

**PENGEMBANGAN APLIKASI FITUR LIVE CHAT DAN
SISTEM TICKETING UNTUK PENGIRIMAN BARANG
MENGUNAKAN FRAMEWORK ANGULAR
(STUDI KASUS : PT. URSABYTE)**

TUGAS AKHIR

Disusun sebagai salah satu syarat untuk kelulusan
Program Strata 1, Program Studi Teknik Informatika,
Universitas Pasundan Bandung

oleh :

Adawiyah Ajriah Lubis
Nrp. 21.304.0054



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PASUNDAN BANDUNG
JANUARI 2026**

LEMBAR PENGESAHAN
LAPORAN TUGAS AKHIR

Telah disetujui dan disahkan Laporan Tugas Akhir, dari :

Nama : Adawiyah Ajriah Lubis
NPM : 21.304.0054

Dengan judul :

“Pengembangan Aplikasi Fitur Live Chat dan Sistem Ticketing Untuk Pengiriman Barang Menggunakan Framework Angular (Studi Kasus : PT. UrsaByte)”



Bandung, 21 Januari 2026

Menyetujui,

Pembimbing Utama

Pembimbing Pendamping

(Ade Sukendar, ST., MT)

(Moch Ilham Apugrah, ST., M.SC.ENG)

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Saya menyatakan dengan sesungguhnya bahwa :

1. Tugas akhir ini Adalah benar-benar asli dan belum pernah diajukan untuk mendapatkan gelar akademik, baik di Universitas Pasundan Bandung maupun di Perguruan Tinggi lainnya.
2. Tugas akhir ini merupakan gagasan, rumusan dan penelitian saya sendiri, tanpa bantuan pihak lain kecuali arahan dari tim Dosen Pembimbing
3. Dalam tugas akhir ini tidak terdapat karya atau pendapat orang lain, kecuali bagian-bagian tertentu dalam penulisan laporan Tugas Akhir yang saya kutip dari hasil karya orang lain telah dituliskan dalam sumbernya secara jelas sesuai dengan norma, kaidah dan etika penulisan karya ilmiah, serta disebutkan dalam Daftar Pustaka pada tugas akhir ini
4. Kakas, perangkat lunak dan alat bantu kerja lainnya yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab saya, bukan tanggung jawab Universitas Pasundan Bandung Apabila di kemudian hari ditemukan seluruh atau Sebagian laporan tugas akhir ini bukan hasil karya sendiri atau adanya plagiasi dalam bagian-bagian tertentu, saya bersedia menerima sanksi akademik, termasuk pencabutan gelar akademik yang saya sandang sesuai dengan norma yang berlaku di Universitas Pasundan Bandung, serta perundang-undangan lainnya.

Bandung , 21 Januari 2026

Yang membuat pernyataan

Adawiyah Ajriah Lubis

NPM. 21.304.0054

ABSTRAK

Penelitian ini bertujuan untuk mengembangkan aplikasi layanan live chat dan sistem ticketing berbasis web pada proses pengiriman barang dengan menggunakan framework *Angular* di PT. UrsaByte. Pengembangan difokuskan pada perancangan fitur komunikasi *Real-time* antara *Customer* dan *Customer service*, serta mekanisme pencatatan otomatis keluhan ke dalam bentuk tiket. Sistem ini dirancang untuk mempermudah *Customer* dalam menyampaikan kendala maupun pertanyaan, sekaligus mendukung *Customer service* dalam melakukan penanganan laporan secara lebih cepat, efisien, dan terstruktur.

PT. UrsaByte merupakan perusahaan yang bergerak di bidang teknologi informasi dan telah mengembangkan berbagai sistem pengiriman barang. Namun, sistem yang telah digunakan sebelumnya belum menyediakan fitur live chat dan ticketing yang memungkinkan proses komunikasi berlangsung secara langsung dan terdokumentasi. Kondisi tersebut mengakibatkan *Customer* mengalami kesulitan dalam memperoleh informasi secara cepat, serta menambah beban kerja *Customer service* dalam menangani pertanyaan berulang. Melalui aplikasi yang dikembangkan, integrasi antara fitur live chat, *Firebase Realtime Database*, dan sistem ticketing diharapkan dapat mengatasi permasalahan tersebut dengan menyediakan layanan yang lebih responsif, akurat, dan mudah digunakan.

Untuk pengembangan lebih lanjut, sistem ini masih dapat ditingkatkan dengan penambahan fungsionalitas baru, seperti peningkatan kapabilitas fitur live chat, optimalisasi proses ticketing, serta perluasan modul layanan sesuai kebutuhan operasional perusahaan. Selain itu, evaluasi dan perbaikan secara berkelanjutan berdasarkan hasil pengujian maupun umpan balik pengguna perlu dilakukan agar sistem dapat berkembang menjadi lebih stabil, efektif, dan adaptif terhadap kebutuhan di masa mendatang.

Kata Kunci: Live Chat, Sistem Ticketing, Pengiriman Barang, *Angular*, *Firebase*, *Node.js*

ABSTRACT

This research aims to develop a web-based live chat service application and ticketing system for the goods delivery process using the *Angular* framework at PT. UrsaByte. The development focuses on designing *Real-time* communication features between *Customers* and *Customer service*, as well as implementing an automatic complaint-logging mechanism in the form of tickets. The system is designed to facilitate *Customers* in submitting issues or inquiries, while also supporting *Customer service* in handling reports more quickly, efficiently, and in a more structured manner.

PT. UrsaByte is an information technology company that has developed various goods delivery systems. However, the previously used system did not provide live chat and ticketing features that enable direct and well-documented communication. This limitation caused *Customers* to experience difficulties in obtaining information promptly and increased the workload of *Customer service* in handling repetitive inquiries. Through the developed application, the integration of live chat features, *Firebase* Realtime Database, and a ticketing system is expected to address these issues by providing services that are more responsive, accurate, and user-friendly.

For further development, the system can still be enhanced by adding new functionalities, such as improving the capabilities of the live chat feature, optimizing the ticketing process, and expanding service modules according to the company's operational needs. In addition, continuous evaluation and improvement based on testing results and user feedback are necessary to ensure that the system evolves to become more stable, effective, and adaptive to future demands.

Keywords: Live Chat, Ticketing System, Goods Delivery, *Angular*, *Firebase*, *Node.js*

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT atas rahmat dan hidayah-Nya sehingga penulis berhasil menyusun Laporan Tugas Akhir dengan judul “Pengembangan Aplikasi Fitur Live Chat dan Sistem Ticketing untuk Pengiriman Barang Menggunakan Framework *Angular* (Studi Kasus: PT. UrsaByte)”.

Penulis menyadari bahwa laporan ini tidak akan terwujud tanpa bantuan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar-besarnya atas segala dukungan, baik moril maupun materil, yang telah diberikan sehingga laporan ini dapat terselesaikan :

1. Bapak Anggoro Ari Nurcahyo, ST, M.Kom sebagai Ketua Program Studi Teknik Informatika Universitas Pasundan.
2. Bapak Ade Sukendar, S.T., M.T., selaku dosen pembimbing, penulis mengucapkan terima kasih atas bimbingan, arahan, serta ilmu yang telah diberikan selama proses penyusunan skripsi ini.
3. Bapak Moch. Ilham Anugrah, S.T., M.Sc.Eng., selaku pembimbing pendamping, penulis mengucapkan terima kasih atas masukan, saran, dan dukungan yang sangat membantu dalam penyelesaian skripsi ini.
4. Seluruh dosen Program Studi Teknik Informatika Universitas Pasundan, penulis mengucapkan terima kasih atas ilmu dan pengalaman berharga yang telah diberikan selama penulis menempuh pendidikan di program studi ini.
5. Kedua orang tua tercinta, ayah Indra Fahrudin Lubis, S.Kep dan Ibu Vita Novianti Nasution, penulis mengucapkan terima kasih atas seluruh pengorbanan, ketulusan, serta dukungan moral dan finansial yang senantiasa diberikan. Didikan dan perjuangan ayah dan ibu menjadi motivasi bagi penulis untuk menjadi pribadi yang kuat, mandiri, dan bertanggung jawab. Semoga skripsi ini dapat menjadi salah satu bentuk kebanggaan bagi ayah dan ibu
6. Ketiga adikku, Nayla Indriyanti Lubis, Salsafira Nur Indriyanti Lubis, dan Amir Yusuf Martua Lubis. Terima kasih telah menjadi bagian dari perjalanan penulis dalam menempuh pendidikan selama ini. Terima kasih atas semangat, doa, dan kasih sayang yang selalu diberikan. Tumbuhlah menjadi pribadi yang terbaik dan membanggakan, adik-adikku.
7. Yusuf Andrianto, penulis mengucapkan terima kasih yang tulus atas kesediaannya selalu menemani, mendengarkan keluh kesah, serta memberikan dukungan, doa, semangat, dan bantuan kepada penulis.
8. Sahabat penulis, Gloria Rustama Simbolon, S.T., penulis mengucapkan terima kasih atas dukungan, motivasi, dan kebersamaan selama masa perkuliahan.
9. Kepada Putri Azizah, Raisa Isna Ainun, dan Rika Febriyanti, penulis mengucapkan terima kasih karena telah menjadi teman seperbimbingan yang selalu mendukung dan memberi semangat selama penulisan skripsi.

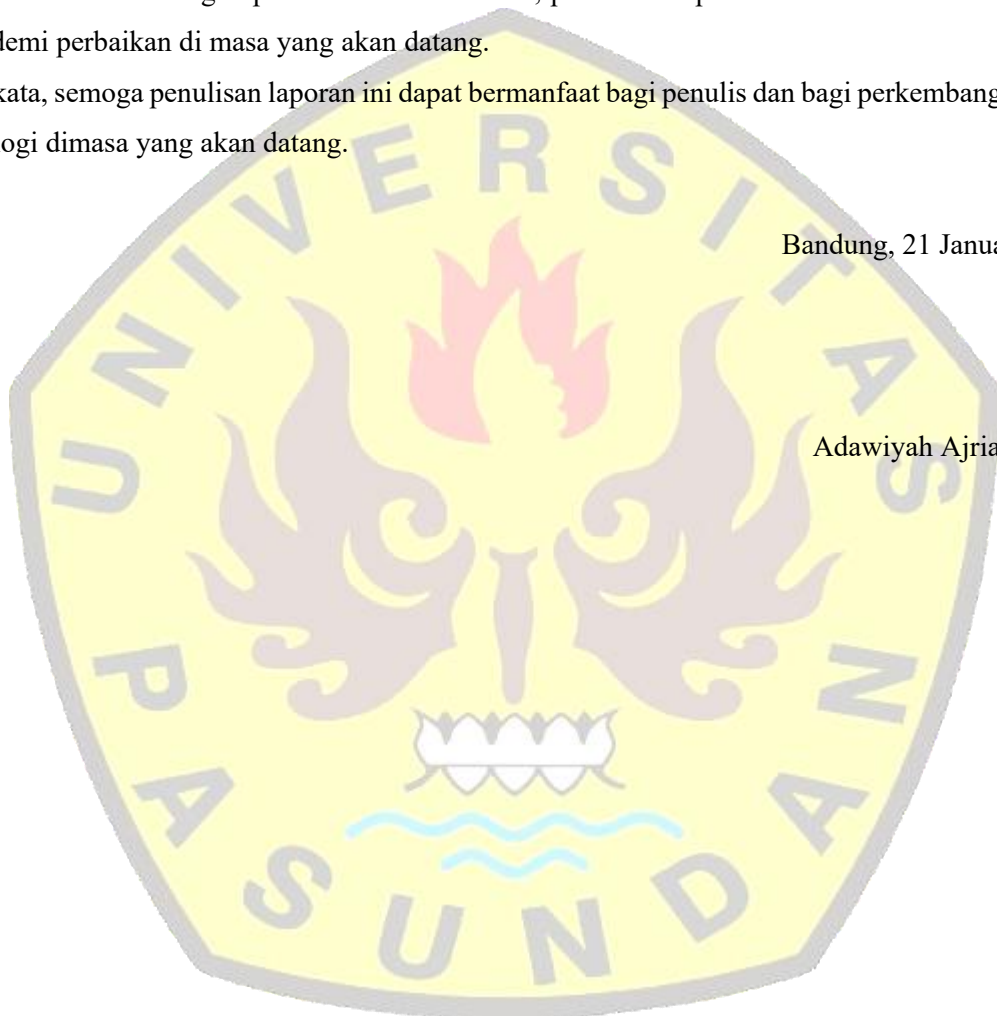
10. *Last but not least*, terima kasih untuk Adawiyah Ajriah ya!, diri saya sendiri. Rasa syukur yang mendalam atas keberanian untuk terus melangkah dan menyelesaikan apa yang telah dimulai. Terima kasih telah bertahan di hari-hari tersulit dan tetap percaya bahwa setiap usaha memiliki arti. Teruslah menjadi manusia yang mau belajar, berproses, dan tidak lelah mencoba. Seluruh proses yang telah dijalani menjadi bagian berharga dalam perjalanan hidup penulis.

Tiada gading yang tak retak, tiada gelombang tanpa ombak, segala kesalahan merupakan kelemahan dan kekurangan penulis. oleh karena itu, penulis harapkan kritik dan saran dari semua pihak demi perbaikan di masa yang akan datang.

Akhir kata, semoga penulisan laporan ini dapat bermanfaat bagi penulis dan bagi perkembangan ilmu Teknologi dimasa yang akan datang.

Bandung, 21 Januari 2026

Adawiyah Ajriah Lubis

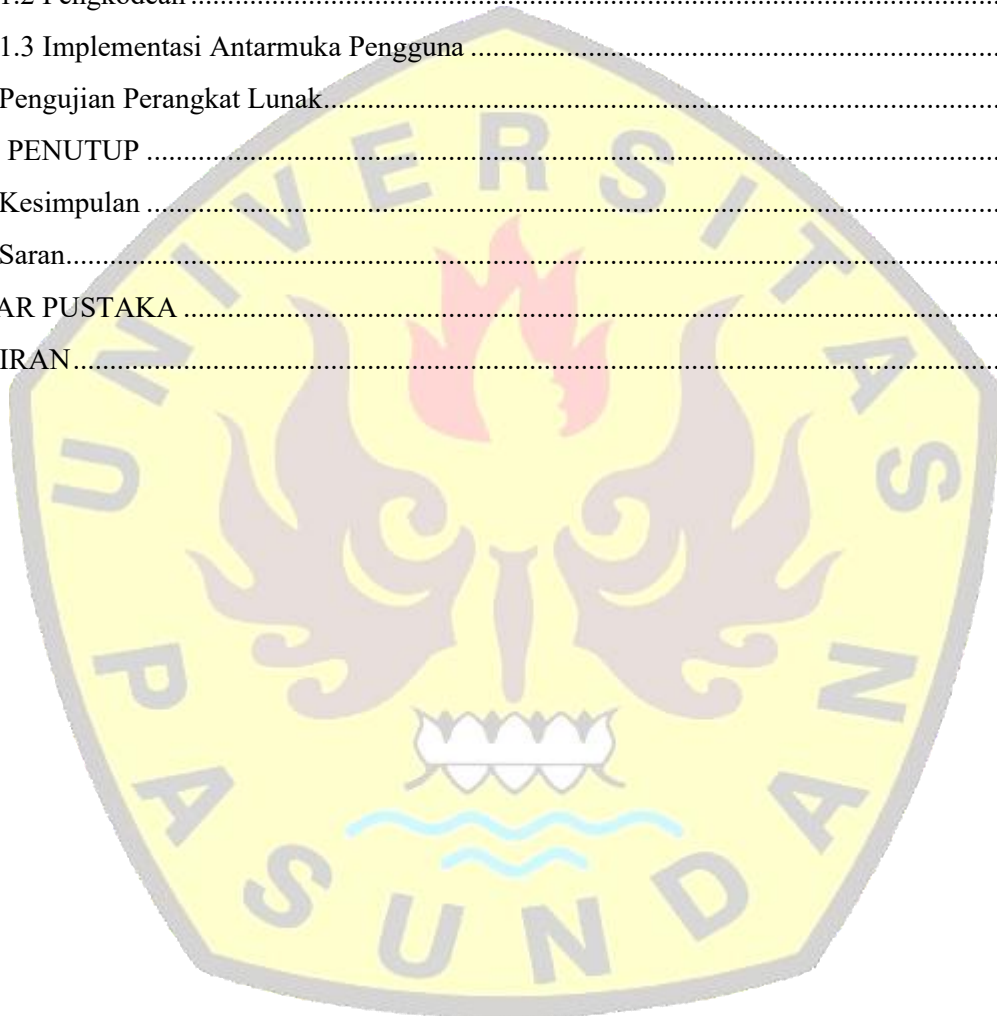


DAFTAR ISI

LEMBAR PENGESAHAN	i
LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR	ii
ABSTRAK.....	iii
ABSTRACT.....	iv
KATA PENGANTAR	v
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xiv
DAFTAR ISTILAH	xvii
DAFTAR SIMBOL.....	xviii
BAB 1 PENDAHULUAN	1-1
1.1 Latarbelakang.....	1-1
1.2 Identifikasi Masalah.....	1-2
1.3 Tujuan Tugas Akhir	1-2
1.4 Lingkup Tugas Akhir	1-3
1.5 Metodologi Penyelesaian Tugas Akhir	1-4
1.6. Sistematika Penulisan Tugas Akhir	1-7
BAB 2 LANDASAN TEORI DAN PENELITIAN TERDAHULU.....	2-1
2.1. Teori Pendukung	2-1
2.1.1 Pengembangan	2-1
2.1.2 Aplikasi	2-1
2.1.3 Live Chat.....	2-1
2.1.4 Pengiriman Barang.....	2-2
2.1.5 <i>Firebase</i>	2-3
2.1.6 NoSQL	2-3
2.1.7 Sistem Ticketing	2-3
2.1.8 <i>Real-time</i>	2-4
2.1.9 Framework	2-5
2.1.9.1 <i>Angular</i>	2-5
2.1.10 Website.....	2-6
2.1.11 Hypertext Markup Language	2-6
2.1.12 Node.JS	2-7
2.1.13 Modular Monolith Architecture	2-7

2.1.14 Role Based Access Control (RBAC)	2-8
2.1.15 Metode Waterfall	2-8
2.1.16 Metode Pengujian <i>Black-box</i>	2-9
2.2. Penelitian Terdahulu	2-10
BAB 3 SKEMA PENELITIAN	3-1
3.1. Alur Penyelesaian Tugas Akhir	3-1
3.2. Rumusan masalah.....	3-3
3.2.1 Analisis sebab akibat.....	3-3
3.2.2 Solusi Masalah	3-4
3.3. Kerangka pemikiran Teoritis.....	3-4
3.3.1 Gambaran Produk TA	3-5
3.3.2 Skema Analisis Teori	3-6
3.4 Profil Penelitian.....	3-7
3.4.1 Sejarah Perusahaan.....	3-7
3.4.2 Visi dan Misi PT. Ursabyte.....	3-7
3.4.2 Struktur Organisasi PT. Ursabyte	3-8
BAB 4 ANALISIS DAN PERANCANGAN	4-1
4.1. Sistem Saat Ini (Current System).....	4-1
4.2. Analisis Kebutuhan	4-2
4.2.1 Menentukan Kebutuhan Perangkat Lunak	4-2
4.2.2 Analisis Hasil dan Pengumpulan Data	4-2
4.2.3 Analisis Stakeholder.....	4-2
4.2.4 Analisis Fungsional.....	4-3
4.2.5 Analisis Pengguna	4-5
4.2.6 Pemodelan berbasis skenario	4-5
4.2.7 Deskripsi Use Case	4-7
4.2.8 Skenario <i>Use Case</i>	4-7
4.2.9 Aliran Aktivitas.....	4-12
4.3. Pemodelan Data	4-14
4.3.1 Objek Data	4-14
4.3.2 Atribut Data.....	4-14
4.3.3 Pemodelan Berbasis Perilaku	4-16
4.4. Pemodelan Berbasis Kelas	4-23
4.4.1 Identifikasi Kelas-Kelas Analisis	4-23
4.4.2 Menentukan Atribut	4-24
4.4.3 Menentukan Perilaku	4-26

4.5. Perancangan Perangkat Lunak	4-28
4.5.1 Perancangan Data.....	4-28
4.5.2 Perancangan Kelas	4-29
4.5.3 Perancangan Arsitektur	4-30
4.6. Perancangan Antarmuka Pengguna.....	4-32
BAB 5 IMPLEMENTASI DAN PENGUJIAN	5-1
5.1. Kontruksi Perangkat Lunak.....	5-1
5.1.1 Kebutuhan Pembangunan Perangkat Lunak	5-1
5.1.2 Pengkodean	5-2
5.1.3 Implementasi Antarmuka Pengguna	5-5
5.2. Pengujian Perangkat Lunak.....	5-25
BAB 6 PENUTUP	6-1
6.1. Kesimpulan	6-1
6.2. Saran.....	6-1
DAFTAR PUSTAKA	
LAMPIRAN.....	



