

## BAB 3 PELAKSANAAN KERJA MAGANG

### 3.1 Kedudukan dan Organisasi

Magang ini dilakukan di divisi *Mobility* dengan posisi sebagai *Developer Intern*, yang bertanggung jawab untuk membantu dalam aktivitas riset dan pengembangan proyek-proyek teknologi informasi yang sedang dilakukan oleh mitra perusahaan. Proses magang disupervisi oleh Perdana Bagas Tirta Kumbara, yang mengampu rekan-rekan magang *Developer Intern*. Tugas supervisi melibatkan pemantauan, memberikan arahan dalam mengatasi kendala dan kekhawatiran terkait proyek yang diampu, serta menyediakan informasi yang diperlukan terkait pengembangan proyek.

Penentuan proyek yang diampu selama pelaksanaan magang telah dilakukan pada awal periode magang, di mana peserta magang diberikan kesempatan untuk berdiskusi mengenai proyek yang akan dihadapi, termasuk kesulitan yang mungkin muncul atau batas waktu pengerjaan yang ditetapkan. Setelahnya, proyek dilaksanakan secara individu dengan koordinasi sesuai dengan kebutuhan yang telah dibahas bersama *supervisor*. Setiap minggu, diadakan rapat bersama *supervisor* dan anggota tim *Developer* lainnya sebagai sarana pemantauan kemajuan dan penentuan target proyek berikutnya. Selain rapat utama tersebut, peserta magang juga melakukan rapat harian dengan perwakilan *supervisor* untuk melaporkan kemajuan proyek dan membahas kendala serta kebutuhan dalam pengerjaan proyek. Jika diperlukan, koordinasi juga dilakukan dengan anggota *Developer* lainnya.

### 3.2 Tugas yang Dilakukan

Setelah proses penentuan, proyek yang menjadi fokus tugas magang adalah *Live Chat*. Proyek ini merupakan inisiatif riset dan pengembangan PT Helios Informatika Nusantara yang berkaitan dengan penyediaan layanan pelanggan berbasis obrolan online untuk mendigitalisasi sistem solusi TI. Fitur yang diinginkan untuk *Live Chat Widget* adalah sistem obrolan online yang beroperasi dalam satu platform. Pengembangan fitur ini terfokus pada *platform web*. Sistem *Live Chat* dikembangkan sebagai fondasi untuk PT Helios Informatika Nusantara yang menekankan interaksi intensif antar pengguna, khususnya dalam menanggapi pertanyaan-pertanyaan mengenai kebutuhan dasar.

Selain pengembangan fitur *Live Chat*, tugas magang juga melibatkan penelitian awal terkait framework yang diperlukan untuk menjalankan proyek tersebut. Karena fitur *Live Chat* akan diintegrasikan sebagai pengembangan dari proyek yang sedang berjalan, keberfungsian fitur yang sudah ada harus dipastikan agar dapat berinteraksi secara sinergis satu sama lain.

### 3.3 Uraian Pelaksanaan Magang

Magang dilaksanakan selama 20 minggu kerja. Pada saat penyerahan laporan ini, proses magang masih berlangsung hingga minggu ke-19. Rincian aktivitas selama proses magang dijelaskan seperti yang tertera pada Tabel 3.1 berikut.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1-2	Proses <i>onboarding</i> dilakukan oleh mitra, dengan penunjukan seorang mentor selama periode <i>onboarding</i> . Mentor ini bertanggung jawab untuk memperkenalkan lingkungan kerja, peralatan, dan program yang akan digunakan selama proses magang berlangsung.
3-4	Pengenalan dilakukan terhadap proyek yang akan diampu, termasuk riset mengenai <i>framework</i> yang diperlukan untuk proyek tersebut. Pembelajaran mandiri yang terarah dilakukan dengan fokus pada proyek yang sedang dikembangkan, dimulai dengan menyusun daftar <i>framework</i> yang diperlukan untuk proyek tersebut.
4-5	Eksplorasi mandiri dilakukan untuk mengeksplorasi <i>framework</i> yang dibutuhkan oleh proyek tersebut. Selain itu, dilakukan koordinasi dengan rekan yang sebelumnya telah mengampu proyek serupa untuk memahami <i>tools</i> apa saja yang perlu dipersiapkan.
6-7	Melakukan eksplorasi mandiri ketika menggunakan <i>framework</i> yang dibutuhkan proyek tersebut. Eksplorasi dilakukan dengan membuat gambaran kasus sederhana terlebih dahulu untuk memastikan apa yang dieksplorasi berhasil.

