antarmuka pengguna dasbor siswa untuk aplikasi advokasi siswa. Hal ini dilakukan sebagai persiapan untuk melibatkan pengguna pada tahap tinjauan dan evaluasi selanjutnya.

#### **4.3.1 Prototype *User interface* Tahap Awal**

Halaman ini menampilkan visualisasi dari awal implementasi fitur konsultasi dan *live chat* untuk mahasiswa. Desainnya difokuskan untuk mempermudah pengguna, khususnya mahasiswa, dalam melakukan konsultasi dan berinteraksi langsung dengan admin. Tujuannya adalah memberikan akses yang mudah dipahami dan efisien bagi mahasiswa saat menggunakan layanan konsultasi yang tersedia. Ini mencakup fitur yang memudahkan mahasiswa untuk mengajukan pertanyaan, menerima respons secara langsung, serta mengelola proses konsultasi dengan admin dengan lebih efektif.

- Home
- Dashboard
- Pelaporan
  - Tambah Laporan
- Pusat Informasi
  - Pusinfo
- Histori
  - Histori Aduan
- Tracer Penyelesaian
  - Tracer Penyelesaian
- Konsultasi
  - Konsultasi Laporan
- Live Chat
- ACCOUNT
  - Manajemen Akun

Logout

Hi, "Name"  
Mahasiswa



List Detail

Type to search...

Filter

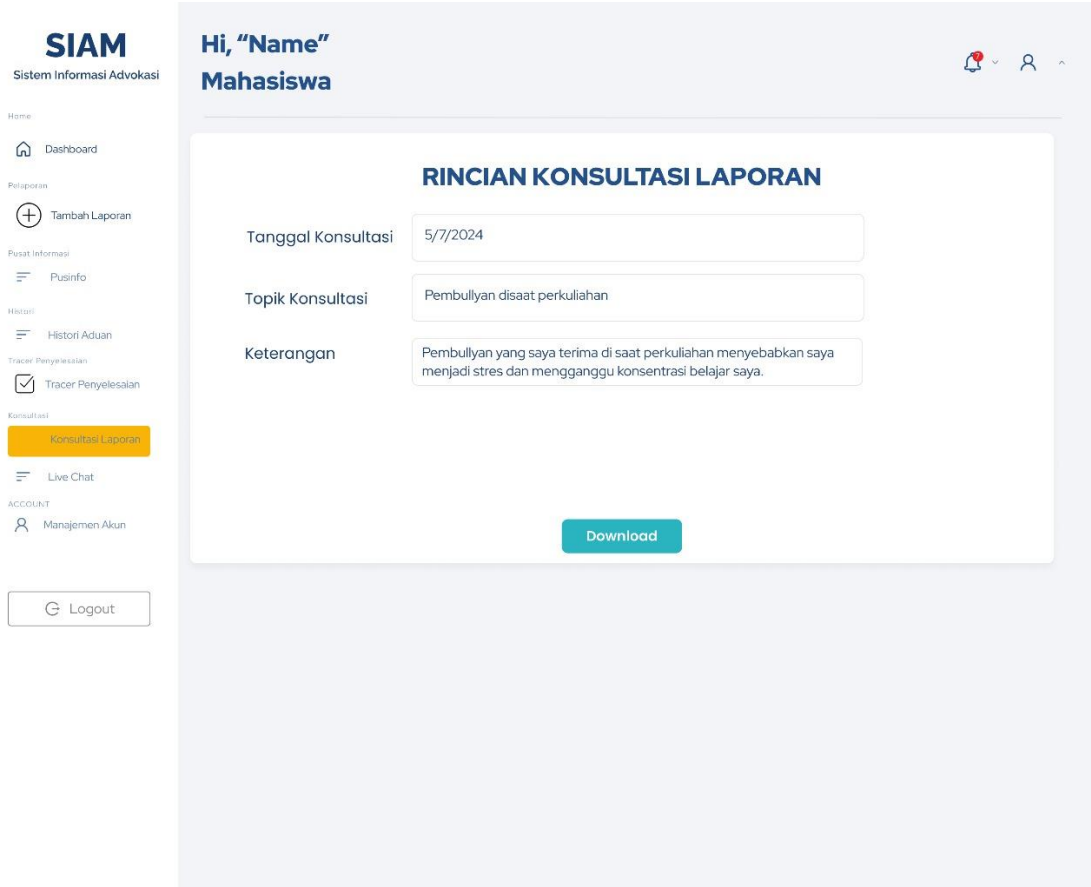
Tambah

Tanggal Konsultasi	Topik Konsultasi	PIC	Konselor	Status	Aksi
5/7/2024	Pembullying disaat perkuliahan	KASTRAD BEM UPJ	Sdr. Iffat Dwi Ananto	On Progress	