DAFTAR TABEL

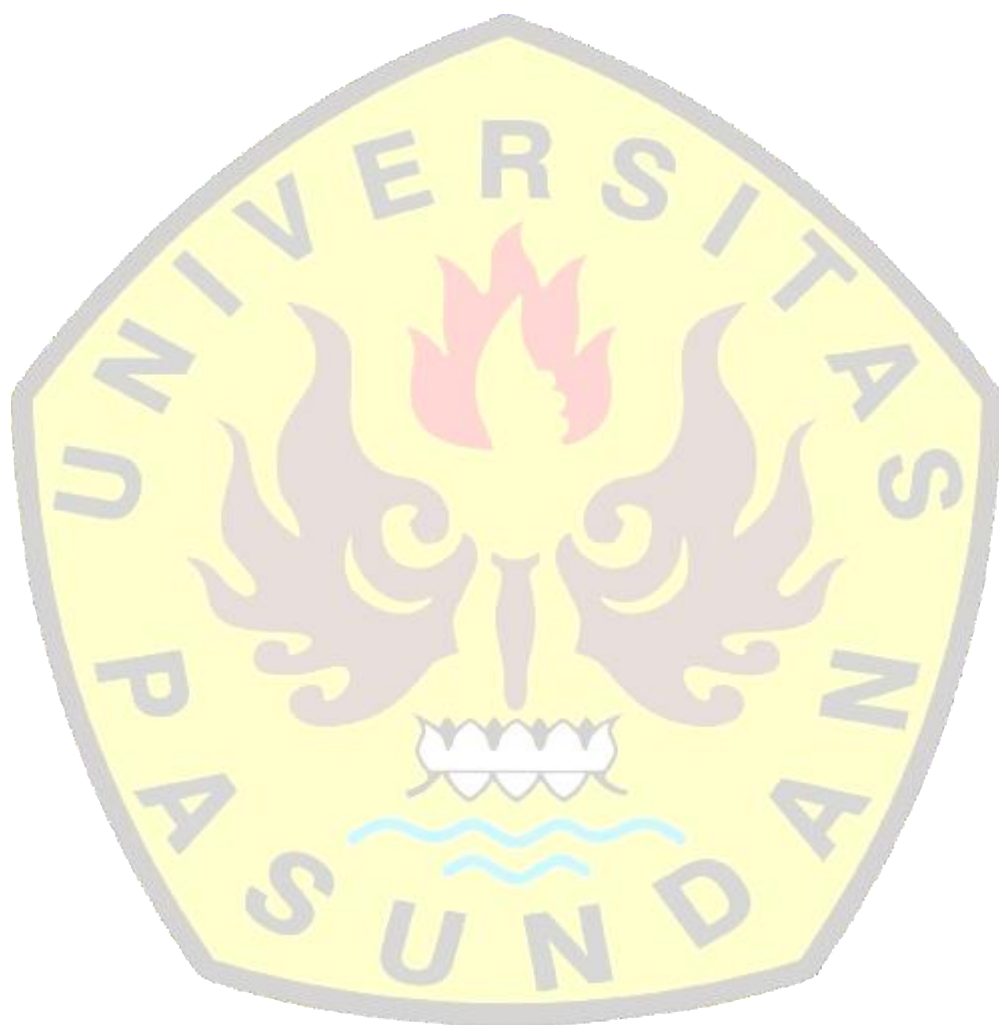
Tabel 2.1 Penelitian Terdahulu	2-10
Tabel 3.1 Alur Penyelesaian Tugas Akhir	3-1
Tabel 3.2 Tabel solusi Masalah.....	3-4
Tabel 3.3 Skema Analisis Teori	3-6
Tabel 3.4 Penjelasan Skema Teori	3-7
Tabel 4.1 Current Activity Diagram Pelaporan Masalah	4-1
Tabel 4.2 Analisis Stakeholder	4-2
Tabel 4.3 Analisis Fungsional.....	4-3
Tabel 4.4 Fitur Kebutuhan Fungsional.....	4-4
Tabel 4.5 Analisis Pengguna.....	4-5
Tabel 4.6 Pemodelan Berbasis Skenario	4-6
Tabel 4.7 Deskripsi Use Case	4-7
Table 4.8 Skenario Use Case Manajemen Akun Pengguna	4-8
Table 4.9 Skenario Use Case Mengelola Tiket.....	4-9
Table 4.10 Skenario Use Case Memulai Sesi Live Chat.....	4-9
Table 4.11 Skenario Use Case Menampilkan Daftar Sesi Chat	4-10
Table 4.12 Skenario Use Case Melihat Status Online atau Offline Pengguna.....	4-10
Table 4.13 Skenario Use Case Mengirim Pesan	4-10
Table 4.14 Skenario Use Case Menerima Pesan.....	4-11
Table 4.15 Skenario Use Case Menampilkan Identitas Pengirim Pesan.....	4-11
Table 4.16 Skenario Use Case Melihat Pesan yang Tersimpan	4-12
Tabel 4.17 Diagram Activity Sistem LiveChat dan Ticketing.....	4-13
Tabel 4.18 Objek Data	4-14
Tabel 4.19 Atribut Data	4-15
Tabel 4.20 Diagram Sequence Mengelola Akun Pengguna.....	4-17
Tabel 4.21 Diagram Sequence Memulai Sesi Live Chat	4-18
Tabel 4.22 Diagram Sequence Mengelola Tiket.....	4-19
Tabel 4.23 Diagram Sequence Menampilkan Daftar Sesi Chat.....	4-20
Tabel 4.24 Diagram Sequence Melihat Status Online atau Offline	4-20
Tabel 4.25 Diagram Sequence Mengirim Pesan	4-21
Tabel 4.26 Diagram Sequence Menerima Pesan.....	4-21
Table 4.27 Diagram Sequence Menampilkan Identitas Pengirim Pesan.....	4-22
Tabel 4.28 Diagram Sequence Melihat Pesan yang Tersimpan	4-22
Tabel 4.29 Identifikasi Kelas-kelas Analisis.....	4-23
Tabel 4.30 Menentukan Atribut	4-24

Tabel 4.31 Menentukan Perilaku	4-26
Tabel 4.32 Perancangan Data.....	4-28
Tabel 4.33 Perancangan Kelas	4-29
Tabel 4.34 Perancangan Antarmuka Halaman Beranda.....	4-32
Tabel 4.35 Perancangan Antar Muka Halaman Register	4-32
Tabel 4.36 Perancangan Antar Muka Halaman konfirmasi Pendaftaran	4-33
Tabel 4.37 Perancangan Antar Muka Fitur Live Chat Sebelum <i>Customer</i> Melakukan Login	4-33
Tabel 4.38 Perancangan Antar Muka Halaman Jendela Chat	4-34
Tabel 4.39 Perancangan Antar Muka Halaman Quick Question Chat.....	4-35
Tabel 4.40 Perancangan Antar Muka Halaman Pelaporan Kendala	4-35
Tabel 4.41 Perancangan Antar Muka Halaman Chat Laporan Tiket	4-36
Tabel 4.42 Perancangan Antar Muka Halaman Layanan.....	4-36
Tabel 4.43 Perancangan Antar Muka Halaman Detail Layanan	4-37
Tabel 4.44 Perancangan Antar Muka Halaman Frequently Asked Questions	4-37
Tabel 4.45 Perancangan Antar Muka Halaman Packaging Information.....	4-38
Tabel 4.46 Perancangan Antar Muka Halaman Company Profile	4-39
Tabel 4.47 Perancangan Antarmuka Halaman Pesan Penutup Percakapan	4-39
Tabel 4.48 Perancangan Antar Muka Halaman List Ticket	4-40
Tabel 4.49 Perancangan Antar Muka Halaman Notifications.....	4-40
Tabel 4.50 Perancangan Antar Muka Halaman Edit Ticket.....	4-41
Tabel 4.51 Perancangan Antar Muka Halaman Konfirmasi Update Tiket	4-42
Tabel 4.52 Perancangan Antar Muka Halaman Pemberitahuan Ketika Tidak Ada Percakapan yang Dipilih	4-42
Tabel 4.53 Perancangan Antar Muka Halaman Chat History	4-43
Tabel 4.54 Perancangan Antar Muka Halaman Ticket Recap	4-43
Tabel 4.55 Perancangan Antar Muka Halaman Detail Ticket Recap.....	4-44
Tabel 4.56 Perancangan Antar Muka Halaman Users	4-44
Tabel 4.57 Perancangan Antar Muka Halaman Detail Users.....	4-45
Tabel 4.58 Perancangan Antar Muka Halaman Edit Users.....	4-45
Tabel 4.59 Perancangan Antar Muka Halaman Konfirmasi Penghapusan User.....	4-46
Tabel 4.60 Perancangan Antar Muka Halaman Daftar Tiket yang Ditugaskan ke CS	4-46
Tabel 4.61 Perancangan Antar Muka Halaman Notifications.....	4-47
Tabel 4.62 Perancangan Antar Muka Halaman Edit Ticket.....	4-47
Tabel 4.63 Perancangan Antar Muka Halaman Konfirmasi Update Ticket.....	4-48
Tabel 4.64 Perancangan Antar Muka Halaman Detail Tickets	4-48
Tabel 4.65 Perancangan Antar Muka Pemberitahuan Ketika Tidak Ada Percakapan yang Dipilih	4-49

Tabel 4.66 Perancangan Antar Muka Halaman Percakapan	4-49
Tabel 4.67 Perancangan Antar Muka Halaman Konfirmasi Akhiri Percakapan.....	4-50
Tabel 4.68 Perancangan Antar Muka Halaman Pesan Penutup Percakapan.....	4-50
Tabel 4.69 Perancangan Antar Muka Halaman List Ticket Recap	4-51
Tabel 4.70 Perancangan Antar Muka Halaman Detail Ticket Recap.....	4-51
Tabel 5.1 Spesifikasi perangkat lunak untuk pembangunan	5-1
Tabel 5.2 Perangkat Lunak yang digunakan dalam pembangunan	5-1
Tabel 5.3 Model Aplikasi Web	5-2
Tabel 5.4 Controller Aplikasi Web	5-3
Tabel 5.5 View Aplikasi Web.....	5-4
Tabel 5.6 Implementasi Antar Muka Halaman Beranda.....	5-5
Tabel 5.7 Implementasi Antar Muka Halaman Register.....	5-6
Tabel 5.8 Implementasi Antar Muka Halaman konfirmasi Pendaftaran.....	5-6
Tabel 5.9 Implementasi Antar Muka Fitur Live Chat Sebelum <i>Customer</i> Melakukan Login	5-7
Tabel 5.10 Implementasi Antar Muka Halaman Jendela Chat.....	5-7
Tabel 5.11 Implementasi Antar Muka Halaman Quick Question Chat	5-8
Tabel 5.12 Implementasi Antar Muka Halaman Pelaporan Kendala.....	5-8
Tabel 5.13 Implementasi Antar Muka Halaman Chat Laporan Tiket.....	5-9
Tabel 5.14 Implementasi Antar Muka Halaman Layanan	5-9
Tabel 5.15 Implementasi Antar Muka Halaman Detail Layanan.....	5-10
Tabel 5.16 Implementasi Antar Muka Halaman Frequently Asked Questions.....	5-10
Tabel 5.17 Implementasi Antar Muka Halaman Packaging Information	5-11
Tabel 5.18 Implementasi Antar Muka Halaman Company Profile.....	5-11
Tabel 5.19 Perancangan Antarmuka Halaman Notifikasi Penutup Percakapan	5-12
Tabel 5.20 Implementasi Antar Muka Halaman List Ticket.....	5-12
Tabel 5.21 Implementasi Antar Muka Halaman Notifications	5-13
Tabel 5.22 Implementasi Antar Muka Halaman Edit Ticket	5-13
Tabel 5.23 Implementasi Antar Muka Halaman Konfirmasi Update Tiket.....	5-14
Tabel 5.24 Implementasi Antar Muka Halaman Pemberitahuan Ketika Tidak Ada Percakapan yang Dipilih	5-14
Tabel 5.25 Implementasi Antar Muka Halaman Chat History.....	5-15
Tabel 5.26 Implementasi Antar Muka Halaman Ticket Recap.....	5-16
Tabel 5.27 Implementasi Antar Muka Halaman Detail Ticket Recap	5-16
Tabel 5.28 Implementasi Antar Muka Halaman Users	5-17
Tabel 5.29 Implementasi Antar Muka Halaman Detail Users	5-17
Tabel 5.30 Implementasi Antar Muka Halaman Edit Users	5-18
Tabel 5.31 Implementasi Antar Muka Halaman Konfirmasi Penghapusan User	5-18

Tabel 5.32 Implementasi Antar Muka Halaman Daftar Tiket yang Ditugaskan ke CS	5-19
Tabel 5.33 Implementasi Antar Muka Halaman Notifications	5-19
Tabel 5.34 Implementasi Antar Muka Halaman Edit Ticket	5-20
Tabel 5.35 Implementasi Antar Muka Halaman Konfirmasi Update Ticket	5-20
Tabel 5.36 Implementasi Antar Muka Halaman Detail Tickets.....	5-21
Tabel 5.37 Implementasi Antar Muka Halaman Pemberitahuan Ketika Tidak Ada Percakapan yang Dipilih	5-21
Tabel 5.38 Implementasi Antar Muka Halaman Percakapan.....	5-22
Tabel 5.39 Implementasi Antar Muka Halaman Konfirmasi Akhiri Percakapan	5-22
Tabel 5.40 Implementasi Antar Muka Halaman Pesan Penutup Percakapan	5-23
Tabel 5.41 Implementasi Antar Muka Halaman List Ticket Recap.....	5-23
Tabel 5.42 Implementasi Antar Muka Halaman Detail Ticket Recap	5-24
Tabel 5.4 Pengujian Perangkat Lunak	5-25





DAFTAR GAMBAR

Gambar 1.1 Metodologi Penyelesaian Tugas Akhir	1-4
Gambar 2.1 Metode Waterfall.....	2-8
Gambar 3.1 <i>Fishbone</i> Penyebab Masalah.....	3-3
Gambar 3.2 <i>Fishbone</i> Solusi.....	3-5
Gambar 3.4 Struktur Organisasi.....	3-8
Gambar 4.1 Current Activity Diagram Pelaporan Masalah	4-1
Gambar 4.2 Pemodelan Berbasis skenario.....	4-6
Gambar 4.3 Current Activity Diagram Sistem LiveChat dan Ticketing.....	4-13
Gambar 4.4 Diagram Sequence Mengelola Akun <i>Customer service</i>	4-17
Gambar 4.5 Diagram Sequence Memulai Sesi Live Chat.....	4-18
Gambar 4.6 Diagram Sequence Mengelola Tiket	4-19
Gambar 4.7 Diagram Sequence Menampilkan Daftar Sesi Chat	4-20
Gambar 4.8 Diagram Sequence Melihat Status Online atau Offline	4-20
Gambar 4.9 Diagram Sequence Mengirim Pesan	4-21
Gambar 4.10 Diagram Sequence Menerima Pesan.....	4-21
Gambar 4.11 Diagram Sequence Menampilkan Identitas Pengirim Pesan.....	4-22
Gambar 4.12 Diagram Sequence Melihat Pesan yang Tersimpan	4-22
Gambar 4.13 Perancangan Kelas	4-29
Gambar 4.14 Rancangan Arsitektur Sistem.....	4-31
Gambar 4.15 Perancangan Antar Muka Halaman Beranda	4-32
Gambar 4.16 Perancangan Antar Muka Halaman Register	4-32
Gambar 4.17 Perancangan Antar Muka Halaman konfirmasi Pendaftaran	4-33
Gambar 4.18 Perancangan Antar Muka Fitur Live Chat Sebelum <i>Customer</i> Melakukan Login..	4-33
Gambar 4.19 Perancangan Antar Muka Halaman Jendela Chat	4-34
Gambar 4.20 Perancangan Antar Muka Halaman Quick Question Chat.....	4-35
Gambar 4.21 Perancangan Antar Muka Halaman Pelaporan Kendala	4-35
Gambar 4.22 Perancangan Antar Muka Halaman Chat Laporan Tiket	4-36
Gambar 4.23 Perancangan Antar Muka Halaman Layanan.....	4-36
Gambar 4.24 Perancangan Antar Muka Halaman Detail Layanan	4-37
Gambar 4.25 Perancangan Antar Muka Halaman Frequently Asked Questions	4-37
Gambar 4.26 Perancangan Antar Muka Halaman Packaging Information	4-38
Gambar 4.27 Perancangan Antar Muka Halaman Company Profile	4-39
Gambar 4.28 Perancangan Antarmuka Halaman Pesan Penutup Percakapan	4-39
Gambar 4.29 Perancangan Antar Muka Halaman List Ticket	4-40
Gambar 4.30 Perancangan Antar Muka Halaman Notifications.....	4-40

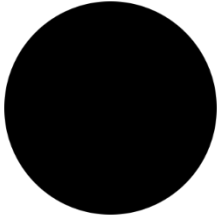

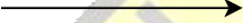

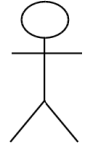
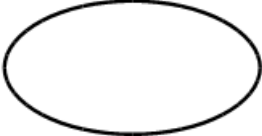

Gambar 4.31 Perancangan Antar Muka Halaman Edit Ticket.....	4-41
Gambar 4.32 Perancangan Antar Muka Halaman Konfirmasi Update Tiket.....	4-42
Gambar 4.33 Perancangan Antar Muka Halaman Pemberitahuan Ketika Tidak Ada Percakapan yang Dipilih.....	4-42
Gambar 4.34 Perancangan Antar Muka Halaman Chat History	4-43
Gambar 4.35 Perancangan Antar Muka Halaman Ticket Recap.....	4-43
Gambar 4.36 Perancangan Antar Muka Halaman Detail Ticket Recap.....	4-44
Gambar 4.37 Perancangan Antar Muka Halaman Users.....	4-44
Gambar 4.38 Perancangan Antar Muka Halaman Detail Users.....	4-45
Gambar 4.39 Perancangan Antar Muka Halaman Edit Users	4-45
Gambar 4.40 Perancangan Antar Muka Halaman Konfirmasi Penghapusan User	4-46
Gambar 4.41 Perancangan Antar Muka Halaman Daftar Tiket yang Ditugaskan ke CS.....	4-46
Gambar 4.42 Perancangan Antar Muka Halaman Notifications	4-47
Gambar 4.43 Perancangan Antar Muka Halaman Edit Ticket.....	4-47
Gambar 4.44 Perancangan Antar Muka Halaman Konfirmasi Update Ticket.....	4-48
Gambar 4.45 Perancangan Antar Muka Halaman Detail Tickets	4-48
Gambar 4.46 Perancangan Antar Muka Halaman Pemberitahuan Ketika Tidak Ada Percakapan yang Dipilih.....	4-49
Gambar 4.47 Perancangan Antar Muka Halaman Percakapan	4-49
Gambar 4.48 Perancangan Antar Muka Halaman Konfirmasi Akhiri Percakapan.....	4-50
Gambar 4.49 Perancangan Antar Muka Halaman Pesan Penutup Percakapan	4-50
Gambar 4.50 Perancangan Antar Muka Halaman List Ticket Recap	4-51
Gambar 4.51 Perancangan Antar Muka Halaman Detail Ticket Recap.....	4-51
Gambar 5.1 Implementasi Antar Muka Halaman Beranda	5-5
Gambar 5.2 Implementasi Antar Muka Halaman Register	5-6
Gambar 5.3 Implementasi Antar Muka Halaman konfirmasi Pendaftaran	5-6
Gambar 5.4 Implementasi Antar Muka Fitur Live Chat Sebelum <i>Customer</i> Melakukan Login ..	5-7
Gambar 5.5 Implementasi Antar Muka Halaman Jendela Chat.....	5-7
Gambar 5.6 Implementasi Antar Muka Halaman Quick Question Chat.....	5-8
Gambar 5.7 Implementasi Antar Muka Halaman Pelaporan Kendala	5-8
Gambar 5.8 Implementasi Antar Muka Halaman Chat Laporan Tiket	5-9
Gambar 5.9 Implementasi Antar Muka Halaman Layanan.....	5-9
Gambar 5.10 Implementasi Antar Muka Halaman Detail Layanan	5-10
Gambar 5.11 Implementasi Antar Muka Halaman Frequently Asked Questions	5-10
Gambar 5.12 Implementasi Antar Muka Halaman Packaging Information.....	5-11
Gambar 5.13 Implementasi Antar Muka Halaman Company Profile	5-12
Gambar 5.14 Perancangan Antarmuka Halaman Notifikasi Penutup Percakapan.....	5-12

Gambar 5.15 Implementasi Antar Muka Halaman List Ticket.....	5-12
Gambar 5.16 Implementasi Antar Muka Halaman Notifications	5-13
Gambar 5.17 Implementasi Antar Muka Halaman Edit Ticket	5-13
Gambar 5.18 Implementasi Antar Muka Halaman Konfirmasi Update Tiket	5-14
Gambar 5.19 Implementasi Antar Muka Halaman Pemberitahuan Ketika Tidak Ada Percakapan yang Dipilih.....	5-14
Gambar 5.20 Implementasi Antar Muka Halaman Chat History	5-15
Gambar 5.21 Implementasi Antar Muka Halaman Ticket Recap	5-16
Gambar 5.22 Implementasi Antar Muka Halaman Detail Ticket Recap	5-16
Gambar 5.23 Implementasi Antar Muka Halaman Users	5-17
Gambar 5.24 Implementasi Antar Muka Halaman Detail Users	5-17
Gambar 5.25 Implementasi Antar Muka Halaman Edit Users.....	5-18
Gambar 5.26 Implementasi Antar Muka Halaman Konfirmasi Penghapusan User.....	5-18
Gambar 5.27 Implementasi Antar Muka Halaman Daftar Tiket yang Ditugaskan ke CS	5-19
Gambar 5.28 Implementasi Antar Muka Halaman Notifications	5-19
Gambar 5.29 Implementasi Antar Muka Halaman Edit Ticket	5-20
Gambar 5.30 Implementasi Antar Muka Halaman Konfirmasi Update Ticket	5-20
Gambar 5.31 Implementasi Antar Muka Halaman Detail Tickets.....	5-21
Gambar 5.32 Implementasi Antar Muka Halaman Pemberitahuan Ketika Tidak Ada Percakapan yang Dipilih.....	5-21
Gambar 5.33 Implementasi Antar Muka Halaman Percakapan.....	5-22
Gambar 5.34 Implementasi Antar Muka Halaman Konfirmasi Akhiri Percakapan	5-22
Gambar 5.35 Implementasi Antar Muka Halaman Pesan Penutup Percakapan.....	5-23
Gambar 5.36 Implementasi Antar Muka Halaman List Ticket Recap	5-23
Gambar 5.37 Implementasi Antar Muka Halaman Detail Ticket Recap	5-24

DAFTAR ISTILAH

Istilah	Keterangan
<i>Customer</i>	Pihak yang menggunakan layanan sistem, seperti melakukan pengiriman atau pelacakan barang.
<i>Customer service</i>	Pihak yang bertugas menangani keluhan atau pertanyaan dari <i>Customer</i> melalui fitur layanan seperti live chat atau ticketing.
Real-time	Sistem yang memberikan respon secara langsung sesuai waktu kejadian tanpa jeda.
HTTPClient	Modul <i>Angular</i> untuk mengirim dan menerima data dari server melalui protokol HTTP.
RESTful API	Antarmuka layanan web berbasis HTTP yang digunakan untuk pertukaran data antara frontend dan backend.
JSON(JavaScript Object Notation)	Format data ringan berbasis teks yang digunakan untuk pertukaran data antara sistem, dalam bentuk pasangan key-value.
Fishbone	Diagram sebab-akibat yang digunakan untuk menganalisis akar permasalahan dalam suatu proses.
Angular	Sebuah kerangka kerja (framework) berbasis TypeScript yang bersifat open-source untuk pengembangan aplikasi web satu halaman yang memungkinkan pemisahan antara logika bisnis dan tampilan antarmuka secara efisien.
Dependency Injection	Pola desain perangkat lunak di mana dependensi suatu objek disediakan secara eksternal untuk meningkatkan modularitas dan memudahkan pengujian kode tanpa perlu instansiasi manual di setiap komponen.
Firebase	Platform pengembangan aplikasi berbasis awan (<i>cloud-hosted</i>) yang menyediakan layanan basis data NoSQL untuk sinkronisasi data secara instan melalui protokol nirkabel
Two-way Data Binding	Mekanisme sinkronisasi otomatis secara dua arah antara model data dan tampilan antarmuka pengguna tanpa memerlukan pemuatan ulang halaman secara manual.