Minggu Ke -	Pekerjaan yang dilakukan
8	Membuat tampilan sederhana <i>Live Chat</i> untuk menjadi tempat mengobrol <i>one to one</i> secara <i>online</i> sementara
9-11	Membuat beberapa fungsi untuk mendapatkan data yang diinput kemudian diproses oleh <i>framework backend</i> yang dibutuhkan proyek tersebut.
12-13	Membuat fungsi data yang diinput bisa tersimpan di <i>database</i> dan bisa ditampilkan kembali data yang diperlukan, serta membuat tampilan data yang ditampilkan dinamis.
14-15	Mengeplorasi mandiri bagaimana mengembangkan fungsi agar <i>Live Chat</i> yang dilakukan bisa <i>one to many</i> tidak hanya <i>one to one</i> serta memperbaiki tampilan agar lebih <i>responsive</i> .
15-17	Membuat tampilan menjadi dua, yaitu tampilan <i>Live Chat</i> untuk pengguna melakukan permintaan obrolan online dengan <i>agent</i> dan tampilan <i>dashboard agent</i> untuk menerima permintaan obrolan yang masuk dari pengguna agar <i>agent</i> bisa menentukan dengan pasti pengguna akan mengobrol dengan <i>agent</i> .
18-19	Memperbaiki fungsi yang sudah dibuat serta mengembangkan <i>database</i> dan fungsi-fungsi baru untuk membuat <i>real-time</i> ketika melakukan obrolan online serta memperbaiki dan mengembangkan tampilan agar memudahkan <i>agent</i> atau pengguna dalam melakukan obrolan <i>online</i> .

### 3.3.1 Peralatan Pendukung Aktivitas Magang

Untuk menjalankan proses magang ini, digunakan beberapa peralatan yang dapat dikategorikan sebagai perangkat keras, seperti laptop, dan perangkat lunak berupa aplikasi. Rincian perangkat keras yang digunakan adalah sebagai berikut.

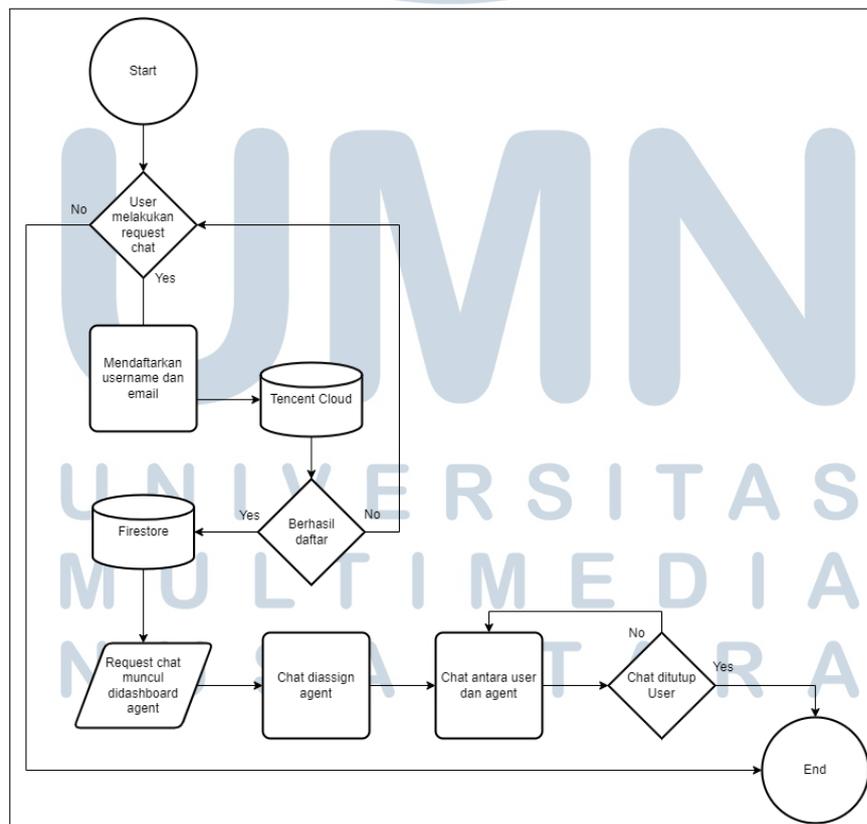
- Dell Latitude 7490
- RAM: 16GB
- Processor: Intel(R) Core(TM) i7-8650U CPU @ 1.90GHz, 2112 Mhz, 4 Core(s), 8 Logical Processor(s)
- Graphics Processor: Intel® UHD Graphics

Di sisi lain, rincian perangkat lunak yang digunakan dalam pelaksanaan proyek magang adalah sebagai berikut.