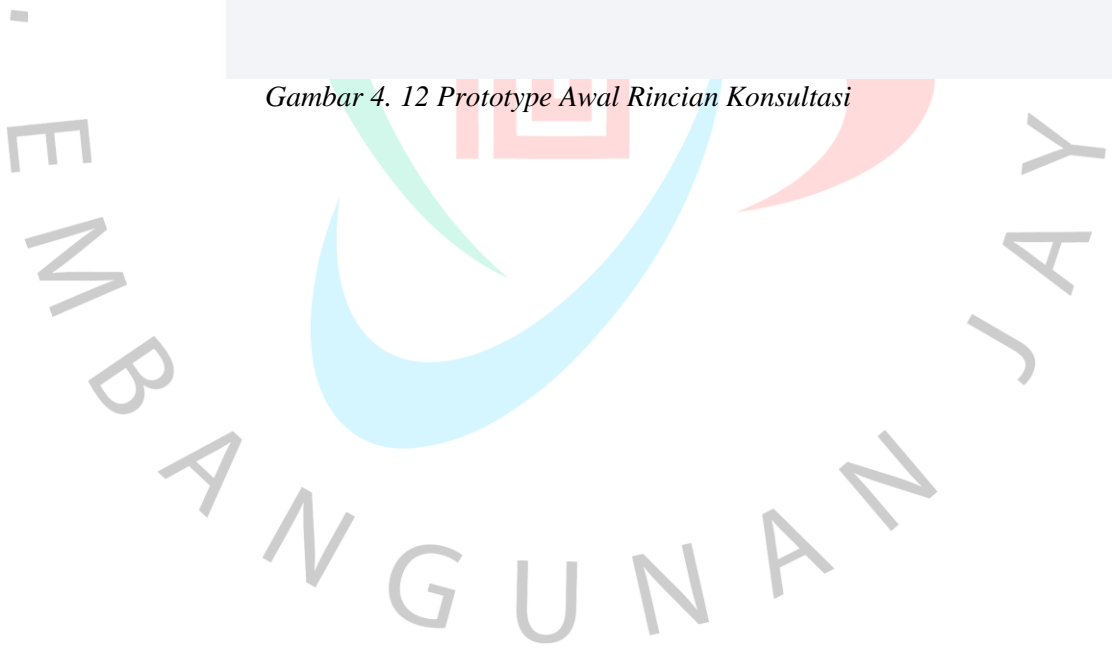
1

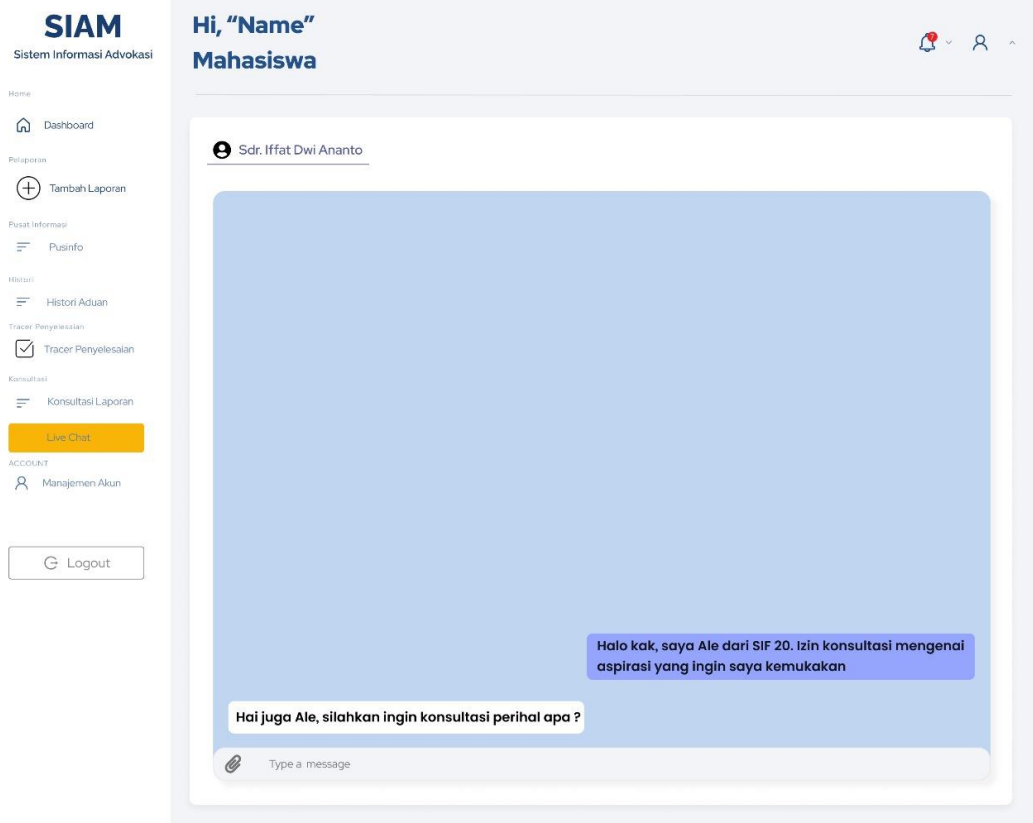
Gambar 4. 11 Prototype Tahap Awal





Gambar 4. 12 Prototype Awal Rincian Konsultasi

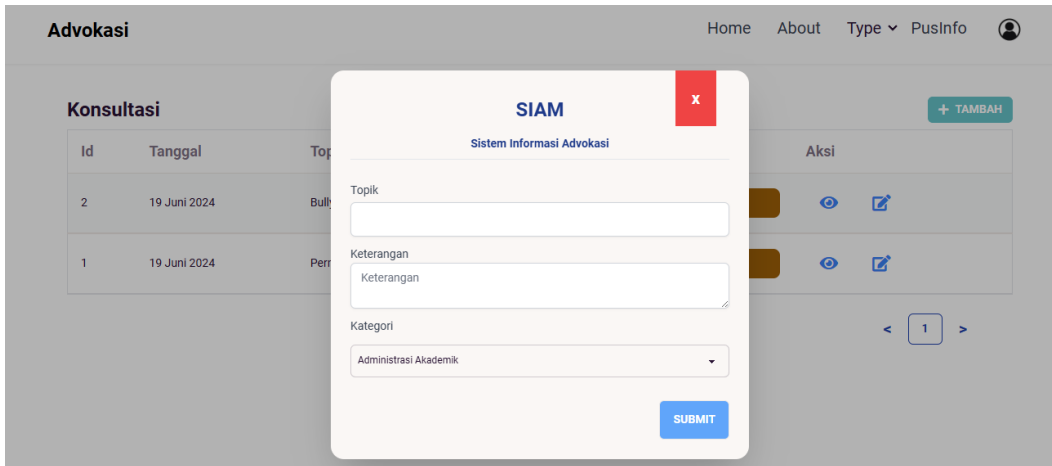




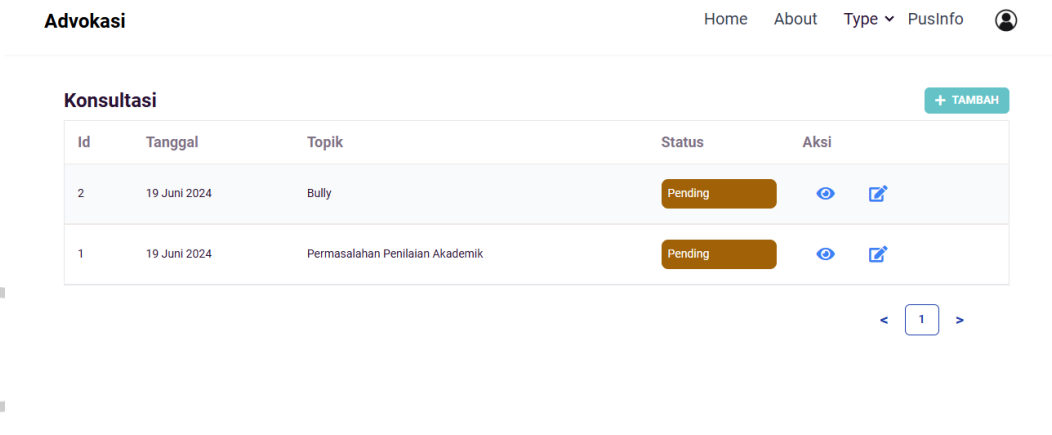
Gambar 4. 13 Prototype Awal Live Chat

#### 4.3.2 Prototype User interface Tahap Akhir

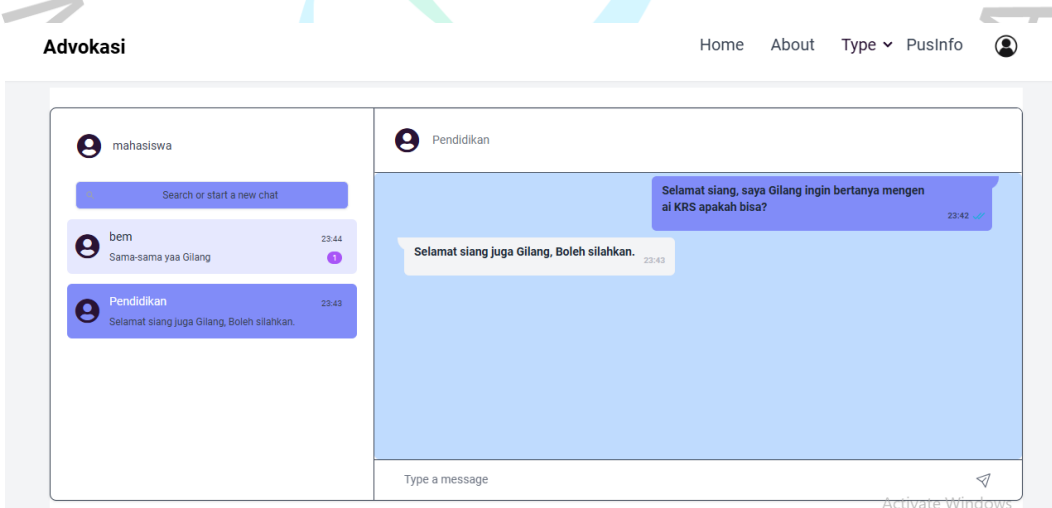
Prototype *User interface* tahap akhir ini merupakan hasil dari evaluasi oleh pengguna. Fitur konsultasi dan *live chat* dalam aplikasi advokasi mahasiswa telah disesuaikan dengan masukan pengguna untuk memenuhi kebutuhan dan preferensi mereka. Perubahan yang diminta telah diimplementasikan untuk meningkatkan kemudahan penggunaan, kejelasan antarmuka, dan