DAFTAR SIMBOL

No.	Gambar	Nama	Keterangan
1.		Status Awal	Kondisi awal dari sebuah aktivitas sistem, di mana setiap diagram aktivitas memiliki satu status awal.
2.		Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
3.		Penggabungan / <i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
4.		Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
5.		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>usecase</i> .
6.		<i>Use case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil terukur untuk <i>actor</i> .
7.		<i>Association</i>	Yang menghubungkan antara <i>actor</i> dengan <i>use case</i>

BAB 1

PENDAHULUAN

1.1 Latarbelakang

Revolusi industri keempat (Industri 4.0) menekankan pentingnya integrasi teknologi digital secara sistematis dalam berbagai sektor industri, termasuk sektor logistik. Penerapan teknologi digital tidak hanya berfungsi sebagai pendukung operasional, tetapi juga sebagai strategi dalam meningkatkan daya saing perusahaan di era digital. Pemanfaatan teknologi digital diyakini mampu meningkatkan efisiensi operasional, mempercepat proses produksi dan distribusi, serta menekan biaya operasional, sehingga dapat mendukung pertumbuhan perusahaan secara berkelanjutan. Dalam sektor logistik, transformasi digital berperan penting dalam menciptakan transparansi informasi yang berkesinambungan mulai dari pemasok hingga *Customer*. Oleh karena itu, keterbukaan informasi di sepanjang rantai pasokan menjadi faktor penting dalam meningkatkan kualitas layanan serta kinerja e-logistic secara keseluruhan [PUR22].

Seiring dengan perkembangan tersebut, perusahaan penyedia jasa pengiriman perlu mengadopsi teknologi digital yang mampu mendukung proses komunikasi dan pelayanan kepada *Customer* secara cepat dan efektif. Pemanfaatan teknologi digital dalam proses komunikasi menjadi bagian penting dalam mendukung kualitas layanan pengiriman barang. PT. Ursabyte merupakan perusahaan yang bergerak di bidang teknologi informasi dan berperan sebagai penyedia jasa pengembangan sistem berbasis web bagi klien, termasuk dalam pengembangan sistem pengiriman barang. Sistem pengiriman barang berbasis web yang dikembangkan oleh PT. Ursabyte digunakan oleh klien sebagai sarana pendukung operasional pengiriman, khususnya dalam pengelolaan data pengiriman dan pemantauan proses distribusi barang. Sistem pengiriman barang yang digunakan oleh klien PT. Ursabyte tersebut telah mampu mencatat dan menyimpan data pengiriman secara terstruktur, sehingga membantu proses administrasi dan pengelolaan informasi pengiriman. Namun demikian, dalam penerapannya masih ditemukan kendala pada aspek komunikasi antara *Customer* dan *Customer service*. Sistem yang ada belum menyediakan fitur *live chat* yang memungkinkan terjadinya interaksi secara langsung antara *Customer* dan *Customer service*. Kondisi ini menyebabkan proses penyampaian informasi, pertanyaan, maupun keluhan terkait pengiriman barang belum terfasilitasi secara optimal melalui sistem yang terintegrasi.

Ketiadaan fitur *live chat* tersebut menyebabkan *Customer* mengalami kesulitan dalam memperoleh informasi terkait status pengiriman, keterlambatan pengiriman, maupun ketidaksesuaian data. *Customer* harus memanfaatkan media komunikasi lain yang tidak terintegrasi dengan sistem pengiriman barang, sehingga proses penyampaian informasi menjadi kurang optimal. Dari sisi perusahaan, kondisi ini berdampak pada meningkatnya beban kerja *Customer service* karena harus menangani pertanyaan dan keluhan pelanggan yang bersifat repetitif secara manual. Selain itu, tidak tersedianya sistem komunikasi dan pencatatan keluhan yang terintegrasi juga menyulitkan perusahaan

dalam melakukan pencatatan, pemantauan, serta evaluasi kinerja layanan *Customer service* secara terukur dan berkelanjutan.

Untuk mengatasi permasalahan tersebut, diperlukan pengembangan aplikasi pengiriman barang berbasis web yang dilengkapi dengan fitur *live chat* dan sistem *ticketing* terintegrasi. Fitur *live chat* memungkinkan terjadinya komunikasi secara real-time antara *Customer* dan *Customer service*, sehingga pertanyaan maupun keluhan yang berkaitan dengan pengiriman dapat ditangani secara langsung. Sementara itu, sistem *ticketing* berfungsi untuk mencatat setiap keluhan atau permintaan layanan sebagai tiket yang dapat dikelola berdasarkan tingkat prioritas dan status penanganannya, sehingga mempermudah proses pengelolaan serta pengukuran kinerja layanan *Customer service* secara sistematis [FAR22]. Penggunaan framework Angular dalam pengembangan aplikasi ini diharapkan mampu menghasilkan antarmuka yang responsif dan mudah digunakan, serta mendukung pengelolaan data yang terstruktur melalui konsep modularitas komponen dan *two-way data binding*, sehingga dapat meningkatkan efisiensi proses pengembangan dan kualitas layanan sistem pengiriman barang.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, masalah utama dalam penelitian tugas akhir ini adalah bagaimana merancang fitur *live chat* mendukung komunikasi secara langsung dan *Real-time* antara *Customer* dan *Customer service*?

1.3 Tujuan Tugas Akhir

Berdasarkan identifikasi masalah yang telah disampaikan sebelumnya, permasalahan utama dalam penelitian tugas akhir ini adalah mengembangkan fitur *live chat* pada sistem yang mendukung komunikasi secara langsung antara *Customer* dan *Customer service* dalam menangani kendala atau pertanyaan seputar pengiriman barang.

1.4 Lingkup Tugas Akhir

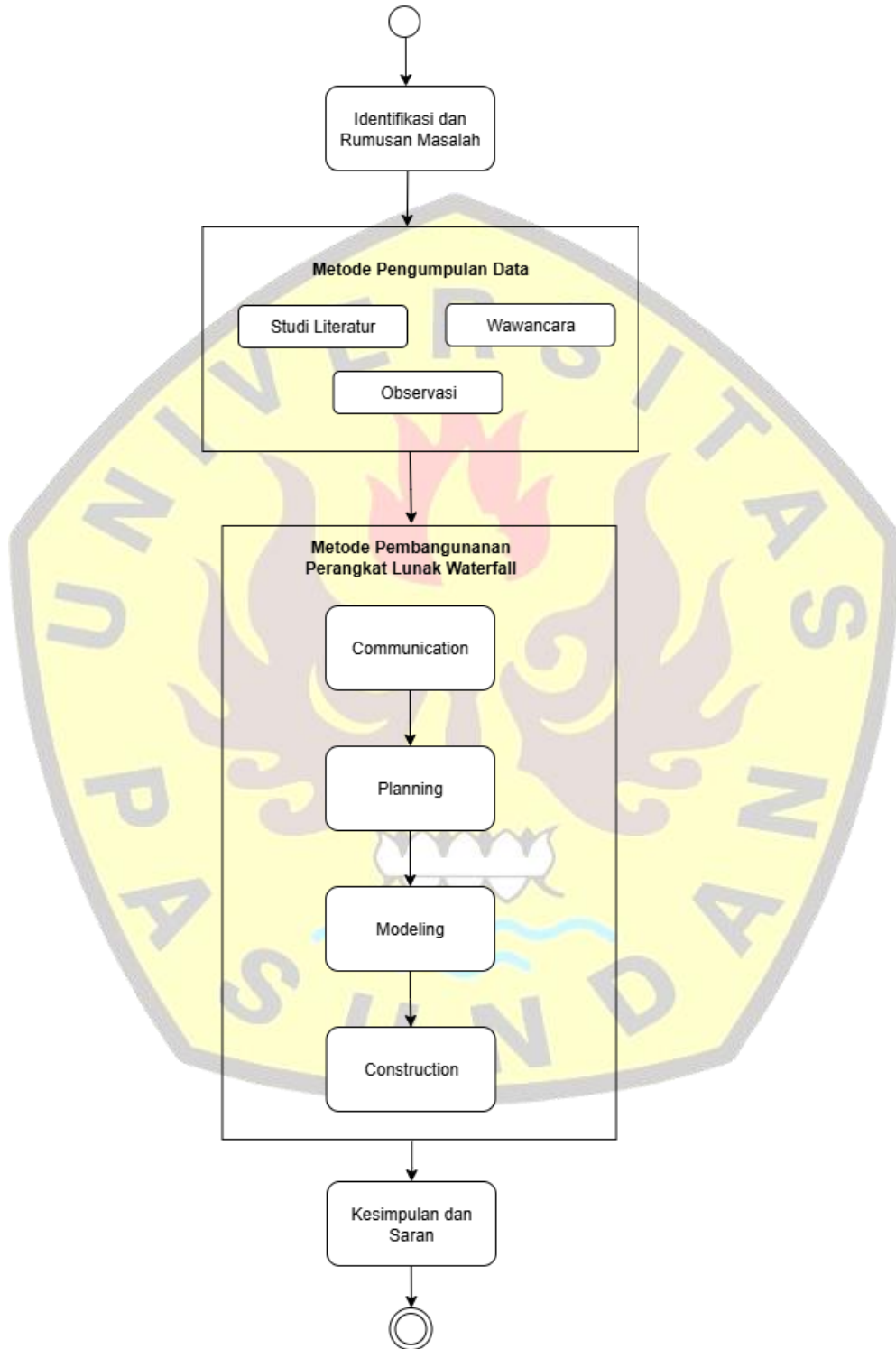
Penyelesaian Tugas Akhir ini memiliki batasan lingkup sebagai berikut :

- a. Sistem ini mengimplementasikan fitur live chat *Real-time* antara *Customer* dan *Customer service* untuk menyampaikan pertanyaan, keluhan, atau permintaan bantuan.
- b. Setiap keluhan atau kendala yang disampaikan akan dicatat sebagai satu tiket, dilengkapi informasi permasalahan *Customer* dan id resi.
- c. Sistem ticketing memungkinkan *Customer service* melihat daftar tiket yang masuk, termasuk ID Resi, status dan prioritas dari setiap tiket.
- d. Manajemen ID resi tidak termasuk dalam lingkup sistem karena tidak dikelola secara langsung oleh aplikasi.
- e. Sistem mendukung multi-login bagi *Customer service*, sehingga lebih dari satu petugas dapat mengaksesnya secara bersamaan.
- f. Setiap tiket hanya dapat ditangani oleh satu *Customer service* untuk mencegah tumpang tindih dalam penanganan.
- g. Penelitian ini menggunakan Metode *waterfall* yang hanya melibatkan pengerjaan hingga tahap *construction*, dan tahap *deployment* tidak dilakukan.
- h. Antarmuka pengguna dikembangkan menggunakan *Angular*, yang mendukung tampilan responsif dan komunikasi API melalui *HTTPClient*.
- i. Backend menggunakan Node.js untuk manajemen akun, autentikasi pengguna berbasis peran (role-based), serta penyediaan *RESTful API* yang mengelola data pengiriman barang, informasi user, sistem tiket, dan interaksi live chat.
- j. *Firebase* digunakan sebagai database *Real-time* untuk menyimpan pesan percakapan secara langsung serta memastikan sinkronisasi data antara *Customer* dan *Customer service* secara *real-time*.



1.5 Metodologi Penyelesaian Tugas Akhir

Metodologi yang digunakan dalam penyelesaian tugas akhir ini ditunjukkan pada Gambar 1.1 Metodologi Penyelesaian Tugas Akhir berikut ini :



Gambar 1.1 Metodologi Penyelesaian Tugas Akhir

Berdasarkan gambar 1.1 Metodologi penelitian diatas menjelaskan dari masing-masing tahapan metodologi penelitian tugas akhir sebagai berikut :

1. Identifikasi dan Rumusan Masalah

Pada tahap ini, penulis memulai langkah penelitian dengan mengidentifikasi masalah yang ada, kemudian merumuskan masalah berdasarkan identifikasi tersebut. Hal ini bertujuan untuk menghasilkan solusi yang relevan dan dapat memberikan kontribusi signifikan terhadap kelanjutan penelitian.

2. Metode Pengumpulan Data

Pada tahap ini, dilakukan pengumpulan data yang menjadi bahan informasi penelitian. Ada beberapa metode pengumpulan data yang digunakan dalam penelitian, antara lain:

a. Studi Literatur

Metode pengumpulan data dilakukan dengan menelusuri dan mempelajari berbagai referensi, baik dari buku, jurnal, skripsi, maupun sumber lain yang memiliki keterkaitan dengan objek penelitian. Selanjutnya, penulis melakukan kajian pustaka terhadap penelitian-penelitian sebelumnya yang relevan dengan topik tugas akhir ini. Tujuan dari kajian pustaka tersebut adalah untuk mengetahui keunggulan dan keterbatasan dari metode yang telah digunakan pada penelitian terdahulu, sehingga dapat dijadikan acuan dalam menentukan metode yang digunakan dalam penelitian ini.

b. Wawancara

Pengumpulan data dilakukan melalui wawancara dengan Kak Abiyan Bagus Baskoro selaku pembimbing lapangan, yang juga merupakan staf yang memiliki tanggung jawab atas informasi mengenai prosedur pengelolaan pengiriman barang. Dari hasil wawancara tersebut diperoleh informasi terkait alur pengiriman barang, penanganan kendala, dan prosedur pengiriman yang berlaku.

c. Observasi

Observasi dilakukan dengan cara melihat dan mengamati masalah-masalah yang terjadi untuk memperoleh data yang kita peroleh.

3. Metode Pembangunan Perangkat Lunak

Metode pembangunan aplikasi web yang digunakan dalam penelitian ini adalah metode Waterfall. Metode ini dipilih karena memiliki alur pengembangan yang terstruktur dan berurutan, dimulai dari tahap komunikasi (communication) hingga tahap implementasi dan pemeliharaan (deployment). Tahapan dalam metode Waterfall yang digunakan pada penelitian ini dijelaskan sebagai berikut:

a. *Communication*

Tahap awal dilakukan dengan mengidentifikasi kebutuhan tambahan dan perubahan pada sistem yang telah ada. Proses ini melibatkan analisis terhadap fitur-fitur eksisting, pengumpulan masukan dari pengguna, serta peninjauan dokumentasi lama. Informasi yang diperoleh kemudian digunakan untuk merumuskan spesifikasi kebutuhan baru yang akan dikembangkan.

b. *Planning*

Pada tahap perencanaan, dilakukan analisis struktur sistem lama untuk menentukan bagian mana yang dapat dipertahankan, dimodifikasi, atau ditambahkan. Rencana pengembangan mencakup penjadwalan pekerjaan, estimasi sumber daya, serta penentuan arsitektur teknis yang akan tetap digunakan atau ditingkatkan agar sesuai dengan kebutuhan baru.

c. *Modeling*

Tahap modeling difokuskan pada pembuatan rancangan teknis yang menyesuaikan sistem eksisting dengan fitur baru. Hal ini meliputi desain antarmuka pengguna yang diperbarui, penyesuaian alur proses, serta pemodelan basis data untuk memastikan integrasi berjalan dengan baik. Desain harus tetap kompatibel dengan struktur lama guna meminimalkan gangguan pada sistem yang sudah beroperasi.

d. *Construction*

Proses construction mencakup implementasi kode untuk fitur baru serta modifikasi pada komponen yang sudah ada. Pengembangan dilakukan secara bertahap agar tidak mengganggu fungsi utama sistem lama. Setiap bagian yang diperbarui diuji secara unit, kemudian dilakukan integrasi untuk memastikan kompatibilitas antara modul baru dan sistem eksisting.

4. Kesimpulan dan Saran

Setelah pembangunan aplikasi selesai, proses selanjutnya adalah melakukan pengambilan kesimpulan dan memberikan saran berdasarkan hasil analisis dan perancangan, yang bertujuan untuk mengevaluasi pengembangan penelitian di masa mendatang.

1.6. Sistematika Penulisan Tugas Akhir

Sistematika penulisan tugas akhir bertujuan untuk memberikan gambaran umum mengenai isi laporan penelitian yang dilakukan. Sistematika penulisan laporan ini terbagi menjadi enam bab, dengan sub pokok bahasan masing-masing bab sebagai berikut :

BAB 1. Pendahuluan

Bab ini berisi penjabaran umum mengenai masalah yang diangkat selama proses penelitian. Pembahasan dalam bab ini mencakup latar belakang penelitian, identifikasi masalah, tujuan tugas akhir, ruang lingkup penelitian, metodologi tugas akhir, serta penjelasan mengenai sistematika penulisan tugas akhir.

BAB 2. Landasan Teori

Bab ini memuat teori-teori dasar yang relevan dan mendukung topik penelitian, diperoleh dari berbagai sumber seperti buku, jurnal, skripsi, dan referensi lainnya. Selain itu, disajikan pula tinjauan pustaka yang membahas penelitian-penelitian sebelumnya yang berkaitan dengan metode atau topik yang digunakan dalam tugas akhir ini.

BAB 3. Skema Penelitian

Bab ini menjelaskan tahapan-tahapan penelitian berdasarkan metode yang diterapkan. Di dalamnya dibahas alur pengerjaan tugas akhir, rumusan masalah, kerangka pemikiran teoritis, objek yang diteliti, waktu pelaksanaan penelitian, serta profil instansi atau lokasi tempat penelitian dilakukan.

BAB 4. Analisis dan Perancangan

Bab ini mendeskripsikan mengenai usulan model implementasi, yang meliputi analisis dan perancangan sebagai pedoman pembangunan aplikasi yang berisi tentang identifikasi masalah, analisa prosedur yang sedang berjalan, analisis kebutuhan sistem, spesifikasi kebutuhan sistem, analisis fungsional sistem, perancangan sequence diagram, perancangan arsitektur, perancangan struktur menu, perancangan basis data, dan perancangan antarmuka aplikasi.

BAB 5. Implementasi dan Pengujian

Bab ini mendeskripsikan mengenai hasil implementasi dan pengujian aplikasi, yang meliputi implementasi perangkat keras (*hardware*), implementasi perangkat lunak (*software*), implementasi basis data, dan implementasi antarmuka aplikasi. Pada bab ini juga dijabarkan hasil uji coba menggunakan metode black box untuk memastikan aplikasi yang dibangun sesuai dengan perencanaan awal.