- Node.js v20.9.0
- Next.js v13.5.3
- Chakra UI v2.8.1
- Firebase v10.4.0
- Tencent Cloud Chat v3.2.0

### 3.3.2 Perancangan Sistem

Dalam pengembangan sistem *live chat*, diperlukan visualisasi alur untuk memahami bagaimana sistem akan berjalan. Visualisasi alur ini bermanfaat untuk menentukan prosedur kerja sistem secara keseluruhan. Oleh karena itu, sebuah diagram alur koneksi untuk *live chat* yang sedang dikembangkan dibuat melalui ilustrasi diagram pada 3.1 berikut.



Gambar 3.1. Flowchart Sistem Live Chat yang dikembangkan dalam proyek

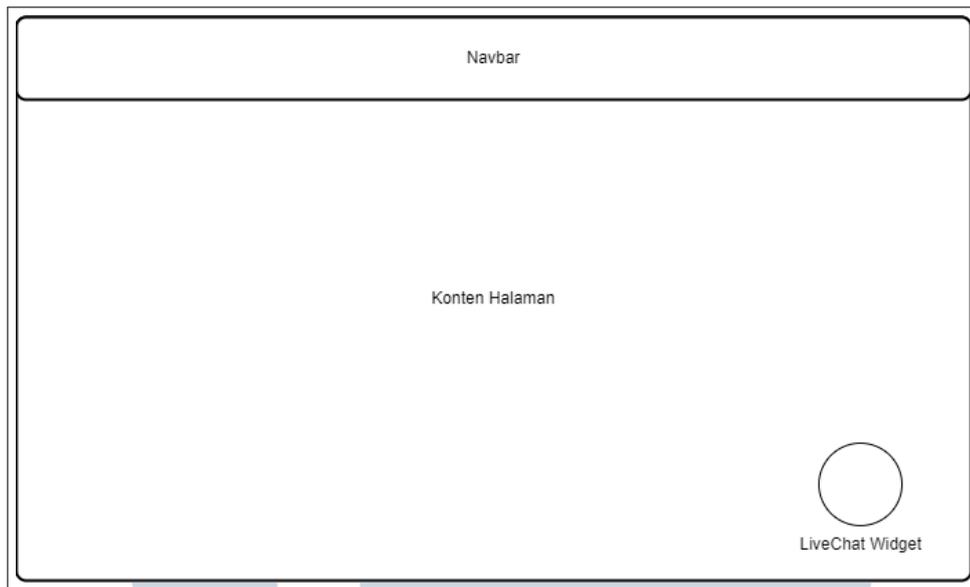
Berdasarkan gambar tersebut, bagi pengguna atau pengunjung situs web yang ingin menggunakan layanan *Live Chat*, dibutuhkan pengisian *username* dan *email* sebagai persyaratan untuk mengirimkan permintaan atau request chat. Setelah pengguna mengisi *username* dan *email*, informasi pengguna akan didaftarkan terlebih dahulu di *Tencent Cloud Chat* sebelum akhirnya disimpan di *Firestore*. *Firestore* dalam konteks ini berperan sebagai tempat penyimpanan kedua yang digunakan untuk memanggil data dari fungsi-fungsi tertentu guna membantu pengolahan data atau menjadi pemicu aksi tertentu.

Setelah pengguna berhasil mengirimkan permintaan *chat* kepada *agent*, pengguna akan dimasukkan ke dalam antrian untuk menunggu agar seorang *agent* menerima permintaan *chat* tersebut. Setelah *agent* menerima permintaan *chat* dari pengguna, sebuah ruang *chat* akan dibuat di *backend* yang mencatat masing-masing *username*, menunjukkan bahwa *agent* dan pengguna telah berada dalam ruang *chat* yang sama, dan memungkinkan komunikasi antara *agent* dan pengguna dimulai.