responsivitas sistem. Tahap ini menandai kesuksesan dalam menyempurnakan desain antarmuka pengguna berdasarkan umpan balik pengguna langsung.



Gambar 4. 14 Form Konsultasi



Gambar 4. 15 Rincian Konsultasi



Gambar 4. 16 Halaman Live Chat



#### 4.4 Perancangan Implementasi

Pada tahap implementasi ini, fokus utamanya adalah mengembangkan dan mengintegrasikan modul *Helpdesk* yang mencakup fitur konsultasi dan *live chat* dalam aplikasi advokasi mahasiswa di Institusi Pendidikan XYZ. Langkah-langkah ini mencakup penjadwalan implementasi dan penyusunan skenario pengujian aplikasi untuk memastikan keefektifan dan keamanan modul tersebut dalam sistem yang ada. Tahap ini menjadi pondasi yang krusial untuk memastikan bahwa pengguna dapat mengakses dan menggunakan fitur konsultasi dan *live chat* dengan baik dalam konteks advokasi mahasiswa di institusi tersebut.

##### 4.4.1 Jadwal Implementasi

Tabel 4. 16 Gantt Chart Pelaksanaan

Kegiatan	Bulan 1				Bulan 2				Bulan 3			
	1	2	3	4	1	2	3	4	1	2	3	4
Tahap <i>Requirements Planning</i>												
Analisis Kebutuhan	■											
Pengumpulan data	■											
Tahap Desain Perangkat Lunak												
Perancangan UML		■										
Perancangan UI/UX			■									
Pembuatan <i>prototype</i>			■									
Tahap Rapid Construction												
Pemrograman sistem			■	■	■	■	■					
<i>Feedback user</i>			■	■	■	■						
Perbaikan			■	■	■	■						
Tahap Implementasi												
<i>Black Box Testing</i>							■	■	■			
<i>Deployment</i>									■			