BAB 6. Kesimpulan dan Saran

Bab ini berisi kesimpulan yang diperoleh dari hasil penelitian tugas akhir, serta saran-saran untuk pengembangan selanjutnya agar dapat dilakukan perbaikan-perbaikan pada masa yang akan datang

BAB 2

LANDASAN TEORI DAN PENELITIAN TERDAHULU

2.1. Teori Pendukung

2.1.1 Pengembangan

Pengembangan perangkat lunak merupakan sebuah proses yang mencakup kegiatan membuat, mengembangkan, serta memelihara sebuah perangkat lunak agar dapat berfungsi sesuai dengan tujuan yang diinginkan. Dalam proses pengembangan ini, terdapat beberapa tahapan penting yang harus dilalui secara sistematis, mulai dari analisis kebutuhan yang bertujuan untuk memahami dan mengidentifikasi kebutuhan pengguna atau bisnis secara mendetail. Selanjutnya, dilakukan tahap desain yang berfokus pada perancangan struktur dan arsitektur perangkat lunak yang akan dibuat. Setelah desain selesai, tahap implementasi atau pengkodean dilakukan untuk mewujudkan rancangan tersebut menjadi program yang berjalan. Tahap berikutnya adalah pengujian, di mana perangkat lunak diuji guna memastikan bahwa semua fungsi berjalan dengan benar dan sesuai spesifikasi yang telah ditetapkan. Terakhir, proses pemeliharaan dilakukan untuk memperbaiki, memperbarui, dan meningkatkan perangkat lunak agar tetap relevan dan dapat memenuhi kebutuhan pengguna yang mungkin berubah seiring waktu. Secara keseluruhan, pengembangan perangkat lunak dilakukan dengan tujuan utama untuk memenuhi kebutuhan bisnis maupun kebutuhan pengguna dalam menjalankan tugas-tugasnya secara efektif dan efisien [ZEI23].

2.1.2 Aplikasi

Aplikasi adalah perangkat lunak yang dikembangkan oleh suatu perusahaan teknologi untuk membantu individu dalam menyelesaikan tugas-tugas tertentu. Kemudian ada pula pendapat lain yang menyatakan persamaan mengenai pengertian aplikasi, yaitu “Application software yang dibuat oleh suatu perusahaan computer untuk mengerjakan tugas-tugas tertentu”. Jadi dari dua sumber di atas dapat disimpulkan bahwa aplikasi adalah suatu program komputer yang digunakan untuk mengerjakan tugas-tugas tertentu yang dibuat oleh perusahaan komputer [FAU21].

2.1.3 Live Chat

Live chat merupakan salah satu fitur penting dalam platform digital yang dirancang untuk memfasilitasi interaksi langsung antara pengguna dengan sistem maupun antar sesama pengguna. Kehadiran fitur ini memiliki peran strategis dalam mendukung komunikasi yang cepat, efisien, dan responsif, khususnya pada sektor layanan publik dan bisnis. Tidak hanya sebagai media pertukaran pesan, live chat juga berfungsi sebagai instrumen pelayanan yang dapat meningkatkan kepuasan pengguna secara signifikan [RIZ23].

Agar suatu sistem dapat dikategorikan sebagai sistem live chat yang efektif, terdapat beberapa aspek fundamental yang perlu diperhatikan. Pertama, aspek kehandalan dan kesiapan (reliability & availability) mengacu pada kemampuan sistem untuk melaksanakan tugas sesuai dengan arahan pengguna dengan konsisten dan terus menerus. Hal ini mencerminkan bahwa sistem tersebut dapat

diandalkan dalam berbagai kondisi operasional. Kedua, aspek responsivitas (*responsiveness*), yakni kemampuan sistem dalam menyampaikan informasi yang berkaitan dengan proses pelaksanaan tugas serta kesiapannya untuk melayani pengguna setiap saat [ALV23].

Secara teknis, sistem live chat yang dikembangkan dalam penelitian ini tidak terhubung secara langsung dengan layanan messenger pihak ketiga, seperti WhatsApp karena adanya perbedaan arsitektur sistem dan kebijakan integrasi yang diterapkan oleh masing-masing platform.

2.1.4 Pengiriman Barang

Pengiriman merupakan aktivitas yang berfungsi untuk mendistribusikan produk, baik berupa barang maupun jasa, dari produsen ke tangan konsumen. Pengiriman juga termasuk bagian dari kegiatan pemasaran yang bertujuan untuk mempermudah proses penyaluran produk agar dapat sampai kepada konsumen secara tepat. Berdasarkan definisi tersebut, manfaat utama dari pengiriman adalah sebagai sarana alih kepemilikan suatu barang atau jasa dari produsen kepada konsumen. Proses pengiriman ini juga membentuk suatu alur dalam saluran pemasaran atau dikenal sebagai arus distribusi pengiriman yang mendukung kelancaran pemasaran produk [RAH22].

Dalam sistem logistik, Logistik merupakan elemen penting dalam rantai pasok yang berperan dalam perencanaan, pelaksanaan, dan pengendalian proses pengadaan, pengelolaan, serta penyimpanan barang dan informasi secara efektif dan efisien. Proses ini berlangsung dari titik awal hingga mencapai konsumen akhir, dengan tujuan utama memenuhi kebutuhan *Customer*. Dengan kata lain, logistik mencakup segala bentuk bahan, barang, peralatan, atau sarana yang menunjang aktivitas organisasi guna mencapai tujuan yang telah ditetapkan [KAS16]. Pengiriman merupakan bagian integral dari sistem distribusi yang melibatkan beberapa aktor utama, yaitu pengirim (*shipper*), penyedia jasa logistik (*logistics service provider*), dan penerima (*receiver*). Proses distribusi dimulai dari pengirim yang menyiapkan barang, dilanjutkan oleh penyedia jasa logistik yang menangani proses transportasi, pelacakan, penyimpanan sementara (jika diperlukan), dan pengelolaan dokumen pengiriman. Selanjutnya, barang dikirimkan ke penerima akhir. Setiap aktor memiliki peran penting dalam memastikan kelancaran dan ketepatan waktu pengiriman, serta menjamin integritas barang selama proses berlangsung. Dalam sistem logistik modern, proses ini sering didukung oleh teknologi informasi untuk memantau status pengiriman secara *Real-time*, meningkatkan transparansi, dan mempercepat pengambilan keputusan apabila terjadi kendala di lapangan. Pengirim dan penerima adalah pihak-pihak yang memerlukan pergerakan produk antara dua lokasi dalam rantai pasok, dan keberhasilan transaksi tersebut membutuhkan pergerakan barang-barang dari lokasi asal ke lokasi tujuan dengan biaya transportasi yang paling rendah. Isu-isu penting yang harus disolusikan oleh pengirim dan penerima meliputi waktu pengambilan dan pengantaran barang, waktu singgah, kehilangan dan kerusakan barang, penagihan, dan keakuratan informasi. Perusahaan penyedia jasa transportasi (*carrier* dan *agent*) berperan sebagai pihak yang menyelenggarakan transportasi barang dan bertanggung jawab atas efisiensi operasional dalam proses distribusi [SUS16].

2.1.5 *Firestore*

Firestore merupakan layanan yang dimiliki oleh Google dan digunakan untuk mempermudah proses pengembangan aplikasi. Dengan menggunakan *Firestore*, pengembang tidak perlu membangun fitur backend dan infrastruktur dari awal, sehingga dapat lebih fokus untuk menciptakan aplikasi yang berkualitas tanpa harus mengeluarkan upaya yang besar. *Firestore* menggunakan tipe basis data NoSQL milik Google yang bersifat fleksibel dan mendukung struktur data semi-terstruktur.

Pada penelitian ini, fitur *Firestore* yang digunakan adalah *Firestore Authentication*, yang digunakan untuk mengelola akun pengguna, baik *Customer*, *Customer service*, maupun admin. Hak akses pengguna dapat diatur secara fleksibel sesuai peran masing-masing. Dan *Firestore Cloud Firestore*, yang digunakan untuk menyimpan data percakapan secara *Real-time*, mencatat riwayat pesan, serta mendokumentasikan detail laporan *Customer* dalam bentuk tiket [ROS20].

2.1.6 NoSQL

NoSQL merupakan singkatan dari “*Not Only SQL*” dan sering juga disebut sebagai “*Non-relational database*” merupakan istilah yang dikenal dalam teknologi komputasi untuk merujuk kepada kelas yang luas dari database management system yang diidentifikasi tidak mematuhi aturan pada model sistem manajemen basis data relasional yang banyak digunakan. Database NoSQL dibuat dengan tujuan khusus yaitu untuk model data spesifik dan memiliki skema fleksibel untuk membuat aplikasi modern. Dalam penelitian ini digunakan *Firestore* termasuk dalam kategori basis data NoSQL tipe document-based, di mana data disimpan dalam bentuk dokumen yang menyerupai format JSON.

Kelebihan NoSQL adalah kemampuannya dalam menyimpan data terstruktur, semi-terstruktur, maupun tidak terstruktur secara efisien, terutama dalam skala besar (*big data*). Format penyimpanan yang fleksibel, seperti dokumen *JSON (JavaScript Object Notation)*, membuat NoSQL lebih mudah diintegrasikan dengan bahasa pemrograman modern, terutama yang berorientasi objek, sehingga mempermudah proses pengaksesan dan manipulasi data dalam pengembangan aplikasi. Meskipun NoSQL tidak secara langsung menerapkan prinsip OOP (Object-Oriented Programming), format data yang digunakan, seperti dokumen JSON, memiliki struktur yang serupa dengan objek dalam OOP [ROS20].

2.1.7 Sistem Ticketing

Sistem ticketing merupakan suatu perangkat lunak terintegrasi yang digunakan untuk menghimpun, mengelola, dan memantau seluruh permintaan layanan dari *Customer*, baik berupa pertanyaan, keluhan, maupun permintaan dukungan lainnya. Dengan implementasi sistem ticketing, tim

Customer service dapat melakukan prioritisasi, pelacakan, serta tindak lanjut terhadap setiap permintaan *Customer* secara lebih terpusat dalam satu platform [FAR22].

Pada sistem yang dirancang dalam tugas akhir ini, ticketing diintegrasikan secara langsung dengan fitur live chat. Melalui fitur ini, *Customer* dapat menyampaikan laporan atau pertanyaan secara langsung kepada *Customer service* tanpa perlu perantara pencatatan manual. Setiap keluhan yang dikirim secara otomatis dicatat menjadi tiket di dalam sistem, sehingga data laporan *Customer* terdokumentasi dengan lebih cepat, akurat, dan *Real-time*. Setelah tiket tercatat, sistem akan memproses validasi serta mendistribusikan tiket tersebut ke bagian yang sesuai berdasarkan kategori laporan.

Dalam penerapannya, sistem ticketing ini dilengkapi dengan pengelompokan berdasarkan *prioritas* dan *status* pada setiap tiket yang masuk. Prioritas ditentukan oleh admin yang mencerminkan tingkat urgensi laporan, yang diklasifikasikan menjadi *High* dan *Low*. Laporan dengan prioritas *High* menunjukkan permasalahan yang bersifat kritikal dan memerlukan penanganan segera, sedangkan *Low* memiliki urgensi lebih rendah. Sementara itu, status tiket menggambarkan tahapan penanganan, yaitu *New* (baru diterima), *In Progress* (sedang diproses), dan *Done* (telah selesai). Penggunaan skema ini bertujuan untuk mempermudah manajemen tiket secara terstruktur, serta mendukung pengambilan keputusan dalam peningkatan layanan pelanggan.

2.1.8 Real-time

Real-time merupakan jenis sistem yang dituntut untuk memberikan respons secara tepat dan akurat dalam batas waktu tertentu yang telah ditetapkan, atau sering disebut dengan deadline. Dalam sistem ini, keberhasilan operasi tidak hanya diukur dari kebenaran logis hasil keluarannya, tetapi juga dari sejauh mana sistem tersebut mampu menghasilkan respons dalam waktu yang sesuai atau tepat waktu. Artinya, meskipun output yang dihasilkan secara logika benar, jika tidak disampaikan dalam waktu yang telah ditentukan, maka sistem dianggap gagal memenuhi fungsinya sebagai sistem *Real-time*. Oleh karena itu, sistem ini sangat bergantung pada ketepatan waktu [MAN20].

Dalam konteks penelitian ini, implementasi *Real-time* tampak pada fitur live chat yang menuntut komunikasi antara *Customer* dan *Customer service* dituntut untuk secara langsung tanpa adanya penundaan waktu. Respon yang tidak cepat dapat mengganggu kenyamanan pengguna dan menurunkan kualitas layanan. Oleh karena itu, Ketepatan waktu dalam proses pengiriman dan penerimaan pesan ini merupakan faktor utama dalam menjaga kelancaran komunikasi serta memenuhi harapan pengguna terhadap layanan yang cepat dan tepat.

2.1.9 Framework

Web application framework adalah sebuah kerangka yang didesain untuk mendukung pembuatan aplikasi web. Sebuah framework menyediakan standar atau cara untuk membuat sebuah aplikasi dengan dukungan fitur-fitur yang telah disediakan. Seperti bagaimana bekerja dengan database, bagaimana mengelola template atau tampilan dan lain-lain [ISK23].

2.1.9.1 *Angular*

Angular adalah sebuah framework yang menawarkan berbagai kemudahan dalam pengembangan aplikasi web. Salah satu aspek penting dari *Angular* adalah pemanfaatan HTML sebagai template yang dilengkapi dengan berbagai fitur khusus. *Angular* mengadopsi pendekatan deklaratif dalam membangun antarmuka pengguna, di mana pengembang menyusun struktur aplikasi dengan cara menghubungkan komponen-komponen yang ada. Sistem modul dalam *Angular* memungkinkan pengembang untuk membagi kode menjadi modul-modul terpisah, sehingga mempermudah pengembangan dan pemeliharaan aplikasi yang kompleks. Modul-modul ini memungkinkan pengembang untuk mengimpor fungsi dan komponen yang dibutuhkan ke dalam aplikasi. *Angular* adalah generasi terbaru dari framework *AngularJS* yang memiliki berbagai perbaikan. *AngularJS* dibangun menggunakan bahasa JavaScript yang memisahkan data dari antarmuka, sementara *Angular* dibangun dengan menggunakan bahasa TypeScript yang mengadopsi arsitektur berbasis komponen, yang memungkinkan adanya data binding atau penempelan data. Penggunaan bahasa TypeScript oleh Google dalam pembuatan framework *Angular* didorong oleh peningkatan strukturisasi kode melalui konsep type safety yang diterapkan. Performa *Angular* tercatat tujuh kali lebih cepat dibandingkan *AngularJS*, karena *Angular* memanipulasi struktur DOM secara langsung tanpa perlu membuat dokumen HTML terlebih dahulu [KIR16].

Pemanfaatan *Angular* dalam pengembangan fitur live chat dan ticketing memberikan sejumlah keunggulan, khususnya dalam membangun antarmuka pengguna yang terstruktur, efisien, dan interaktif. *Angular* mendukung pengembangan komponen secara modular, seperti komponen daftar percakapan yang menampilkan riwayat pesan, jendela obrolan yang menampilkan isi komunikasi secara waktu nyata (*Real-time*), formulir tiket untuk pelaporan kendala, serta halaman histori interaksi yang mencatat proses penyelesaian permasalahan. Komponen-komponen tersebut dikembangkan secara independen dan dapat digunakan kembali di berbagai bagian aplikasi. Salah satu fitur penting *Angular* adalah *Two-way Data Binding*, yaitu mekanisme sinkronisasi otomatis antara data model dan tampilan antarmuka. Dengan adanya fitur ini, perubahan data yang terjadi pada sisi pengguna, seperti saat mengetik pesan, akan langsung tercermin pada tampilan antarmuka secara dinamis tanpa memerlukan pemuatan ulang halaman (manual refresh).

Hal ini meningkatkan interaktivitas dan kenyamanan dalam penggunaan sistem, terutama pada fitur komunikasi seperti live chat. *Angular* juga menyediakan fitur routing bawaan yang memungkinkan navigasi antar halaman dalam aplikasi dilakukan secara efisien dan tanpa gangguan. *Customer* dapat diarahkan dari halaman utama ke halaman tiket atau ke ruang percakapan tertentu melalui struktur URL, sehingga mendukung pengelolaan tampilan yang terorganisir dan meningkatkan pengalaman pengguna. Selain itu, *Angular* menerapkan konsep *Dependency Injection* (DI), yaitu suatu pendekatan dalam pengelolaan dependensi di mana objek atau layanan yang dibutuhkan suatu komponen disediakan dari luar. Melalui mekanisme ini, layanan seperti *ChatService* dapat digunakan secara langsung oleh berbagai komponen aplikasi melalui penerapan *Dependency Injection*, yang memungkinkan layanan tersebut tersedia tanpa perlu dibuat secara manual dalam setiap komponen.

Secara keseluruhan, penerapan *Two-way Data Binding*, routing, dan *Dependency Injection* dalam *Angular* menjadi komponen penting dalam membangun sistem live chat dan ticketing yang bersifat dinamis, responsif, dan berkelanjutan. Ketiga fitur tersebut tidak hanya meningkatkan kualitas interaksi *Customer*, tetapi juga memperkuat arsitektur aplikasi yang modular, mudah dikembangkan, dan sesuai dengan kebutuhan sistem berskala besar [DAY18].

2.1.10 Website

Dalam perkembangan teknologi informasi yang pesat, website mengalami kemajuan signifikan, terutama dalam pengolahan konten. Website dikategorikan berdasarkan fungsi, karakteristik, gaya, dan bahasa pemrograman yang digunakan, dengan dua jenis utama: dinamis dan statis. Website adalah kumpulan halaman web yang saling terhubung, berisi teks, gambar, animasi, audio, dan video, serta dapat diakses melalui internet untuk kepentingan pribadi, organisasi, atau perusahaan. Dokumentasi ini tersimpan di web server yang tersebar di seluruh dunia dan terhubung melalui jaringan internet. World Wide Web (WWW) adalah layanan yang memungkinkan pengguna mengakses berbagai informasi, dari yang sederhana hingga kompleks, serta gratis maupun berbayar. Website sendiri merupakan kumpulan halaman yang menampilkan informasi dalam berbagai format dan saling terhubung melalui hyperlink, dengan sifat statis atau dinamis [NOV22].

2.1.11 Hypertext Markup Language

Hypertext Markup Language (HTML) merupakan skrip pemrograman yang berfungsi untuk mengatur penyajian informasi di internet serta memungkinkan pengguna berpindah dari satu halaman ke halaman lainnya. HTML dikembangkan oleh Tim Berners-Lee saat bekerja di CERN dan pertama kali dikenal luas melalui browser Mosaic. Sejak awal tahun 1990, HTML mengalami perkembangan pesat, di mana setiap versi terbaru selalu membawa peningkatan fitur dan kemampuan dibandingkan versi sebelumnya [NOV22].

2.1.12 Node.JS

Node.js merupakan platform sisi server yang dibangun di atas mesin JavaScript V8 milik Google Chrome dan dikembangkan oleh Ryan Dahl pada tahun 2009. Teknologi ini bersifat open source, sepenuhnya gratis, dan banyak digunakan oleh pengembang di seluruh dunia untuk membangun aplikasi server serta aplikasi berbasis jaringan. Node.js sendiri berfungsi sebagai runtime environment untuk JavaScript yang menyediakan seluruh komponen yang diperlukan agar program yang ditulis dalam bahasa tersebut dapat dijalankan.

Express.js merupakan framework yang bersifat minimalis dan fleksibel serta berjalan pada Node.js. Framework ini memiliki dokumentasi yang lengkap dan mudah dipahami sehingga mempermudah proses pengembangan berbagai jenis produk, baik aplikasi web maupun *RESTful API*. Selain itu, Express.js juga dapat dijadikan dasar untuk membangun framework web yang lebih kompleks seperti *sails.js*, MEAN (MongoDB, Express.js, *Angular.js*, Node.js), maupun MERN (MongoDB, Express.js, React.js, Node.js).

Express.js memiliki sejumlah keunggulan yang menjadikannya populer di kalangan pengembang web. Framework ini mendukung pembuatan middleware yang berfungsi sebagai mekanisme untuk memfilter setiap permintaan HTTP yang masuk ke aplikasi. Express.js juga mendukung berbagai metode HTTP seperti POST, GET, PUT, DELETE, OPTION, dan HEAD. Selain itu, Express.js memberikan kemudahan bagi pengembang untuk memilih template engine yang sesuai dengan kebutuhan. Framework ini dapat mengelola file statis seperti CSS dan JavaScript. Sifatnya yang fleksibel memungkinkan Express.js untuk disesuaikan sehingga dapat memenuhi berbagai kebutuhan pengembangan aplikasi.

2.1.13 Modular Monolith Architecture

Modular Monolith Architecture merupakan pendekatan arsitektur perangkat lunak di mana sistem dibangun sebagai satu kesatuan aplikasi yang dideploy secara tunggal, namun secara internal diorganisasikan ke dalam beberapa modul berdasarkan fungsi bisnis tertentu. Setiap modul memiliki tanggung jawab yang jelas serta dipisahkan secara logis, seperti modul autentikasi, manajemen pengguna, layanan live chat, dan pengelolaan ticketing. Pendekatan ini bertujuan untuk meningkatkan keteraturan struktur kode, mempermudah proses pemeliharaan dan pengembangan sistem, serta meminimalkan tingkat ketergantungan antar modul tanpa menimbulkan kompleksitas tambahan yang umumnya terdapat pada arsitektur sistem terdistribusi [QOR25].

Pada sistem ini, backend dikembangkan menggunakan Express.js dengan modul-modul terpisah berupa controller dan service, seperti Auth, Chat, dan Ticket. Seluruh modul berjalan dalam satu proses aplikasi, menggunakan *Firebase Authentication* dan *Firestore* sebagai layanan backend dan basis data bersama. Interaksi antar modul dilakukan melalui pemanggilan fungsi dan akses basis data secara

langsung, bukan melalui komunikasi antar layanan. Arsitektur ini juga menerapkan pendekatan Layered (3-Tier) Architecture untuk memisahkan logika presentasi, logika, dan data, sehingga meningkatkan keteraturan sistem, efisiensi operasional, dan kemudahan pengembangan, tanpa menimbulkan kompleksitas microservices.