### 3.3.3 Pengembangan Wireframe

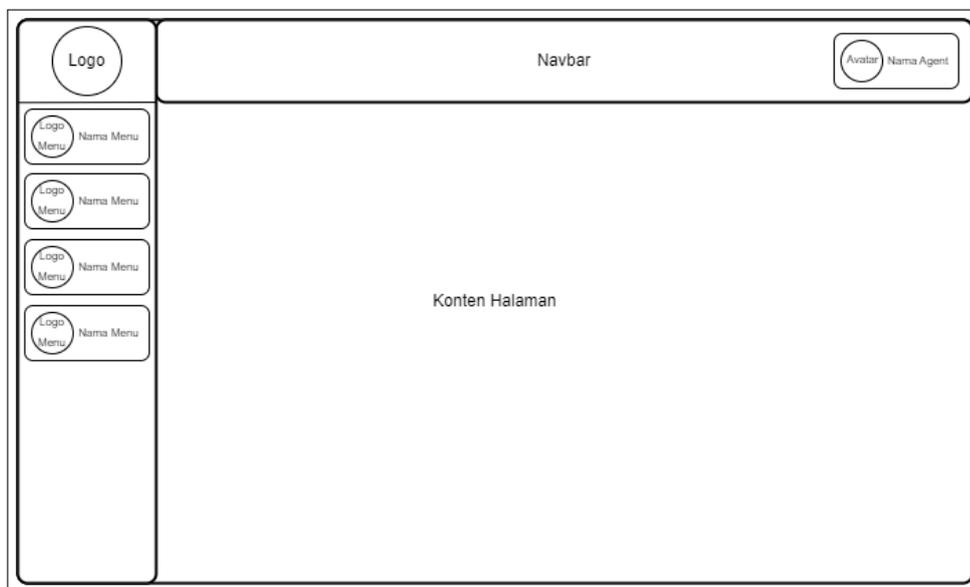
Setelah alur *Live Chat* diketahui, dilakukan briefing terkait dengan cara proyek tersebut ingin dibuat bersama rekan-rekan yang sebelumnya telah menjadi pengampu proyek tersebut. Pada sesi tersebut, rekan tersebut memberikan informasi mengenai cara pengembangan proyek dan apa yang telah saya kembangkan. Komponen yang diperlukan untuk tampilan pengguna yang terdapat dalam target *Live Chat* saat ini dijelaskan dalam Gambar 3.2 berikut.

U I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 3.2. Tampilan Wireframe User Sistem Live Chat

Diskusi pada sesi briefing juga menentukan komponen-komponen tampilan *dashboard agent* yang terdapat dalam target *Live Chat* saat ini, seperti yang diilustrasikan dalam Gambar 3.3 berikut, yang memberikan gambaran visual terhadap antarmuka pengguna yang diimplementasikan dalam sistem.



Gambar 3.3. Tampilan Wireframe Agent Sistem Live Chat

Dengan mempertimbangkan informasi dari kedua gambar tersebut, serta dasar proyek yang telah dikembangkan sebelumnya, sebuah *wireframe* yang menggambarkan bentuk *Live Chat* dibuat sesuai dengan alur yang divisualisasikan

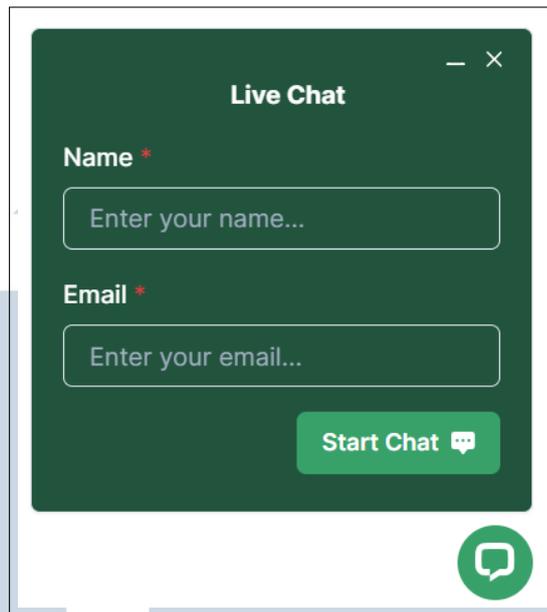
pada Gambar 3.1. Hasil akhir yang dihasilkan dapat dilihat pada Gambar 3.4 berikut.