Diagram Gantt yang dapat dilihat menggambarkan urutan langkah-langkah penelitian menggunakan strategi pengembangan aplikasi Rapid Application Development (RAD), yang berfokus pada pengembangan yang cepat dan berulang. Fase perencanaan, perancangan, pengembangan, dan implementasi aplikasi semuanya termasuk dalam proses tersebut. Diagram ini memberikan petunjuk yang komprehensif tentang bagaimana setiap tahap proyek harus diatur dalam jadwal, menjamin bahwa proyek berjalan dengan lancar dan memenuhi semua tenggat waktu.

Tabel ini menyajikan urutan kegiatan dalam proses pengembangan aplikasi berdasarkan metode *Rapid Application Development (RAD)* selama tiga bulan. Bulan pertama dimulai dengan tahap perencanaan kebutuhan, analisis kebutuhan, dan pengumpulan data. Bulan kedua fokus pada perancangan perangkat lunak, mencakup pembuatan desain UML dan UI/UX, serta pengembangan prototype aplikasi. Bulan ketiga merupakan tahap konstruksi cepat yang meliputi pemrograman sistem, umpan balik dari pengguna, dan perbaikan berdasarkan umpan balik tersebut. Tahap akhir adalah implementasi dengan uji coba *Black Box Testing* sebelum dilakukan deployment aplikasi.

#### **4.4.2 Skenario Pengujian Aplikasi**

Skenario pengujian aplikasi memainkan peran sentral dalam mengatur proses pengujian, sehingga memungkinkan pelaksanaannya menjadi lebih optimal dan terorganisir. Pengujian ini sangat penting karena menggunakan skenario terstruktur yang terbagi berdasarkan fitur aplikasi serta tipe pengujian, yaitu pengujian positif dan negatif.

Dalam pengujian positif, aplikasi diuji untuk memastikan bahwa semua fungsinya bekerja dengan baik dalam kondisi normal. Sementara itu, pengujian negatif bertujuan untuk mengevaluasi bagaimana aplikasi merespons kesalahan atau pelanggaran prosedur dalam kondisi yang tidak seharusnya terjadi. Dengan skenario pengujian negatif, tim pengembang dapat menemukan dan memperbaiki masalah yang mungkin muncul, sehingga meningkatkan keandalan dan responsivitas aplikasi terhadap situasi yang tidak terduga.

Memastikan setiap modul aplikasi berjalan sesuai rencana dan menghasilkan output yang tepat adalah tujuan utama dari pengujian ini. Oleh karena itu, menerapkan skenario pengujian yang terstruktur dan komprehensif sangat penting untuk memverifikasi kinerja program sebelum disebarakan ke pengguna akhir, terutama siswa yang akan menggunakan aplikasi advokasi siswa ini.

Tabel 4. 17 Tabel Pengujian Login

No Test	Skenario	Data Uji	Kondisi Test	Deskripsi Output	Hasil
TC001	Login	Username dan password sesuai (L001)	Positif	Melakukan masuk dengan menggunakan nama pengguna dan kata sandi yang tepat	Dapat login dan diberi akses ke dashboard mahasiswa
TC002		Username sesuai dan password salah (L002)	Negatif	Melakukan masuk dengan menggunakan nama pengguna dan kata sandi yang tidak valid	Tidak dapat login dan tidak diberi akses ke dashboard mahasiswa

Tabel 4. 18 Tabel Pengujian Konsultasi

No Test	Skenario	Data Uji	Kondisi Test	Deskripsi Output	Hasil
TC003	Mengajukan konsultasi dengan data lengkap (K001)	Topik: Akademik, Keterangan: Masalah skripsi	Positif	Mahasiswa mengajukan konsultasi dengan data lengkap	Konsultasi tersimpan dan ditampilkan di halaman konsultasi
TC004	Mengajukan konsultasi tanpa topik (K002)	Topik: -, Keterangan: Masalah skripsi	Negatif	Mahasiswa mengajukan konsultasi tanpa topik	Konsultasi tidak tersimpan dan diberikan pesan error
TC005	Mengajukan konsultasi tanpa keterangan (K003)	Topik: Akademik, Keterangan: -	Negatif	Mahasiswa mengajukan konsultasi tanpa keterangan	Konsultasi tidak tersimpan dan diberikan pesan error