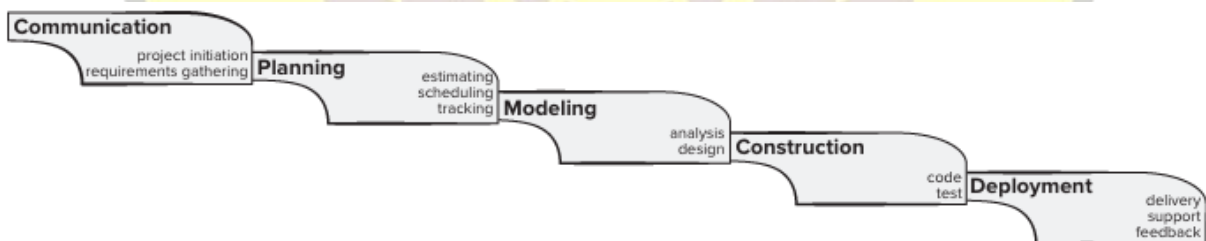
2.1.14 Role Based Access Control (RBAC)

Kontrol terhadap akses (access control) menjadi salah satu solusi ideal yang dapat diterapkan pada sistem yang memiliki isu keamanan data, terutama terkait otorisasi setiap peran yang ada di dalam sistem. Salah satu mekanisme access control yang banyak digunakan adalah Role-Based Access Control (RBAC), yang menawarkan fleksibilitas dan tingkat keamanan yang dapat diandalkan [YUR23].

RBAC memungkinkan setiap modul maupun informasi dalam sistem diatur dan dibatasi hak aksesnya sesuai dengan peran stakeholder yang telah ditentukan. Dengan demikian, sistem dapat menjaga keamanan dan privasi data, sekaligus memberikan kontrol yang lebih cermat terhadap aktivitas setiap pengguna. Penerapan RBAC terbukti efektif, terutama pada sistem dengan struktur peran stakeholder yang kompleks dan membutuhkan kolaborasi yang tinggi [KHA24].

2.1.15 Metode Waterfall

Metode Waterfall merupakan model pengembangan sistem yang digunakan dalam penelitian ini. Tahapan dari metode tersebut ditunjukkan pada Gambar 2.1 Metode Waterfall berikut ini :



Gambar 2.1 Metode Waterfall

Berdasarkan Gambar 2.1, metode *Waterfall* yang juga dikenal sebagai *linear sequential model* merupakan salah satu model proses pengembangan perangkat lunak yang bersifat klasik dan preskriptif. Model ini menekankan alur kerja yang bersifat berurutan, di mana setiap tahapan harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahapan berikutnya. Menurut Pressman [PRE20], tahapan utama dalam metode *Waterfall* meliputi:

1. *Communication*

Tahap *communication* merupakan tahap awal yang berfokus pada proses komunikasi antara pengembang dan pemangku kepentingan (*stakeholder*) untuk memperoleh pemahaman yang menyeluruh mengenai kebutuhan sistem. Tahap ini bertujuan untuk mengidentifikasi

kebutuhan fungsional dan non-fungsional perangkat lunak secara jelas agar permasalahan yang dihadapi pengguna dapat terdefinisi dengan baik.

Pada penelitian ini, tahap *communication* dilakukan melalui proses komunikasi dengan pihak terkait untuk memperoleh gambaran mengenai sistem pengiriman barang yang digunakan oleh klien PT. Ursabyte. Kegiatan yang dilakukan meliputi identifikasi alur kerja sistem yang berjalan, permasalahan yang muncul pada aspek komunikasi antara *Customer* dan *Customer service*, serta pengumpulan kebutuhan pengguna terkait pengembangan fitur live chat dan sistem ticketing. Hasil dari tahap ini berupa dokumen kebutuhan sistem yang menjadi acuan pada tahapan berikutnya.

2. *Planning*

Tahap *planning* bertujuan untuk menyusun rencana pengembangan perangkat lunak secara sistematis. Perencanaan diperlukan untuk memperkirakan sumber daya yang dibutuhkan, menyusun jadwal aktivitas, serta mengidentifikasi risiko yang berpotensi muncul selama proses pengembangan perangkat lunak.

Pada tahap ini, perencanaan dilakukan dengan menganalisis sistem yang telah berjalan untuk menentukan bagian sistem yang dapat dipertahankan, disesuaikan, atau dikembangkan lebih lanjut sesuai dengan kebutuhan baru. Selain itu, disusun rencana pengembangan yang mencakup penjadwalan pekerjaan, estimasi waktu dan sumber daya, serta penentuan keputusan teknis tingkat tinggi terkait arsitektur sistem yang akan digunakan atau ditingkatkan. Tahap perencanaan ini bertujuan agar proses pengembangan sistem dapat berjalan secara terarah dan sesuai dengan ruang lingkup penelitian.

3. *Modelling*

Tahap *modelling* merupakan tahap penerjemahan kebutuhan sistem ke dalam bentuk model analisis dan desain. Tahap ini berfungsi sebagai jembatan antara kebutuhan pengguna dan implementasi sistem, sehingga pengembang memiliki gambaran yang jelas mengenai struktur dan perilaku sistem yang akan dibangun.

Pada tahap ini, dilakukan perancangan sistem yang meliputi pemodelan alur proses sistem, struktur data, serta hubungan antar komponen sistem. Selain itu, dirancang arsitektur perangkat lunak yang mencakup desain antarmuka pengguna, desain basis data, serta desain komponen sistem yang mendukung pengembangan fitur live chat dan sistem ticketing. Hasil dari tahap *modelling* digunakan sebagai acuan dalam proses implementasi sistem.

4. *Construction*

Tahap *construction* merupakan tahap implementasi dari desain sistem yang telah dibuat. Tahap ini mencakup proses pengkodean (*coding*) dan pengujian (*testing*) untuk memastikan bahwa perangkat lunak berjalan sesuai dengan kebutuhan yang telah ditentukan.

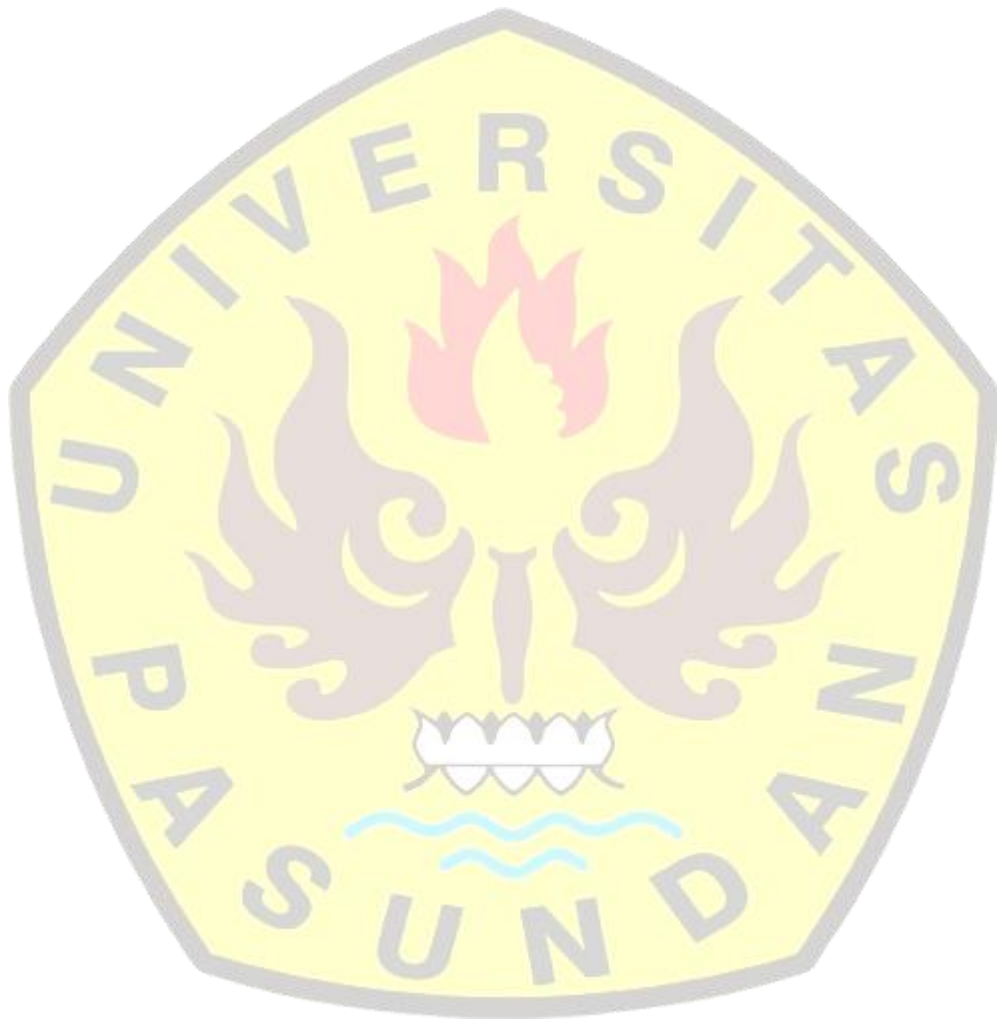
Pada tahap ini, dilakukan pengembangan aplikasi pengiriman barang berbasis web menggunakan framework Angular berdasarkan hasil perancangan yang telah dibuat. Proses pengujian dilakukan untuk memastikan setiap fungsi sistem, termasuk fitur live chat dan sistem ticketing, dapat berjalan sesuai dengan spesifikasi kebutuhan yang telah ditentukan.

5. *Deployment*

Tahap *deployment* merupakan tahap penempatan perangkat lunak ke dalam lingkungan operasional dan penggunaannya oleh pengguna akhir. Namun, pada penelitian ini tahap *deployment* tidak dilaksanakan karena pengembangan sistem difokuskan pada perancangan dan implementasi fitur hingga tahap pengujian. Oleh karena itu, tahap *deployment* dijadikan sebagai batasan penelitian dan tidak dibahas lebih lanjut.

2.1.16 Metode Pengujian *Black-box*

Pengujian ini memberikan gambaran mengenai berbagai kondisi masukan serta melakukan evaluasi terhadap fungsi program yang diuji. Untuk menemukan kesalahan yang tidak terdeteksi oleh White Box Testing, metode alternatif yang dapat digunakan adalah Black Box Testing. Black Box Testing berfungsi untuk mengidentifikasi beberapa jenis permasalahan, seperti ketidaksesuaian atau hilangnya fungsi tertentu, kesalahan dalam antarmuka, kekeliruan dalam struktur data maupun basis data, ketidaktepatan dalam fungsi yang dijalankan, kesalahan dalam deklarasi dan penghentian proses [NUR19].



2.2. Penelitian Terdahulu

Penelitian terdahulu menjadi acuan dalam pengembangan sistem yang dibahas. Rangkuman beberapa penelitian yang relevan disajikan pada Tabel 2.1 Penelitian Terdahulu berikut ini :

Tabel 2.1 Penelitian Terdahulu

No	Peneliti	Hasil Penelitian	Alasan digunakan	Komponen yang digunakan
1.	Brad DayLey, Brendan DayLey, Caleb DayLey	Learning <i>Angular</i> : A Hands-On Guide to <i>Angular 2</i> and <i>Angular 4</i>	Menjelaskan bagaimana membangun aplikasi web modern dengan pendekatan SPA	Data binding, directives, modules, <i>Dependency Injection</i>
2.	Wahyu Kartika Candra Kirana, Zainudin Zuhri	Pemanfaatan Directive pada Framework <i>Angular</i> untuk Pengembangan Website Penerimaan Mahasiswa Baru	Menjelaskan mengenai memudahkan pengembangan aplikasi web dinamis dengan komponen dan directive	Directives, data binding
3.	Lius Alviando, Adhitya Bhawiyuga, Dany Primanita Kartikasari	Penerapan Websocket pada Sistem Live Chat berbasis Web (Studi Kasus Website Kwikku.com)	Menjelaskan implementasi komunikasi <i>Real-time</i> menggunakan WebSocket dalam sistem live chat yang responsif dan efisien	WebSocket
4.	I Gede Bagus Febryntara	Rancang Bangun Sistem Layanan Live Chat Berbasis Website pada Politeknik Negeri Bali Menggunakan Framework Laravel	Menjelaskan penerapan Laravel untuk membangun fitur live chat yang terstruktur dan terintegrasi dalam sistem layanan informasi perguruan tinggi.	Laravel (routing, controller, model), Blade template, AJAX
5.	Edwar Permana, Ade Sutedi, Rickard Elsen	Pengembangan Aplikasi Live Chat Menggunakan Extreme Programing	Menjelaskan proses pengembangan live chat menggunakan metode Extreme Programming untuk meningkatkan efisiensi tim dan kualitas sistem.	Metodologi Extreme Programming, iterative development, user stories, <i>Real-time</i> messaging
6.	R. Mochammad Armand Giri Seno, Arief Jananto	Pengembangan Fitur Chat Pada Website PT. Danatrans Service Logistics	Menjelaskan penerapan fitur chat dalam sistem logistik untuk meningkatkan komunikasi antara <i>Customer</i> dan admin secara langsung.	Web-based chat module, database untuk menyimpan pesan, <i>Real-time</i> interface


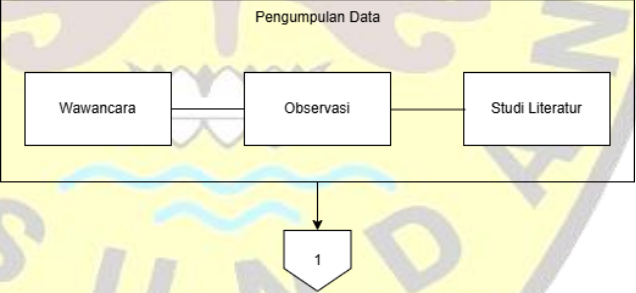
Berdasarkan Tabel 2.1 Penelitian Terdahulu, dapat disimpulkan bahwa masing-masing penelitian memiliki kontribusi terhadap pengembangan sistem, baik dari segi pemanfaatan teknologi, pendekatan metodologi, maupun perancangan fitur. Penelitian-penelitian tersebut menjadi landasan teoritis dan teknis dalam pengembangan sistem live chat pada tugas akhir ini, dengan tujuan menghasilkan sistem yang fungsional, tepat guna, dan sesuai dengan kebutuhan pengguna.

BAB 3
SKEMA PENELITIAN

3.1. Alur Penyelesaian Tugas Akhir

Alur penyelesaian tugas akhir disusun secara sistematis dan bertahap guna memastikan proses penelitian berlangsung secara terarah. Setiap tahapan memiliki tujuan, hasil, serta kontribusi yang berperan dalam mendukung pengembangan sistem secara menyeluruh. Rangkaian tahapan tersebut disajikan pada Tabel 3.1 Alur Penyelesaian Tugas Akhir berikut ini :

Tabel 3.1 Alur Penyelesaian Tugas Akhir

Tahap dan Hasil	Langkah Penelitian	Literatur dan Referensi
<p>Tahap 1 : Menetapkan ide dan melakukan identifikasi masalah</p> <p>Hasil : Rumusan masalah, solusi dan fokus penelitian</p> <p>Kontribusi : Berguna untuk mengidentifikasi masalah yang terjadi serta menentukan solusi</p>		<p>1. Hasil observasi di PT.Ursabyte</p>
<p>Tahap 2 : Tahapan analisis kebutuhan perancangan live chat dilakukan untuk mengidentifikasi dan menganalisis fakta dan proses. Pada tahapan ini dilakukan wawancara dan studi literatur</p> <p>Hasil : Memahami topik penelitian.</p> <p>Kontribusi : Membantu untuk mengidentifikasi kebutuhan perancangan pada perancangan live chat</p>		<p>1. Definisi Perancangan</p>

<p>Tahap 3 : Tahapan pengembangan aplikasi fitur live chat berdasarkan analisis kebutuhan yang sudah dilakukan</p> <p>Hasil : Pengembangan aplikasi fitur live chat</p> <p>Kontribusi : Berguna untuk mempermudah komunikasi <i>Real-time</i> antara <i>Customer</i> dan admin dan meningkatkan responsivitas layanan</p>	<pre> graph TD 1{{1}} --> Requirement[Requirement] Requirement --> Design[Design] Design --> Implementation[Implementation] Implementation --> Verification[Verification] </pre>	<p>1. Hasil wawancara dan observasi</p>
<p>Tahap 4 : Kesimpulan dan saran</p> <p>Hasil : Menyusun kesimpulan dari seluruh tahapan dan memberikan saran untuk pengembangan ke depan.</p> <p>Kontribusi : Berguna untuk menjadi refleksi akhir untuk evaluasi dan pengembangan sistem selanjutnya.</p>	<p>Kesimpulan, saran dan rekomendasi tugas akhir</p>	

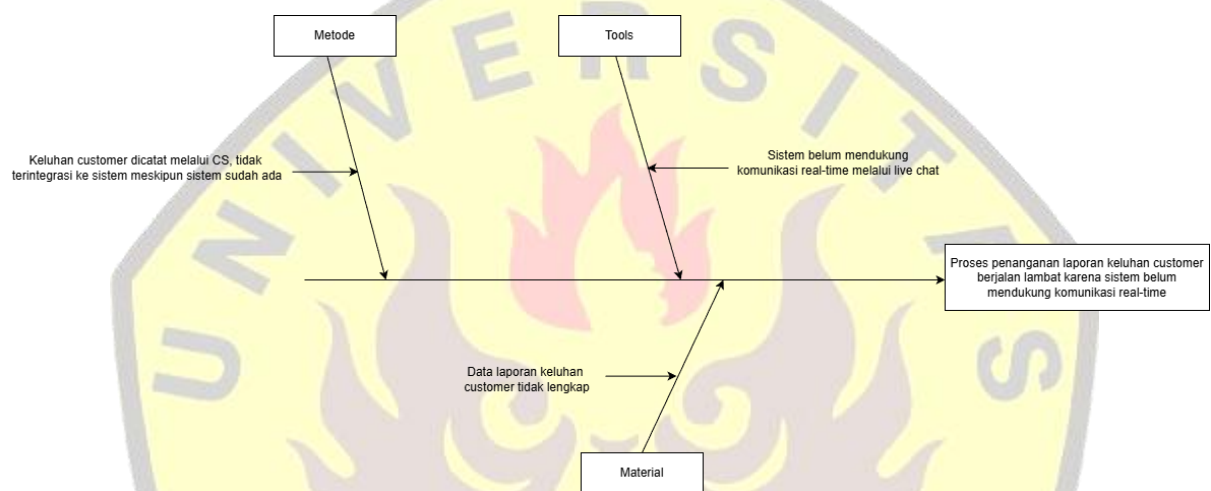
Berdasarkan Tabel 3.1 Alur Penyelesaian Tugas Akhir, dapat diketahui bahwa setiap tahapan memiliki peran strategis dalam mendukung keseluruhan proses penyusunan tugas akhir. Tahapan tersebut dimulai dari kegiatan identifikasi masalah, analisis kebutuhan, pengembangan sistem, hingga penyusunan kesimpulan dan saran sebagai dasar untuk pengembangan lebih lanjut.

3.2. Rumusan masalah

Pada bagian Rumusan Masalah ini menjelaskan penyebab utama permasalahan yang terdapat pada website pengiriman barang, dampak yang ditimbulkan, serta solusi yang diusulkan dalam tugas akhir ini guna mengatasi permasalahan tersebut secara menyeluruh.

3.2.1 Analisis sebab akibat

Analisis ini digunakan untuk menggambarkan keterkaitan antara permasalahan utama dengan faktor-faktor yang menjadi penyebabnya. Tujuan dari analisis ini adalah memperoleh pemahaman yang lebih mendalam terhadap akar permasalahan yang terjadi. Hasil analisis tersebut disajikan pada Gambar 3.1 *Fishbone* Penyebab Masalah berikut ini :



Gambar 3.1 *Fishbone* Penyebab Masalah

Berdasarkan Gambar 3.1 Diagram *Fishbone*, dijelaskan bahwa terdapat beberapa faktor penyebab lambatnya proses penanganan laporan pelanggan. Faktor-faktor tersebut dikategorikan ke dalam beberapa aspek, yaitu sebagai berikut:

- a. Pada aspek tools, sistem yang digunakan saat ini belum mendukung komunikasi secara *Real-time* melalui fitur live chat. Akibatnya, *Customer* tidak dapat menyampaikan keluhan secara langsung ke sistem, dan masih bergantung pada perantara (*Customer service*), sehingga alur pelaporan menjadi tidak efisien.
- b. Pada aspek metode, meskipun sistem pencatatan keluhan *Customer* telah tersedia, proses pelaporan awal masih dilakukan secara manual melalui *Customer service*, tanpa dukungan form input langsung dari *Customer*.
- c. Pada aspek material, data laporan keluhan *Customer* sering kali tidak lengkap atau tidak terdokumentasi dengan baik karena dicatat secara manual. Hal ini menimbulkan risiko

informasi yang hilang atau tidak akurat, sehingga berdampak pada efektivitas dalam menyelesaikan masalah *Customer*.

3.2.2 Solusi Masalah

Berdasarkan hasil analisis sebab-akibat yang telah dilakukan, berikut disajikan solusi yang dirancang untuk mendukung penyelesaian tugas akhir ini.