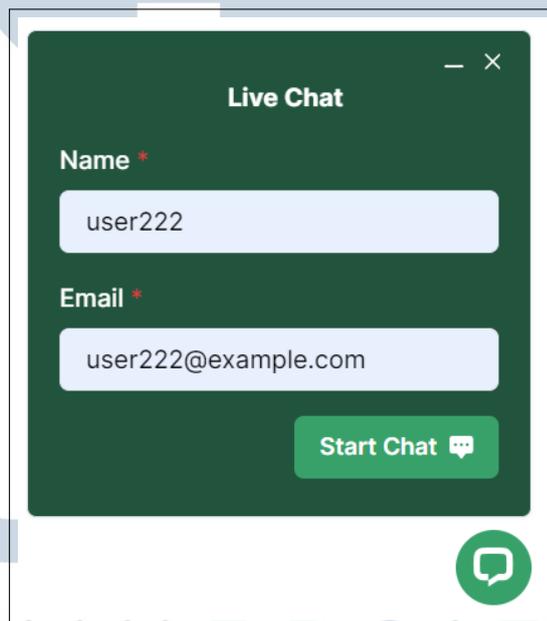
Gambar 3.4. Tampilan *Flow Wireframe* Sistem *Live Chat*

### 3.3.4 Hasil Implementasi

Pada minggu kedelapan, pengembangan proyek dilanjutkan secara langsung. Proyek telah memasuki tahap pengembangan, yang mengharuskan eksplorasi komprehensif terhadap komponen-komponen yang ada dalam proyek, dengan tujuan memahami struktur proyek dan melakukan integrasi sistem *Live Chat*. Pengguna diharapkan untuk mendaftarkan terlebih dahulu agar terhubung dengan server *Tencent Cloud Chat* dan kemudian masuk ke dalam antrian atau menunggu *assign* dari *agent*.

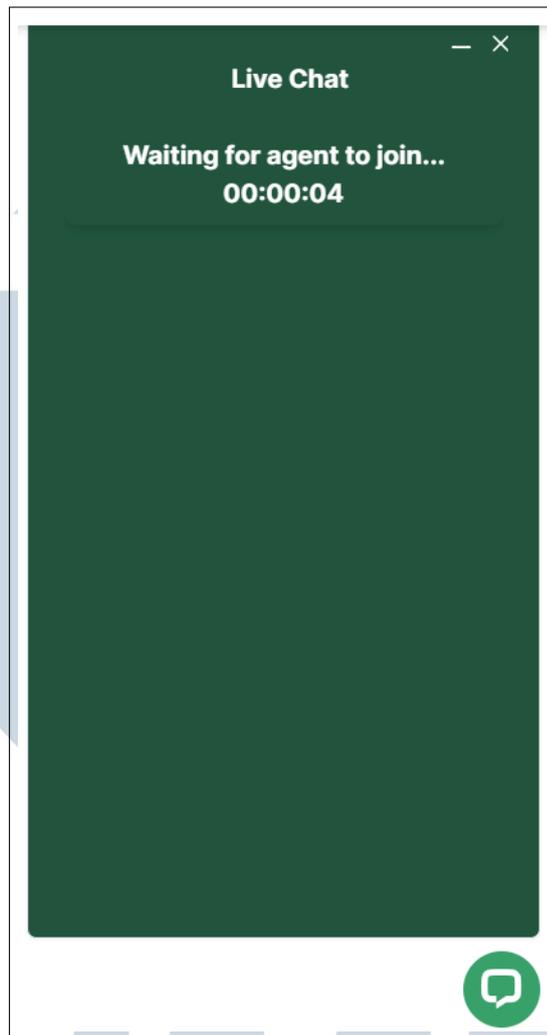


Gambar 3.5. Tampilan *Live Chat* halaman *User*



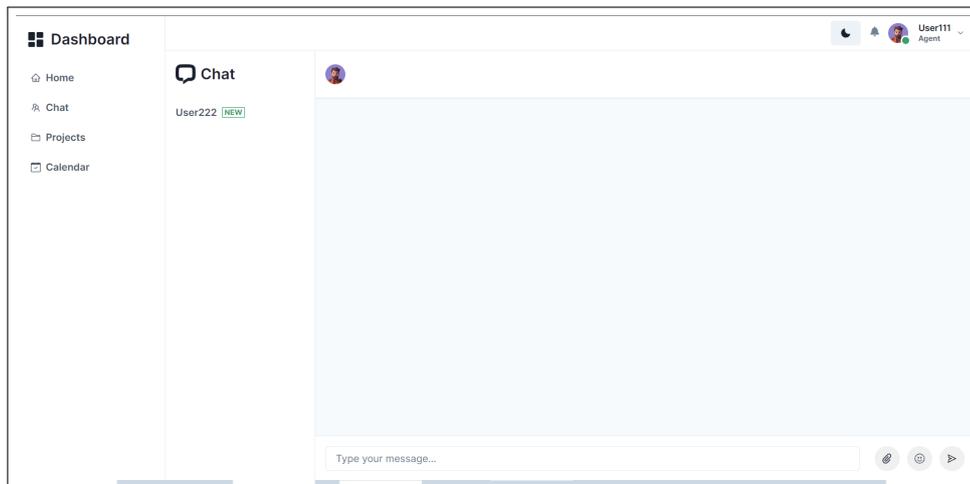
Gambar 3.6. Tampilan halaman *User* ketika *User* memasukan *Name* dan *Email*