Tabel 3.2 Tabel solusi Masalah

No	Masalah	Solusi
1.	Sistem belum memiliki fitur komunikasi langsung (<i>Real-time</i>) antara <i>Customer</i> dan <i>Customer service</i> .	Mengintegrasikan fitur live chat berbasis <i>Real-time</i> ke dalam sistem untuk mendukung komunikasi langsung.
2.	Proses pencatatan laporan keluhan <i>Customer</i> masih manual (via <i>Customer service</i>)	Menyediakan form keluhan secara otomatis di awal sesi live chat agar <i>Customer</i> dapat mengisi laporan secara langsung.
3.	Data laporan keluhan <i>Customer</i> yang diterima sering kali tidak lengkap atau tidak tersimpan dengan baik.	Mengelola data laporan keluhan secara <i>Real-time</i> dan terstruktur di dalam sistem agar seluruh informasi tercatat dengan rapi dan terdokumentasi.

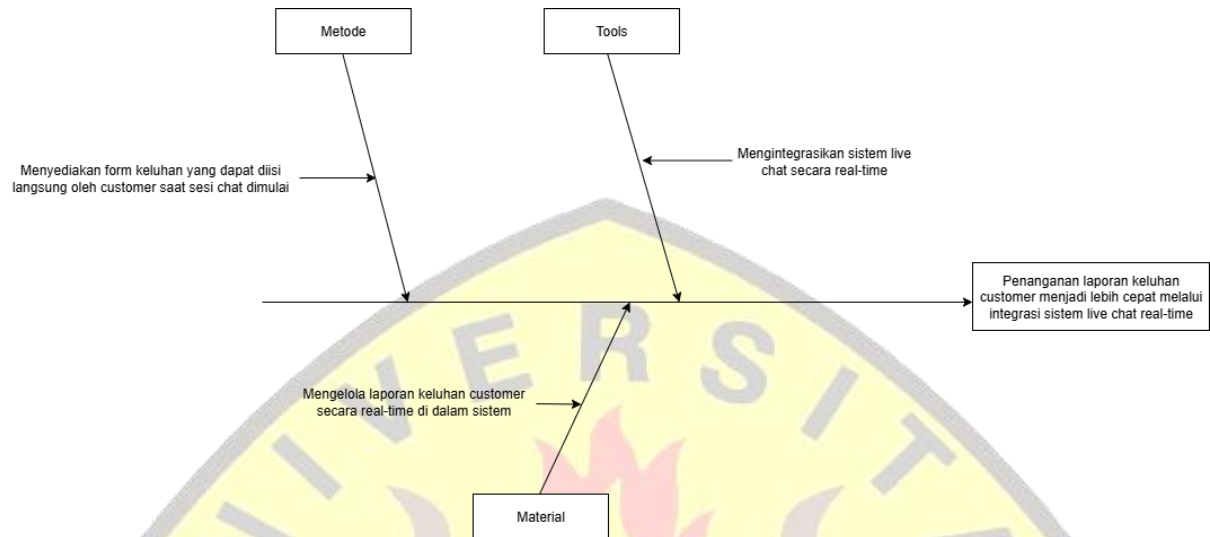
Berdasarkan Tabel 3.2, dapat disimpulkan bahwa solusi yang dirancang bertujuan untuk meningkatkan efisiensi proses penanganan laporan pelanggan melalui pemanfaatan sistem yang memungkinkan komunikasi langsung serta pencatatan data secara terstruktur.

3.3. Kerangka pemikiran Teoritis

Penelitian ini berawal dari permasalahan dalam proses pengiriman barang, di mana sistem yang digunakan saat ini memiliki beberapa keterbatasan. Belum tersedianya fitur live chat membuat komunikasi antara *Customer* dan *Customer service* menjadi terbatas, sehingga memperlambat penyampaian informasi atau penanganan kendala. Oleh karena itu, diperlukan pengembangan sistem yang menyediakan fitur live chat untuk memperlancar proses komunikasi dan penyampaian informasi dalam pengiriman barang.

3.3.1 Gambaran Produk TA

Gambaran produk tugas akhir ini menjelaskan solusi yang dirancang untuk menjawab permasalahan yang telah diidentifikasi sebelumnya. Rangkaian solusi tersebut dirumuskan berdasarkan hasil analisis dan disajikan pada Gambar 3.2 *Fishbone* Solusi berikut.



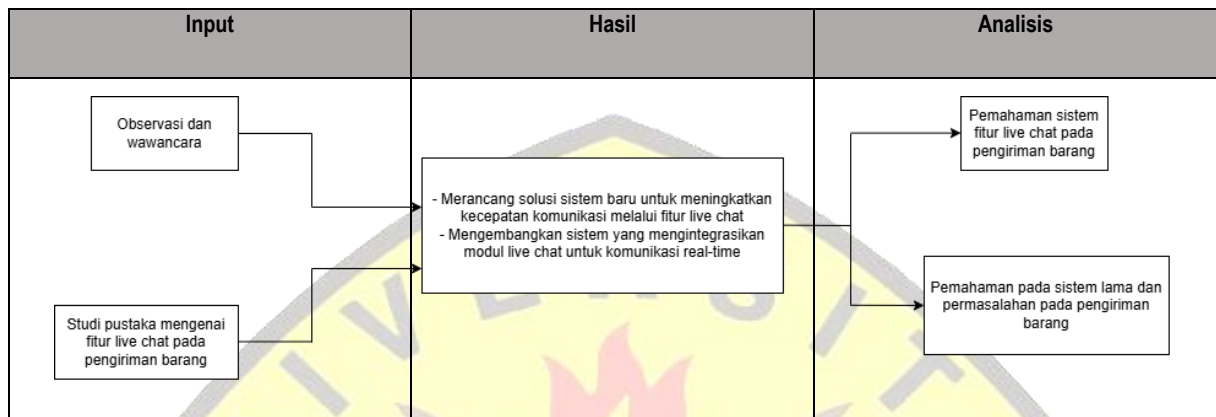
Berdasarkan Gambar 3.2 Diagram *Fishbone* Solusi, dapat diketahui bahwa setiap aspek permasalahan memiliki solusi yang dapat diimplementasikan melalui pengembangan aplikasi. Solusi-solusi tersebut dirancang untuk mengatasi berbagai hambatan yang telah diidentifikasi sebelumnya. Adapun penerapan solusi pada masing-masing aspek dijelaskan sebagai berikut.

1. Pada aspek tools, solusi yang diterapkan adalah mengintegrasikan sistem dengan modul live chat berbasis *Real-time*. Hal ini memungkinkan *Customer* berkomunikasi langsung dengan *Customer service* tanpa perantara, serta memungkinkan sistem mencatat laporan keluhan secara otomatis selama sesi berlangsung.
2. Pada aspek metode, solusi yang diterapkan adalah menyediakan form keluhan yang dapat diisi langsung oleh pengguna saat sesi chat dimulai, agar laporan keluhan *Customer* masuk secara terstruktur sejak awal interaksi dan langsung diproses menjadi tiket tanpa pencatatan ulang.
3. Pada aspek material, Solusi yang diterapkan adalah mengelola laporan keluhan *Customer* secara *Real-time* di dalam sistem, sehingga data yang diterima lebih lengkap, terdokumentasi dengan baik, serta mempermudah proses pelacakan dan penanganan masalah

3.3.2 Skema Analisis Teori

Pada bagian ini menjelaskan analisis yang dilakukan selama pelaksanaan penelitian tugas akhir, yang mencakup identifikasi input yang diperlukan untuk proses analisis serta output yang dihasilkan. Rincian mengenai skema analisis teori yang digunakan dalam penelitian ini disajikan pada Tabel 3.3 Skema Analisis Teori berikut ini :

Tabel 3.3 Skema Analisis Teori



Berdasarkan Tabel 3.3, skema analisis teori menunjukkan bahwa data dari observasi, wawancara, dan studi pustaka dianalisis untuk merancang solusi sistem yang mendukung komunikasi *Real-time* melalui fitur live chat.

Deskripsi proses analisis yang dilakukan dalam penelitian ini mencakup komponen input, langkah-langkah analisis, objek kajian, output yang dihasilkan, serta penjelasan pada setiap tahap secara sistematis. Rincian tersebut disajikan pada Tabel 3.4 Penjelasan Skema Teori berikut ini :

Tabel 3.4 Penjelasan Skema Teori

No	Input	Langkah Analisis	Objek Analisis	Output	Penjelasan
1.	<ul style="list-style-type: none"> - Observasi dan wawancara - Studi pustaka mengenai fitur live chat pada pengiriman barang 	<ul style="list-style-type: none"> - Merancang solusi sistem baru untuk meningkatkan kecepatan komunikasi melalui fitur live chat - Mengembangkan sistem yang mengintegrasikan modul live chat untuk komunikasi <i>Real-time</i> 	PT.UrsaByte	<ul style="list-style-type: none"> - Pemahaman sistem fitur live chat pada pengiriman barang - pemahaman pada sistem lama dan permasalahan pada pengiriman barang 	Tahap ini menjelaskan mengenai observasi, wawancara dan studi Pustaka yang digunakan pada penelitian tugas akhir ini

Berdasarkan Tabel 3.4, diperoleh hasil analisis dari observasi, wawancara, dan studi pustaka yang digunakan untuk memahami sistem lama serta merancang solusi dengan integrasi fitur live chat *Real-time*.

3.4 Profil Penelitian

Pada bagian profil Penelitian akan dijelaskan mengenai sejarah perusahaan, visi dan misi serta struktur organisasi PT. Ursabyte

3.4.1 Sejarah Perusahaan

PT Ursabyte adalah perusahaan teknologi Informasi yang didirikan pada Februari 2013 dengan tujuan untuk memberikan solusi inovatif dalam bidang pengembangan perangkat lunak. Sejak didirikan, PT Ursabyte merupakan perusahaan teknologi Informasi yang berfokus pada pengembangan perangkat lunak seperti web application, mobile application dan desktop application.

3.4.2 Visi dan Misi PT. Ursabyte

Untuk mencapai tujuan sebagai penyedia solusi inovatif dalam pengembangan perangkat lunak, PT. Ursabyte memiliki visi dan misi yang menjadi pedoman dalam menjalankan setiap kegiatannya. Berikut adalah visi dan misi PT. Ursabyte:

Visi:

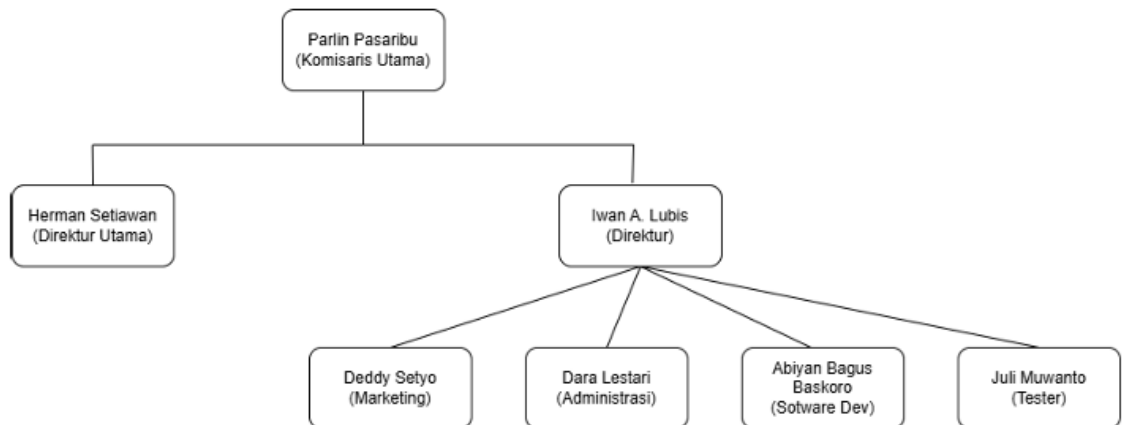
Menjadi perusahaan terkemuka dalam pengembangan perangkat lunak yang menyediakan solusi inovatif dan terpercaya untuk mendukung proses bisnis klien.

Misi:

1. Mengembangkan perangkat lunak berkualitas yang sesuai dengan kebutuhan spesifik klien.
2. Memberikan konsultasi manajemen yang efektif untuk meningkatkan produktivitas dan kelancaran bisnis klien.
3. Menerapkan teknologi terbaru serta metodologi yang terbukti untuk memastikan solusi yang praktis dan dapat diandalkan.

3.4.2 Struktur Organisasi PT. Ursabyte

Struktur organisasi di PT. Ursabyte menggambarkan pembagian tugas dan tanggung jawab antar bagian dalam menjalankan kegiatan operasional perusahaan. Susunan struktur tersebut dapat dilihat pada Gambar 3.4 Struktur Organisasi berikut ini :



Gambar 3.4 Struktur Organisasi

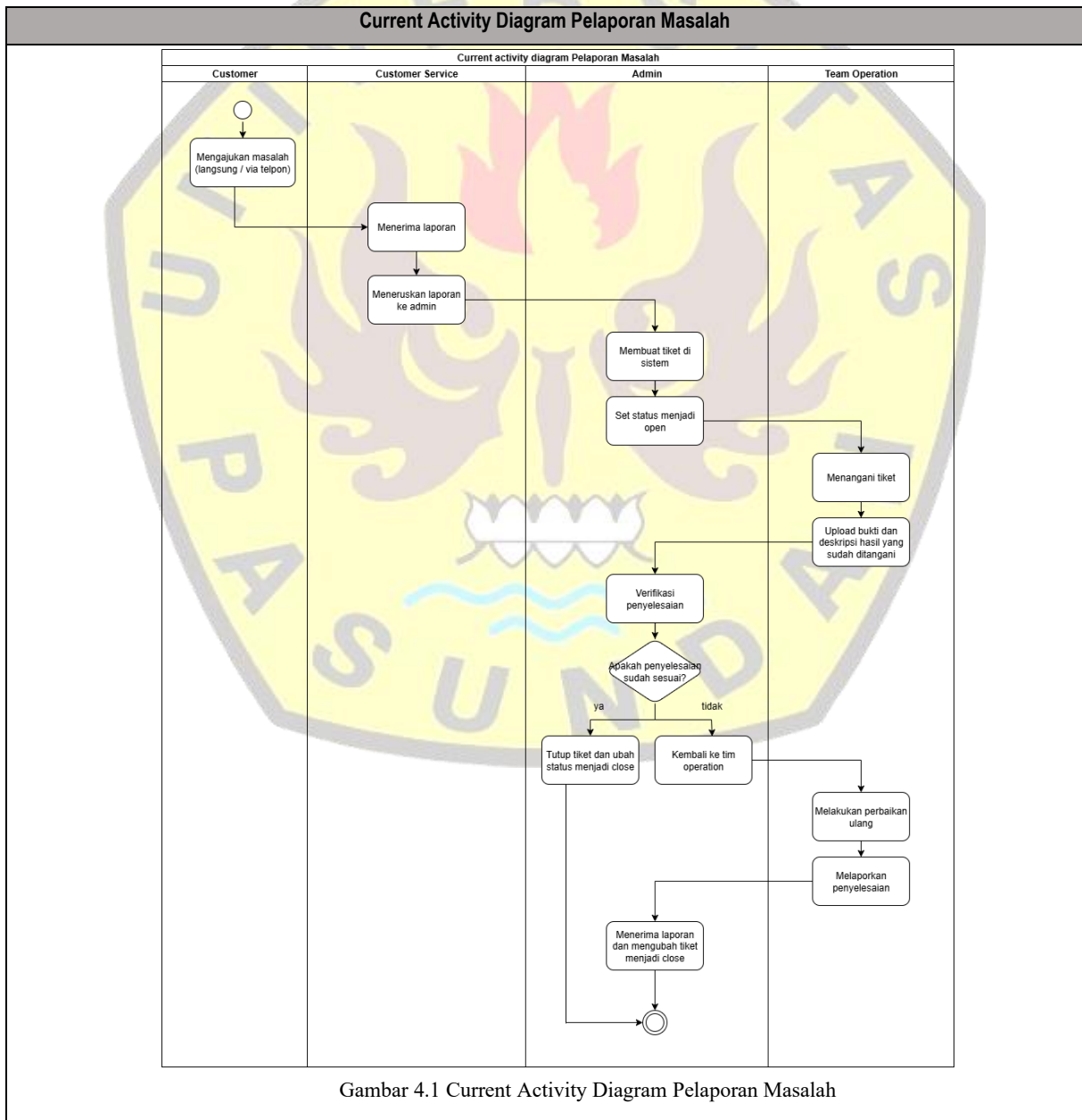
Berdasarkan Gambar 3.4, struktur organisasi ini menjadi acuan dalam pembagian fungsi kerja dan koordinasi antar divisi di lingkungan perusahaan.

BAB 4 ANALISIS DAN PERANCANGAN

4.1. Sistem Saat Ini (Current System)

Pada saat ini, proses pengiriman barang belum dilengkapi dengan fitur live chat sehingga komunikasi antara *Customer service* dan *Customer* kurang efisien. Ketiadaan sarana komunikasi secara *Real-time* menyebabkan penanganan pertanyaan atau kendala *Customer* menjadi terlambat. Oleh karena itu, diperlukan pengembangan fitur live chat yang dapat memungkinkan interaksi langsung dan responsif antara *Customer* dan *Customer service* dalam sistem pengiriman barang. Proses yang berlangsung dalam sistem saat ini ditunjukkan pada Tabel 4.1 Current Activity Diagram Pelaporan Masalah berikut ini :

Tabel 4.1 Current Activity Diagram Pelaporan Masalah



Gambar 4.1 Current Activity Diagram Pelaporan Masalah

Deskripsi
Diagram aktivitas ini menjelaskan alur pelaporan masalah oleh <i>Customer</i> hingga tiket ditutup. <i>Customer</i> melapor ke <i>Customer service</i> , lalu diteruskan ke Admin untuk dibuatkan tiket. Setelah tiket dibuat, Team Operation menangani permasalahan dan mengunggah hasilnya. Tiket kemudian ditutup oleh Admin atau sistem setelah laporan dianggap lengkap.

4.2. Analisis Kebutuhan

Pada tahap analisis kebutuhan dalam tugas akhir berjudul Pengembangan Aplikasi Fitur Live Chat untuk Pengiriman Barang Menggunakan Framework *Angular* (Studi Kasus: PT. UrsaByte), dilakukan kajian terhadap proses bisnis pengiriman barang yang sedang berjalan. Kegiatan ini bertujuan untuk mengidentifikasi fitur-fitur penting yang diperlukan dalam aplikasi, seperti komunikasi *Real-time*, pengelolaan percakapan berbasis sistem ticketing, serta autentikasi pengguna. Metode yang digunakan dalam analisis meliputi studi literatur, wawancara dan observasi lapangan, sehingga kebutuhan sistem baru dapat dirancang dengan tepat dan sesuai dengan kondisi operasional yang ada.

4.2.1 Menentukan Kebutuhan Perangkat Lunak

Pada tahap ini, dilakukan identifikasi kebutuhan atau kondisi yang harus dipenuhi oleh aplikasi yang akan dikembangkan. Semua fitur wajib dan kemampuan yang harus dimiliki oleh aplikasi ditentukan secara jelas.

4.2.2 Analisis Hasil dan Pengumpulan Data

Analisis dilakukan dengan menerapkan metode wawancara dan observasi sebagai upaya untuk mengumpulkan informasi yang menjadi dasar penyusunan penelitian. Wawancara dilakukan secara tertulis melalui media chat untuk mempermudah komunikasi sekaligus memperoleh data secara lebih efektif, sehingga informasi yang diperoleh dapat dianalisis secara mendalam. Selain itu, observasi dilakukan secara langsung untuk melengkapi data yang dikumpulkan melalui wawancara.

4.2.3 Analisis Stakeholder

Analisis stakeholder merupakan proses identifikasi individu atau kelompok yang memiliki pengaruh maupun peran, baik secara langsung maupun tidak langsung, dalam pengembangan sistem. Daftar stakeholder disajikan pada Tabel 4.2 Analisis Stakeholder berikut.

Tabel 4.2 Analisis Stakeholder

No	Stakeholder	Deskripsi
1.	<i>Customer</i>	Pengguna yang memanfaatkan fitur live chat sebagai sarana komunikasi dalam rangka memperoleh informasi atau layanan.
2.	<i>Customer service</i>	Tim layanan <i>Customer</i> yang bertugas menggunakan fitur live chat untuk memberikan bantuan serta merespons pertanyaan dari pengguna secara efektif dan efisien.

No	Stakeholder	Deskripsi
3.	Admin	Pihak yang bertugas mengelola sistem atau platform secara keseluruhan
4.	Team Operation	Pihak yang bertanggung jawab menangani permasalahan di lapangan berdasarkan informasi yang disampaikan oleh <i>Customer service</i> . Team Operation tidak berinteraksi langsung dengan sistem, namun berperan dalam proses penyelesaian masalah sebagai bagian dari alur operasional layanan.

Berdasarkan Tabel 4.2, stakeholder terdiri dari empat pihak utama yang terlibat dalam sistem. Masing-masing memiliki peran berbeda, mulai dari *Customer*, *Customer service*, admin, hingga team operation. Identifikasi ini membantu dalam menyesuaikan fitur sesuai kebutuhan masing-masing pihak.

4.2.4 Analisis Fungsional

Analisis ini merupakan tahap untuk memenuhi kebutuhan fungsional perangkat lunak yang akan dikembangkan, dengan menerjemahkan kebutuhan pengguna menjadi kebutuhan fungsional pada aplikasi. Rincian analisis fungsional disajikan pada Tabel 4.3 Analisis Fungsional berikut.