Sesuai alur yang dijelaskan pada Gambar 3.1, pengguna akan memulai dengan memasukan *username* dan *email* sebagai langkah pertama untuk mengirimkan *request chat* ke *agent*. Saat sudah mengirimkan *request chat* ke *agent* pengguna sudah terhubung dengan server tinggal menunggu *assign* dari *agent* yang *online*. Pengembangan mengikuti alur yang diterapkan pada pengembangan *wireframe*, dengan membuat aturan seperti ini, *request chat* yang masuk akan disortir berdasarkan waktu *user* mengirimkan *request chat*.



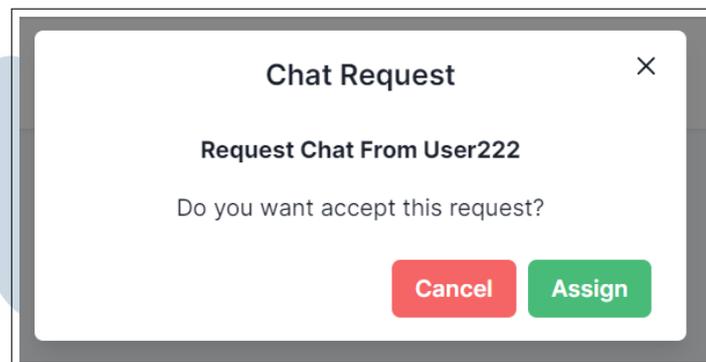
Gambar 3.7. Tampilan *Live Chat* halaman *User* ketika *User* masuk kedalam antrian

Meskipun begitu, pada tahap awal pengembangan dilakukan secara langsung dengan mendaftarkan pengguna untuk memastikan bahwa koneksi antar-pengguna dan server dapat berjalan sebelum pengembangan mencapai tingkat kompleksitas yang lebih tinggi, demi memudahkan proses pengujian. Oleh karena itu, pengguna akan menginisiasi koneksi dengan server segera setelah masuk, sehingga permintaan obrolan dari pengguna dapat segera terlihat pada *dashboard agent*.



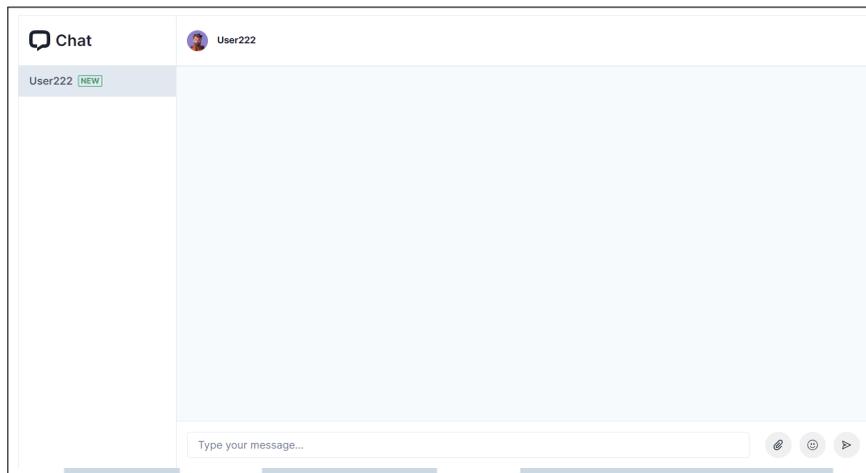
Gambar 3.8. Tampilan *Dashboard Agent* ketika *request chat User* masuk

Setelah munculnya permintaan chat, *agent* bisa menerima permintaan chat tersebut, sebagaimana diperlihatkan dalam Gambar 3.9. Permintaan chat yang diterima dari pengguna dapat segera terlihat pada *dashboard agent*, memungkinkan *agent* untuk merespons dengan cepat dan efisien. Pendekatan ini tidak hanya memastikan ketersediaan akses yang lancar antara pengguna dan server, tetapi juga menyederhanakan proses manajemen permintaan chat untuk meningkatkan responsivitas *agent* dalam menyediakan layanan kepada pengguna.