Tabel 4.3 Analisis Fungsional

No	Kode FR	Kebutuhan Fungsional	Deskripsi
1.	FR-01	Perangkat lunak harus dapat mengelola akun admin secara individual	Sistem harus memungkinkan setiap admin memiliki akun login sendiri dengan hak akses yang dapat diatur.
2.	FR-02	Perangkat lunak harus dapat mengelola tiket	Sistem memungkinkan admin membuat tiket dari laporan <i>Customer</i> , dan memungkinkan <i>Customer service</i> mengambil tiket sesuai prioritas dan ketersediaan.
3.	FR-03	Perangkat lunak harus dapat memfasilitasi sesi live chat	Sistem memungkinkan pengguna memulai komunikasi melalui live chat untuk menyampaikan laporan atau pertanyaan secara langsung kepada <i>Customer service</i> .
4.	FR-04	Perangkat lunak harus dapat menampilkan daftar sesi chat	Sistem harus memungkinkan admin untuk melihat daftar sesi chat yang masuk dari pengguna.
5.	FR-05	Perangkat lunak harus dapat menampilkan status online/offline	Sistem menampilkan status online atau offline pengguna dan <i>Customer service</i> agar komunikasi berjalan transparan.

No	Kode FR	Kebutuhan Fungsional	Deskripsi
6.	FR-06	Perangkat lunak harus dapat mengirim pesan secara <i>Real-time</i>	Sistem memungkinkan pesan dikirim secara langsung tanpa jeda antara pengguna dan <i>Customer service</i> .
7.	FR-07	Perangkat lunak harus dapat menerima pesan secara <i>Real-time</i>	Sistem mampu menampilkan pesan yang diterima secara langsung agar respon cepat terlaksana.
8.	FR-08	Perangkat lunak harus dapat mencatat identitas petugas pengirim pesan	Sistem mencatat dan menampilkan identitas petugas (admin atau <i>Customer service</i>) yang mengirim pesan agar jelas siapa yang menangani percakapan.
9.	FR-09	Perangkat lunak harus dapat menyimpan riwayat pesan chat	Sistem menyimpan seluruh percakapan ke dalam basis data sebagai catatan dan referensi laporan selanjutnya.

Berdasarkan Tabel 4.3, kebutuhan fungsional yang telah dianalisis akan menjadi dasar pengembangan fitur dalam perangkat lunak live chat. Setiap kebutuhan fungsional menggambarkan fungsi utama yang harus disediakan oleh sistem, mulai dari pengelolaan akun, tiket, hingga komunikasi *Real-time*. Kebutuhan tersebut akan diterjemahkan ke dalam fitur-fitur yang mendukung kelancaran interaksi antara pengguna, *Customer service*, dan admin dalam satu platform terintegrasi.

Sebagai landasan dalam perancangan perangkat lunak, daftar fitur kebutuhan fungsional yang mencakup fungsi-fungsi utama untuk memenuhi kebutuhan pengguna dan mendukung kelancaran operasional sistem secara menyeluruh disajikan pada Tabel 4.4 Fitur Kebutuhan Fungsional berikut.

Tabel 4.4 Fitur Kebutuhan Fungsional

No	Fitur	Kode Kebutuhan Fungsional
1.	Pengelolaan akun admin	FR-01
2.	Pengelolaan tiket <i>Customer</i>	FR-02
3.	Sesi live chat <i>Customer</i> dan <i>Customer service</i>	FR-03
4.	Daftar sesi live chat	FR-04
5.	Status online atau offline <i>Customer</i> dan <i>Customer service</i>	FR-05
6.	Pengiriman pesan real-time	FR-06
7.	Penerimaan pesan real-time	FR-07
8.	Pencatatan identitas pengirim pesan	FR-08
9.	Penyimpanan dan riwayat chat	FR-09

Berdasarkan Tabel 4.4 Fitur Kebutuhan Fungsional, dapat diketahui bahwa setiap kebutuhan fungsional telah diterjemahkan ke dalam fitur yang spesifik pada perangkat lunak. Fitur-fitur tersebut dirancang untuk mendukung proses komunikasi dan pengelolaan sistem secara menyeluruh. Dengan pemetaan ini, sistem dapat dikembangkan secara terstruktur sesuai dengan kebutuhan pengguna dan tujuan utama aplikasi.

4.2.5 Analisis Pengguna

Tahapan ini merupakan proses analisis yang bertujuan untuk mengidentifikasi siapa saja pengguna yang akan menggunakan dan berinteraksi dengan aplikasi yang dikembangkan. Daftar pengguna disajikan pada Tabel 4.5 Analisis Pengguna berikut.

Tabel 4.5 Analisis Pengguna

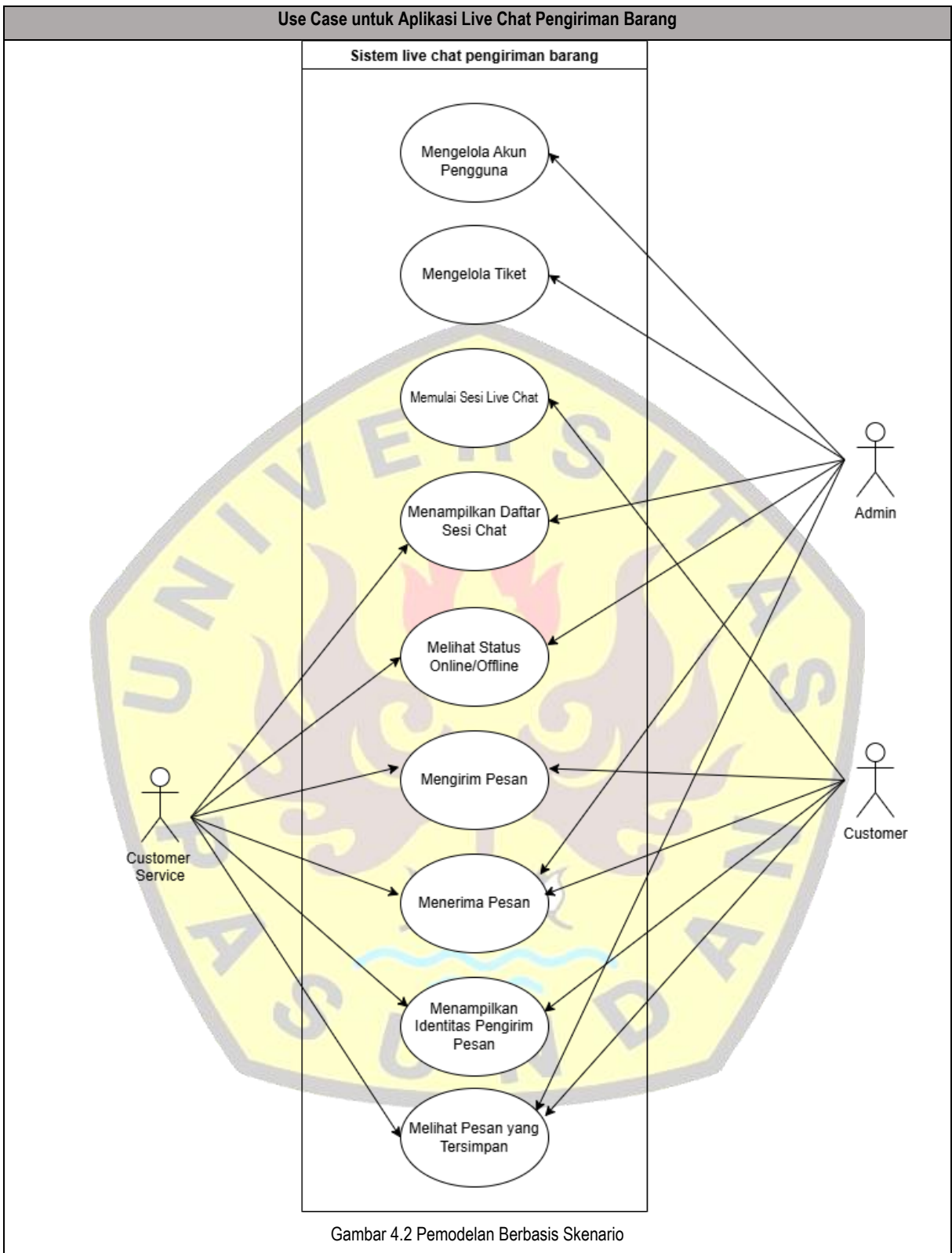
No	Pengguna	Deskripsi
1.	<i>Customer</i>	Orang atau pihak yang menggunakan fitur live chat untuk mendapatkan informasi, bantuan, atau layanan.
2.	<i>Customer service</i>	Orang yang bertugas merespons, membantu, dan menyelesaikan pertanyaan atau masalah yang disampaikan oleh <i>Customer</i> melalui fitur live chat.
3.	Admin	Orang yang bertanggung jawab mengelola dan memelihara sistem aplikasi secara keseluruhan, termasuk pengaturan akses pengguna dan pengawasan fitur live chat.

Berdasarkan Tabel 4.5 Analisis Pengguna, pengguna sistem dibagi menjadi tiga peran utama, yaitu *Customer*, *Customer service*, dan admin. Masing-masing memiliki fungsi dan tanggung jawab yang berbeda dalam interaksi dengan sistem.

4.2.6 Pemodelan berbasis skenario

Pemodelan berbasis skenario digunakan untuk menggambarkan interaksi antara *Customer* dan sistem atau perangkat lunak. Pemodelan ini memanfaatkan UML (Unified Modeling Language) sebagai spesifikasi kebutuhan pengguna, yang umumnya diawali dengan pembuatan skenario dalam bentuk use case, diagram aktivitas, dan diagram swimlane. Berikut adalah use case yang dikembangkan untuk aplikasi live chat berbasis web menggunakan framework *Angular*. Use case diagram dapat dilihat pada Gambar 4.2 Pemodelan Berbasis Skenario berikut.

Tabel 4.6 Pemodelan Berbasis Skenario



Deskripsi

Diagram use case live chat pada sistem pengiriman barang digunakan untuk menggambarkan interaksi antara pengguna dan sistem dalam proses komunikasi.

4.2.7 Deskripsi Use Case

Diagram use case digunakan untuk memvisualisasikan interaksi antara pengguna dan sistem atau perangkat lunak yang dikembangkan. Penjelasan lebih rinci mengenai masing-masing use case pada diagram disajikan dalam Tabel 4.7 Deskripsi Use Case berikut.

Tabel 4.7 Deskripsi *Use Case*

No	Kode	Use Case	Deskripsi
1.	UC-01	Manajemen Akun Pengguna	Use case ini memungkinkan Admin mengelola akun pengguna, termasuk menambah, mengubah, menghapus, dan mengatur hak akses masing-masing.
2.	UC-02	Mengelola Tiket	Use case ini memungkinkan Admin membuat dan memperbarui tiket, menentukan status serta prioritas, dan menugaskan tiket kepada <i>Customer service</i> yang bertanggung jawab menanganinya.
3.	UC-03	Memulai Sesi Live Chat	Use case ini memungkinkan <i>Customer</i> memulai sesi chat dengan <i>Customer service</i> untuk menyampaikan pertanyaan atau keluhan.
4.	UC-04	Menampilkan Daftar Sesi Chat	Use case ini memungkinkan <i>Customer service</i> dan Admin melihat daftar sesi chat yang sedang berlangsung maupun yang telah selesai.
5.	UC-05	Melihat Status Online atau Offline Pengguna	Use case ini memungkinkan <i>Customer service</i> dan Admin melihat status aktif atau tidak aktif dari pengguna lain di sistem.
6.	UC-06	Mengirim Pesan	Use case ini memungkinkan <i>Customer service</i> dan <i>Customer</i> saling mengirim pesan selama sesi chat berlangsung.
7.	UC-07	Menerima Pesan	Use case ini memungkinkan Admin, <i>Customer service</i> , dan <i>Customer</i> menerima pesan sesuai hak aksesnya dan Admin hanya memantau.
8.	UC-08	Menampilkan Identitas Pengirim Pesan	Use case ini menampilkan identitas pengirim pada setiap pesan agar penerima mengetahui asal pesan.
9.	UC-09	Melihat Pesan Tersimpan (Riwayat Chat)	Use case ini memungkinkan Admin, <i>Customer service</i> , dan <i>Customer</i> melihat riwayat percakapan yang tersimpan di sistem.

Berdasarkan Tabel 4.7 Deskripsi *Use Case*, setiap use case menggambarkan fungsi utama yang dilakukan oleh pengguna dalam sistem. Masing-masing use case berperan penting dalam mendukung proses komunikasi antara *Customer*, *Customer service*, dan admin. Dengan analisis ini, pengembangan sistem dapat disesuaikan agar mencakup seluruh kebutuhan interaksi yang diperlukan pengguna.

4.2.8 Skenario Use Case

Skenario use case digunakan untuk memaparkan alur proses yang terjadi antara sistem dan aktor berdasarkan use case yang telah dirancang sebelumnya. Skenario use case pada aplikasi live chat ini disajikan secara rinci pada tabel berikut.

Table 4.8 Skenario Use Case Manajemen Akun Pengguna

Identifikasi	
Kode - Nama	UC-01 - Manajemen Akun Pengguna
Tujuan	Memungkinkan Admin untuk mengelola akun pengguna yang secara otomatis terdaftar dari data masuk, termasuk memperbarui informasi, mengatur hak akses, status aktif/nonaktif, dan menghapus akun bila diperlukan.
Deskripsi	Use case ini menggambarkan proses pengelolaan akun pengguna oleh Admin. Admin dapat mengubah data akun, mengatur hak akses, mengaktifkan atau menonaktifkan akun, serta menghapus akun yang tidak diperlukan.
Aktor	Admin
Kondisi awal	Admin telah login ke sistem.
Skenario Utama	
Aksi aktor	Reaksi sistem
1. Admin memilih menu "Users"	2. Sistem menampilkan daftar akun yang telah terdaftar
3. Admin memilih akun tertentu untuk diperbarui	4. Sistem menampilkan form edit akun
5. Admin memilih akun tertentu untuk dihapus	6. Sistem menampilkan konfirmasi dan menghapus akun dari daftar setelah disetujui
Kondisi akhir	Akun pengguna berhasil diperbarui, dinonaktifkan, atau dihapus sesuai kebutuhan Admin.

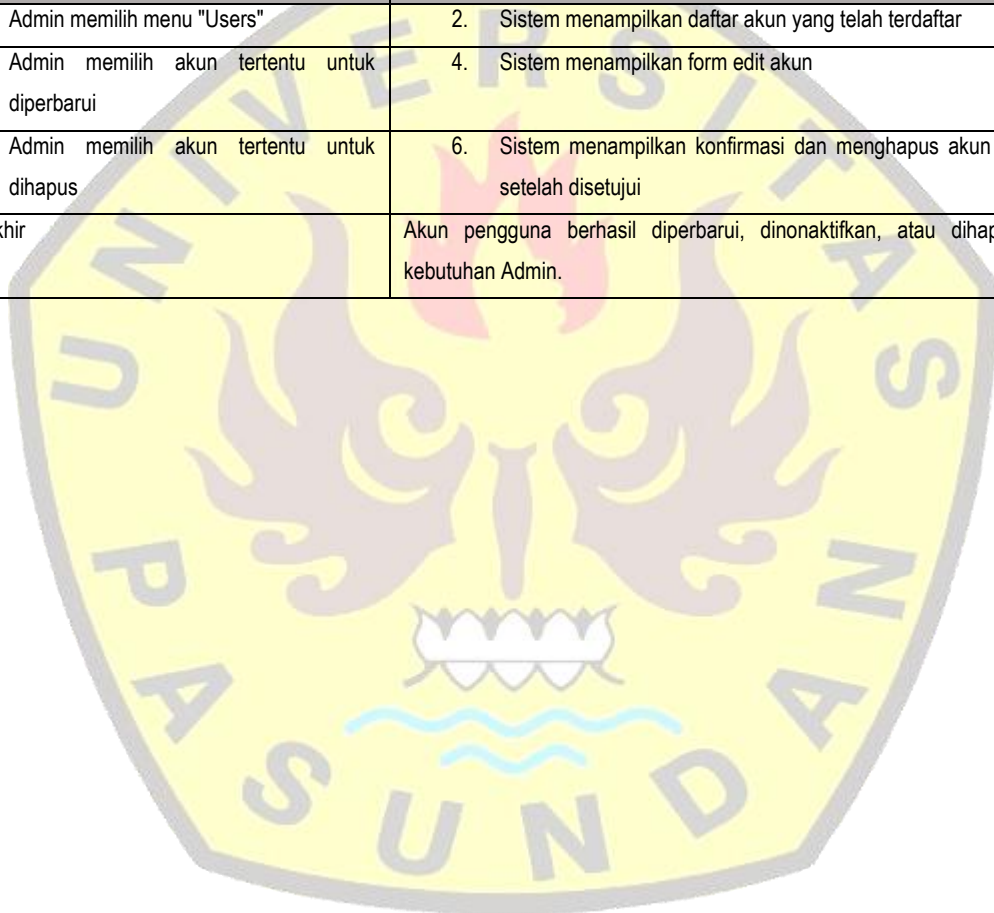


Table 4.9 Skenario Use Case Mengelola Tiket

Identifikasi	
Kode - Nama	UC-02 – Mengelola Tiket
Tujuan	Memungkinkan Admin untuk mengelola tiket yang telah dibuat oleh <i>Customer</i> , termasuk menentukan status, menetapkan prioritas, dan menugaskan tiket kepada <i>Customer service</i> yang akan menanganinya.
Deskripsi	Use case ini menjelaskan bagaimana Admin mengatur tiket yang telah diinput oleh <i>Customer</i> . Admin dapat memperbarui status tiket, menetapkan tingkat prioritas serta menugaskan tiket tersebut kepada <i>Customer service</i> tertentu untuk penanganan lebih lanjut.
Aktor	Admin
Kondisi awal	Admin telah login ke sistem dan data tiket sudah tersedia di sistem hasil input dari <i>Customer</i> .
Skenario Utama	
Aksi aktor	Reaksi sistem
1. Admin membuka menu "Tiket"	2. Sistem menampilkan daftar tiket yang telah diinput oleh <i>Customer</i>
3. Admin memilih salah satu tiket untuk dikelola	4. Sistem menampilkan detail tiket, termasuk informasi ID Resi dan isi laporan.
5. Admin memperbarui status dan prioritas tiket sesuai kebutuhan.	6. Sistem menyimpan pembaruan status dan prioritas tiket.
7. Admin memilih opsi "Assign to CS" dan menentukan <i>Customer service</i> yang akan menangani tiket tersebut.	8. Sistem menetapkan penugasan tiket kepada <i>Customer service</i> yang dipilih dan memperbarui data tiket.
Kondisi akhir	Tiket berhasil diperbarui dengan status dan prioritas baru serta telah ditugaskan kepada <i>Customer service</i> yang ditentukan oleh Admin.

Table 4.10 Skenario Use Case Memulai Sesi Live Chat

Identifikasi	
Kode - Nama	UC-03 - Memulai Sesi Live Chat
Tujuan	Memungkinkan <i>Customer</i> memulai sesi chat untuk berkomunikasi langsung dengan <i>Customer service</i> .
Deskripsi	Use case ini menggambarkan proses di mana <i>Customer</i> memulai sesi percakapan untuk menanyakan atau melaporkan kendala terkait pengiriman barang.
Aktor	<i>Customer</i>
Kondisi awal	<i>Customer</i> telah login ke sistem.
Skenario Utama	
Aksi aktor	Reaksi sistem
1. <i>Customer</i> mengklik icon "bantuan" pada halaman utama	2. Sistem membuka sesi live chat dan menampilkan jendela percakapan
3. <i>Customer</i> mengirim pesan pembuka	4. Sistem membuat sesi chat baru dan menghubungkan dengan <i>Customer service</i>

Kondisi akhir	Sesi live chat berhasil dibuat dan siap digunakan untuk komunikasi.
---------------	---

Table 4.11 Skenario Use Case Menampilkan Daftar Sesi Chat

Identifikasi	
Kode - Nama	UC-04 - Menampilkan Daftar Sesi Chat
Tujuan	Memungkinkan <i>Customer service</i> dan Admin melihat daftar sesi chat yang sedang berlangsung maupun telah selesai.
Deskripsi	Use case ini menampilkan daftar seluruh sesi chat yang tersimpan, lengkap dengan identitas <i>Customer</i> , waktu, dan isi percakapan.
Aktor	<i>Customer service</i> dan Admin
Kondisi awal	<i>Customer service</i> atau admin sudah login ke sistem.
Skenario Utama	
Aksi aktor	Reaksi sistem
1. Aktor memilih menu "Chat"	2. Sistem menampilkan daftar sesi chat aktif dan riwayat yang telah selesai.
Kondisi akhir	Daftar sesi chat berhasil ditampilkan untuk ditinjau.

Table 4.12 Skenario Use Case Melihat Status Online atau Offline Pengguna

Identifikasi	
Kode - Nama	UC-05 - Melihat Status Online atau Offline Pengguna
Tujuan	Memungkinkan <i>Customer service</i> dan Admin mengetahui status aktif atau tidak aktif pengguna lain.
Deskripsi	Use case ini menampilkan status online atau offline pengguna untuk memantau ketersediaan <i>Customer service</i> dan <i>Customer</i> di sistem.
Aktor	<i>Customer service</i> dan Admin
Kondisi awal	Aktor telah login ke sistem.
Skenario Utama	
Aksi aktor	Reaksi sistem
1. Aktor membuka sesi chat pengguna.	2. Sistem menampilkan status online atau offline masing-masing pengguna.
Kondisi akhir	Aktor dapat melihat status online/offline pengguna lain.