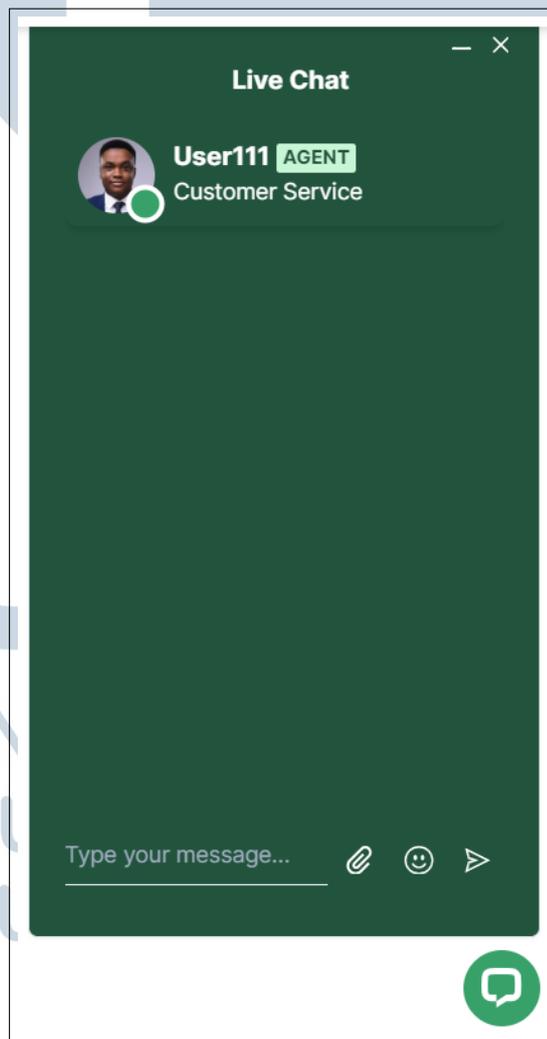


Gambar 3.9. Tampilan *Dashboard Agent* ketika *request chat User* ingin diassign *agent*

Setelah agen berhasil menerima permintaan *chat* dari pengguna, langkah selanjutnya adalah pembuatan room chat yang memfasilitasi interaksi dua arah atau terbuka antara *agent* dan pengguna yang dapat dilihat seperti Gambar 3.10 dan Gambar 3.11.

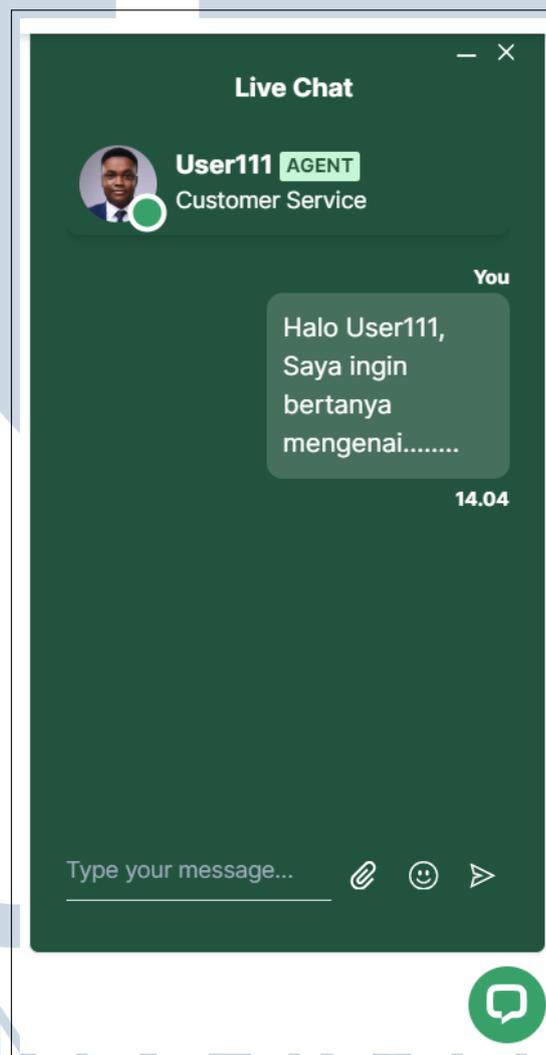


Gambar 3.10. Tampilan *Dashboard Agent* ketika *request chat User* sudah berhasil diassign agent

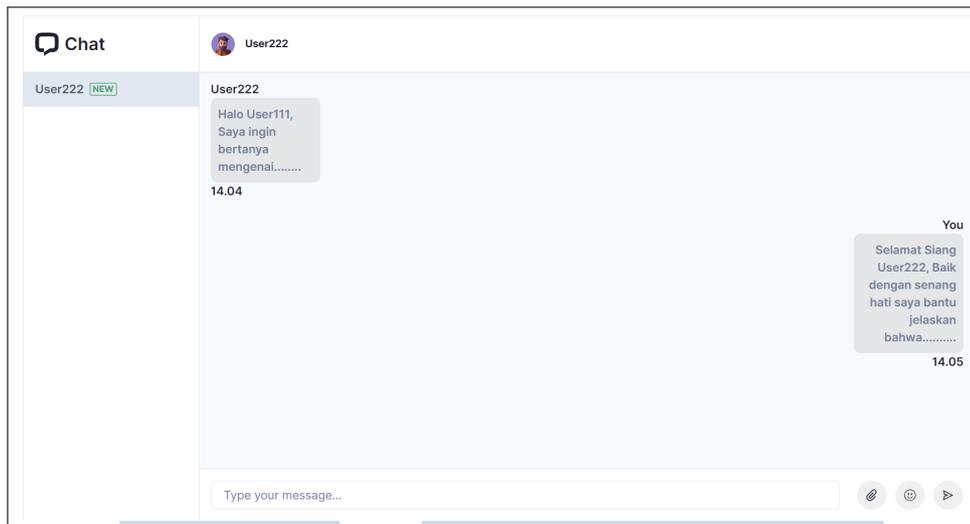


Gambar 3.11. Tampilan *Live Chat* halaman *User* ketika *User* masuk kedalam *Room Chat*

Proses ini memungkinkan terbentuknya lingkungan komunikasi yang dinamis, dimana pertukaran informasi dapat berlangsung secara efektif. Dengan terciptanya *room chat*, *agent* dan pengguna memiliki wadah yang terdedikasi untuk berkomunikasi secara *real-time*, memungkinkan pertanyaan diajukan, jawaban diberikan, serta diskusi atau bantuan dilakukan dengan lebih efisien yang dapat dilihat seperti Gambar 3.12 dan Gambar 3.13.



Gambar 3.12. Tampilan *Live Chat* halaman *User* ketika *User* mengirimkan pesan ke *Agent*



Gambar 3.13. Tampilan *Dashboard Agent* pada *room chat user* yang telah diassign ketika *Agent* menerima pesan dan mengirimkan pesan ke *User*

*Room chat* ini mencatat seluruh dialog antara *agent* dan pengguna, menciptakan jejak komunikasi yang berguna untuk pemantauan dan analisis. Keberadaan *room chat* ini tidak hanya memberikan akses cepat ke histori percakapan, tetapi juga memberikan landasan untuk pemahaman yang lebih baik tentang kebutuhan dan preferensi pengguna. Dengan adanya interaksi dua arah yang terbuka, pengguna merasakan kenyamanan dan kemudahan dalam mendapatkan informasi atau bantuan yang mereka perlukan, sementara *agent* dapat memberikan layanan yang lebih personal dan responsif.

### 3.4 Kendala dan Solusi yang Ditemukan

#### 3.4.1 Kendala

Selama berlangsungnya proses magang, beberapa kendala muncul saat mengintegrasikan sistem *Live Chat*. Beberapa kendala yang dihadapi antara lain:

1. Kurangnya dokumentasi terkait dengan *framework* yang diperlukan untuk proyek, karena *framework* yang digunakan masih tergolong baru.
2. Terdapat kendala ketika mengolah data yang diinput dikarenakan menggunakan dua *database* sehingga membutuhkan waktu yang cukup lama untuk *debugging* ketika mengalami *error*.
3. Terjadi kesalahpahaman antara rekan pengampu yang menyebabkan perbedaan dalam hasil yang diinginkan.

### 3.4.2 Solusi

Untuk mengatasi kendala yang telah disebutkan, sejumlah solusi diterapkan selama berlangsungnya proses magang, antara lain:

1. Dilakukan koordinasi intensif dengan rekan pengampu dan pengeksplorasian lebih lanjut.
2. Melakukan *debug* dan pengujian untuk mengevaluasi logika prosedural dari metode yang melaksanakan perubahan.
3. Menjalin komunikasi yang lebih erat dengan rekan pengampu apabila dalam pengeksplorasian mandiri muncul kendala atau kebingungan.