Table 4.13 Skenario Use Case Mengirim Pesan

Identifikasi	
Kode - Nama	UC-06 - Mengirim Pesan
Tujuan	Memungkinkan <i>Customer</i> dan <i>Customer service</i> mengirim pesan selama sesi chat berlangsung.
Deskripsi	Use case ini mendukung pertukaran pesan secara <i>Real-time</i> antara <i>Customer</i> dan <i>Customer service</i> untuk berkomunikasi dalam sesi chat aktif.
Aktor	<i>Customer service</i> dan <i>Customer</i>
Kondisi awal	Sesi chat telah dimulai.
Skenario Utama	
Aksi aktor	Reaksi sistem

1. Aktor mengetik pesan pada kolom chat.	2. Sistem mengirim pesan dan menampilkannya di jendela percakapan.
Kondisi akhir	Pesan berhasil terkirim dan tampil pada sesi chat.

Table 4.14 Skenario Use Case Menerima Pesan

Identifikasi	
Kode - Nama	UC-07 - Menerima Pesan
Tujuan	Memungkinkan Admin, <i>Customer service</i> dan <i>Customer</i> menerima pesan sesuai hak aksesnya.
Deskripsi	Use case ini menjelaskan proses penerimaan pesan dalam sesi chat, di mana Admin hanya memantau percakapan tanpa ikut berinteraksi langsung.
Aktor	Admin, <i>Customer service</i> , <i>Customer</i>
Kondisi awal	Sesi chat sedang berlangsung.
Skenario Utama	
Aksi aktor	Reaksi sistem
1. Pesan dikirim oleh pengirim.	2. Sistem menampilkan pesan kepada penerima sesuai perannya.
Kondisi akhir	Pesan berhasil diterima oleh penerima.

Table 4.15 Skenario Use Case Menampilkan Identitas Pengirim Pesan

Identifikasi	
Kode - Nama	UC-08 - Menampilkan Identitas Pengirim Pesan
Tujuan	Menampilkan identitas pengirim setiap pesan agar penerima mengetahui sumber pesan.
Deskripsi	Use case ini memastikan bahwa setiap pesan yang dikirim oleh <i>Customer service</i> atau <i>Customer</i> menampilkan nama dan peran pengirim.
Aktor	<i>Customer service</i> dan <i>Customer</i>
Kondisi awal	Sesi chat aktif dan pesan terkirim.
Skenario Utama	
Aksi aktor	Reaksi sistem
1. Aktor mengirim pesan.	2. Sistem menampilkan identitas pengirim (nama) pada pesan.
Kondisi akhir	Pesan ditampilkan dengan identitas pengirim yang sesuai.

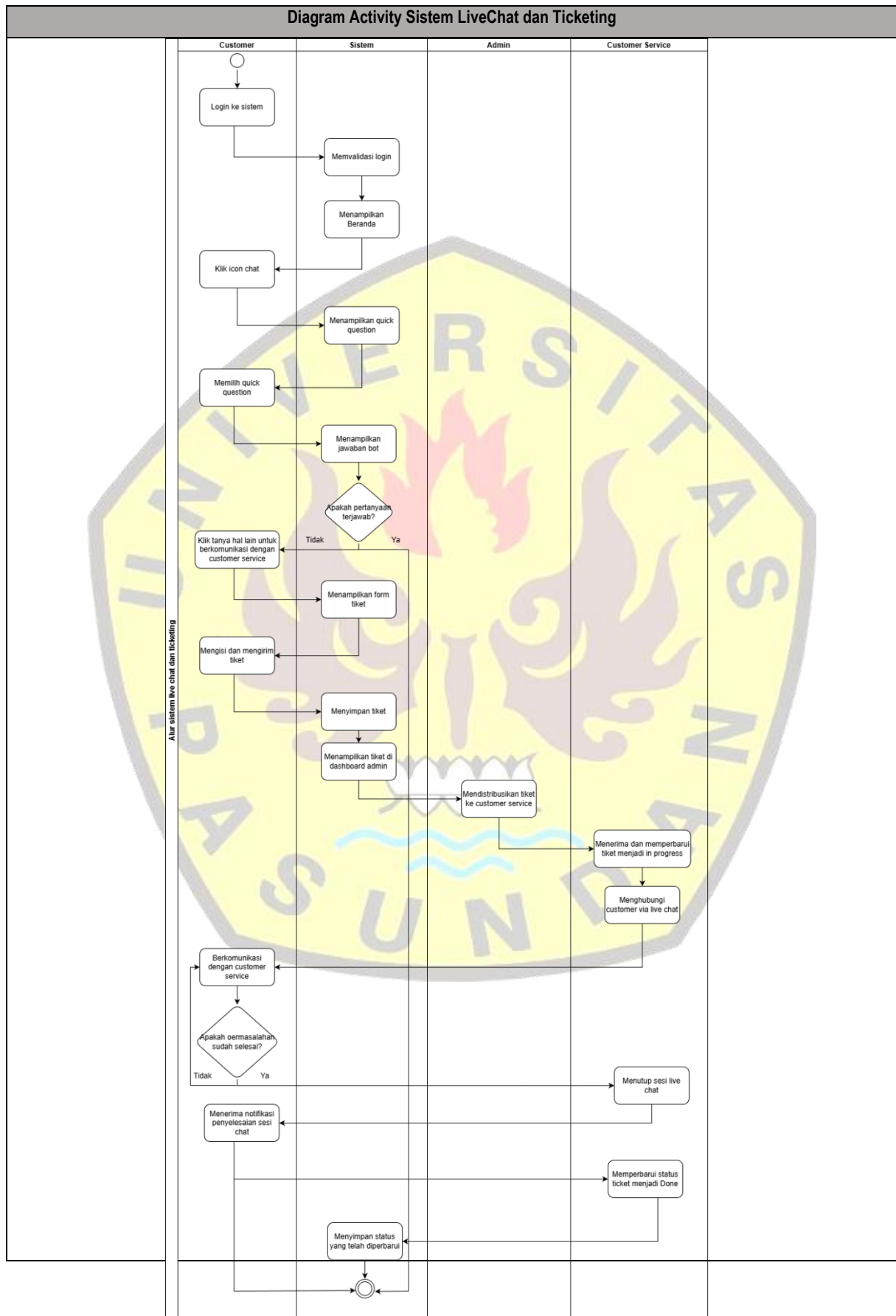
Table 4.16 Skenario Use Case Melihat Pesan yang Tersimpan

Identifikasi	
Kode - Nama	UC-9 - Melihat Pesan yang Tersimpan
Tujuan	Memungkinkan Admin, <i>Customer service</i> , dan <i>Customer</i> melihat riwayat percakapan yang tersimpan di sistem.
Deskripsi	Use case ini menggambarkan akses riwayat chat bagi semua aktor. Admin dapat memantau seluruh percakapan, <i>Customer service</i> melihat riwayat dengan <i>Customer</i> , dan <i>Customer</i> melihat riwayat chat pribadinya.
Aktor	Admin, <i>Customer service</i> , <i>Customer</i>
Kondisi awal	Aktor telah login ke sistem.
Skenario Utama	
Aksi aktor	Reaksi sistem
1. Pengguna membuka halaman chat.	2. Sistem menampilkan daftar sesi chat yang tersedia.
3. Pengguna melihat daftar pesan yang telah dikirim dan diterima.	4. Sistem menampilkan isi pesan secara berurutan sesuai waktu pengiriman.
5. Pengguna menggulir ke atas untuk melihat pesan lama.	6. Sistem menampilkan pesan yang tersimpan.
Kondisi akhir	Riwayat percakapan berhasil ditampilkan kepada pengguna.

4.2.9 Aliran Aktivitas

Activity diagram merupakan diagram yang digunakan untuk memodelkan proses atau aktivitas yang terjadi dalam suatu sistem. Diagram ini menampilkan runtutan proses secara vertikal, mulai dari awal hingga akhir aktivitas, dan merupakan pengembangan dari use case yang menggambarkan alur aktivitas secara lebih rinci. Alur aktivitas tersebut dapat divisualisasikan pada Tabel 4.17 Diagram Activity Sistem LiveChat dan Ticketing berikut.

Tabel 4.17 Diagram Activity Sistem LiveChat dan Ticketing



Gambar 4.3 Current Activity Diagram Sistem Live Chat dan Ticketing
Deskripsi
Diagram ini menunjukkan alur layanan komunikasi <i>Customer</i> mulai dari login, percakapan melalui live chat, pencatatan keluhan oleh <i>Customer service</i> , pembuatan tiket oleh admin jika diperlukan, hingga penyelesaian masalah dengan mengalihkan ke kurir apabila dibutuhkan. Proses ini memastikan penanganan tercatat secara terstruktur dan transparan.

4.3. Pemodelan Data

Pemodelan data adalah proses yang digunakan untuk merancang dan menyusun struktur data yang akan digunakan dalam suatu sistem atau perangkat lunak.

4.3.1 Objek Data

Objek data adalah kumpulan elemen yang memiliki tipe data tertentu dan berfungsi sebagai representasi informasi yang perlu dipahami serta diolah oleh perangkat lunak. Objek data utama yang digunakan dalam sistem dijelaskan secara rinci pada Tabel 4.18 Objek Data berikut.

Tabel 4.18 Objek Data

No	Nama Objek	Deskripsi
1.	User	Objek ini digunakan untuk merepresentasikan akun pengguna (baik <i>Customer service</i> maupun <i>Customer</i>), yang mencakup identitas, email, kata sandi, dan status online.
2.	Chat	Objek ini digunakan untuk merepresentasikan sesi percakapan antara pengguna dan <i>Customer service</i> , mencatat siapa yang terlibat, status sesi, topik percakapan, serta waktu mulai dan selesai.
3.	Ticket	Objek ini digunakan untuk merepresentasikan keluhan atau permintaan layanan pengguna yang dicatat dalam sistem untuk diproses, dipantau, dan diselesaikan oleh <i>Customer service</i> .

Berdasarkan Tabel 4.18 Objek Data, objek data utama dalam sistem mencakup informasi pengguna, sesi chat, pesan, status online, dan riwayat percakapan. Objek-objek ini menjadi dasar dalam pengelolaan data pada fitur live chat.

4.3.2 Atribut Data

Atribut data adalah properti atau karakteristik yang dimiliki oleh suatu objek data. Daftar atribut dari objek-objek yang telah ditentukan sebelumnya disajikan pada Tabel 4.19 Atribut Data berikut.

Tabel 4.19 Atribut Data

No	Nama Objek	Atribut	Deskripsi
1.	User	Id	Atribut ini digunakan untuk menyimpan identifer unik pengguna.
		Name	Atribut ini digunakan untuk menyimpan nama lengkap pengguna.
		Email	Atribut ini digunakan untuk menyimpan alamat email pengguna untuk proses autentikasi dan komunikasi.
		Role	Atribut ini digunakan untuk menyimpan peran pengguna, yaitu admin, <i>Customer service</i> , atau <i>Customer</i> .
		Status	Atribut ini digunakan untuk menyimpan status akun pengguna.
		createdAt	Atribut ini digunakan untuk menyimpan waktu pembuatan data pengguna di sistem.
2.	Chat	<i>CustomerId</i>	Atribut ini digunakan untuk menyimpan identifer unik untuk setiap sesi chat.
		<i>CustomerName</i>	Atribut ini digunakan untuk menyimpan nama <i>Customer</i> .
		<i>CustomerPhone</i>	Atribut ini digunakan untuk menyimpan nomor telepon <i>Customer</i> .
		<i>CustomerEmail</i>	Atribut ini digunakan untuk menyimpan alamat email <i>Customer</i> .
		resi	Atribut ini digunakan untuk menyimpan nomor resi terkait percakapan.
		createdAt	Atribut ini digunakan untuk menyimpan waktu pembuatan sesi chat.
		csId	Atribut ini digunakan untuk menyimpan identifer <i>Customer service</i> yang menangani chat.
3.	Message	senderId	Atribut ini digunakan untuk menyimpan identifer pengirim pesan.
		sender	Atribut ini digunakan untuk menyimpan nama atau jenis pengirim pesan.
		receiverId	Atribut ini digunakan untuk menyimpan identifer penerima pesan.
		text	Atribut ini digunakan untuk menyimpan isi pesan yang dikirimkan.
		createdAt	Atribut ini digunakan untuk menyimpan waktu pengiriman pesan.
4.	Ticket	noTicket	Atribut ini digunakan untuk menyimpan nomor tiket unik sebagai identitas utama tiket layanan.
		noResi	Atribut ini digunakan untuk menyimpan nomor resi pengiriman yang berkaitan dengan tiket yang diajukan.
		description	Atribut ini digunakan untuk menyimpan deskripsi atau rincian permasalahan yang disampaikan oleh <i>Customer</i> .
		priority	Atribut ini digunakan untuk menyimpan tingkat prioritas penanganan tiket, seperti rendah, sedang, atau tinggi.
		status	Atribut ini digunakan untuk menyimpan status penanganan tiket, yaitu open, in_progress, atau closed.
		<i>CustomerId</i>	Atribut ini digunakan untuk menyimpan identitas <i>Customer</i> yang membuat tiket.
		assignedCsId	Atribut ini digunakan untuk menyimpan identitas <i>Customer service</i> yang ditugaskan menangani tiket (bersifat opsional).
		createdAt	Atribut ini digunakan untuk menyimpan waktu pembuatan tiket di dalam sistem.

Berdasarkan Tabel 4.19 Atribut Data, atribut digunakan untuk menyimpan informasi spesifik dari masing-masing objek data. Misalnya, atribut pada objek User menyimpan identitas dan status pengguna, atribut pada ChatSession mencatat waktu dan topik sesi, sedangkan atribut pada Message menyimpan isi pesan dan waktu pengiriman. Atribut-atribut tersebut berfungsi untuk mendukung operasional sistem live chat sesuai kebutuhan masing-masing objek.

4.3.3 Pemodelan Berbasis Perilaku

Pemodelan berbasis perilaku digambarkan menggunakan sequence diagram, yang berfungsi untuk memvisualisasikan interaksi antar objek atau komponen sistem secara berurutan dalam waktu tertentu. Diagram ini menunjukkan alur pertukaran pesan dan urutan eksekusi proses yang terjadi antara aktor dan sistem. Berikut adalah diagram sequence yang menggambarkan interaksi tersebut.



Tabel 4.20 Diagram Sequence Mengelola Akun Pengguna

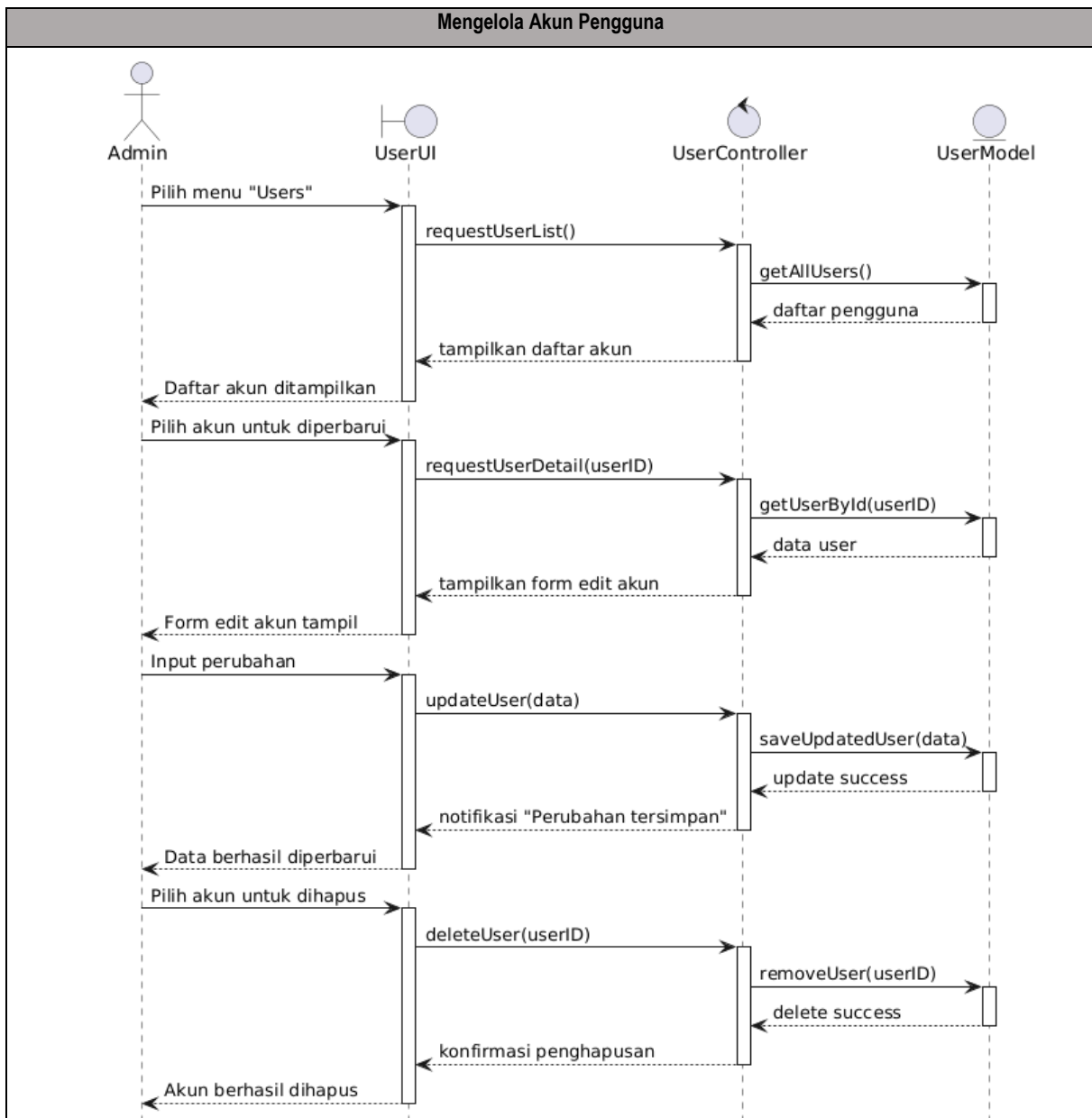
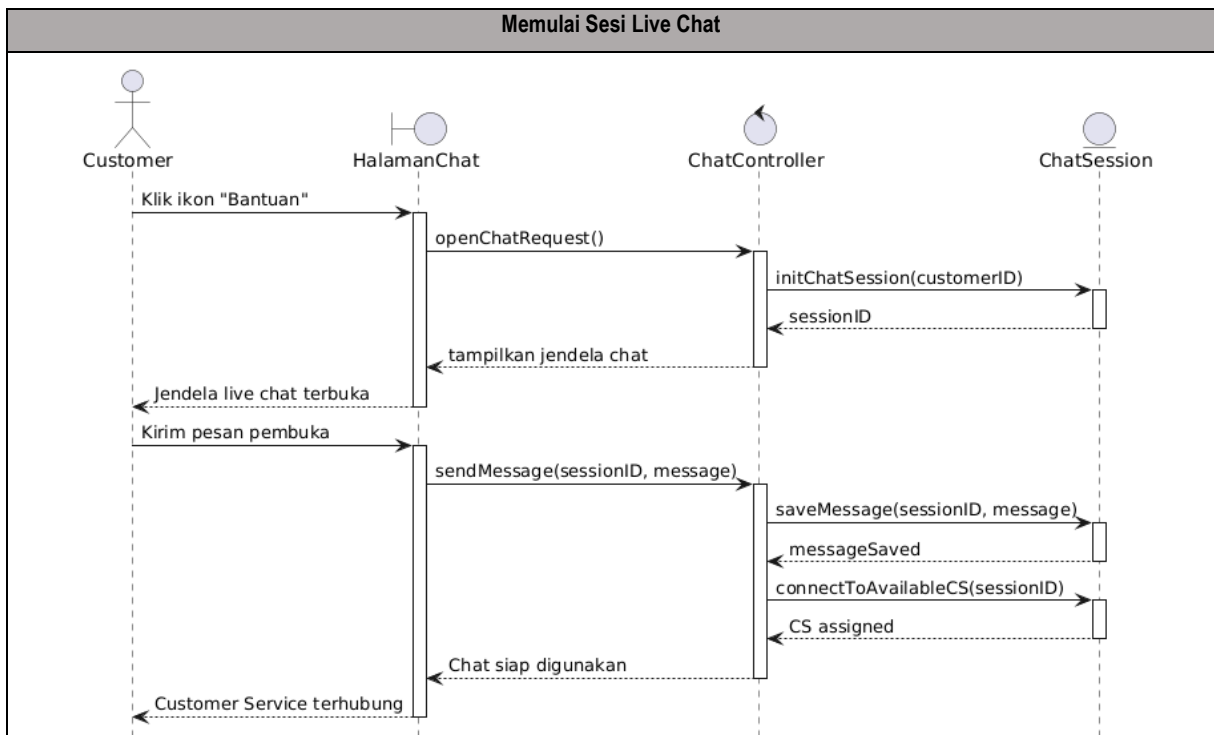
Gambar 4.4 Diagram Sequence Mengelola Akun *Customer service***Deskripsi**

Diagram ini menggambarkan bagaimana admin mengelola akun pengguna, mulai dari melihat daftar user, memperbaiki data seperti peran dan status akun, hingga menonaktifkan atau menghapus akun yang tidak diperlukan lagi oleh sistem.

Tabel 4.21 Diagram Sequence Memulai Sesi Live Chat

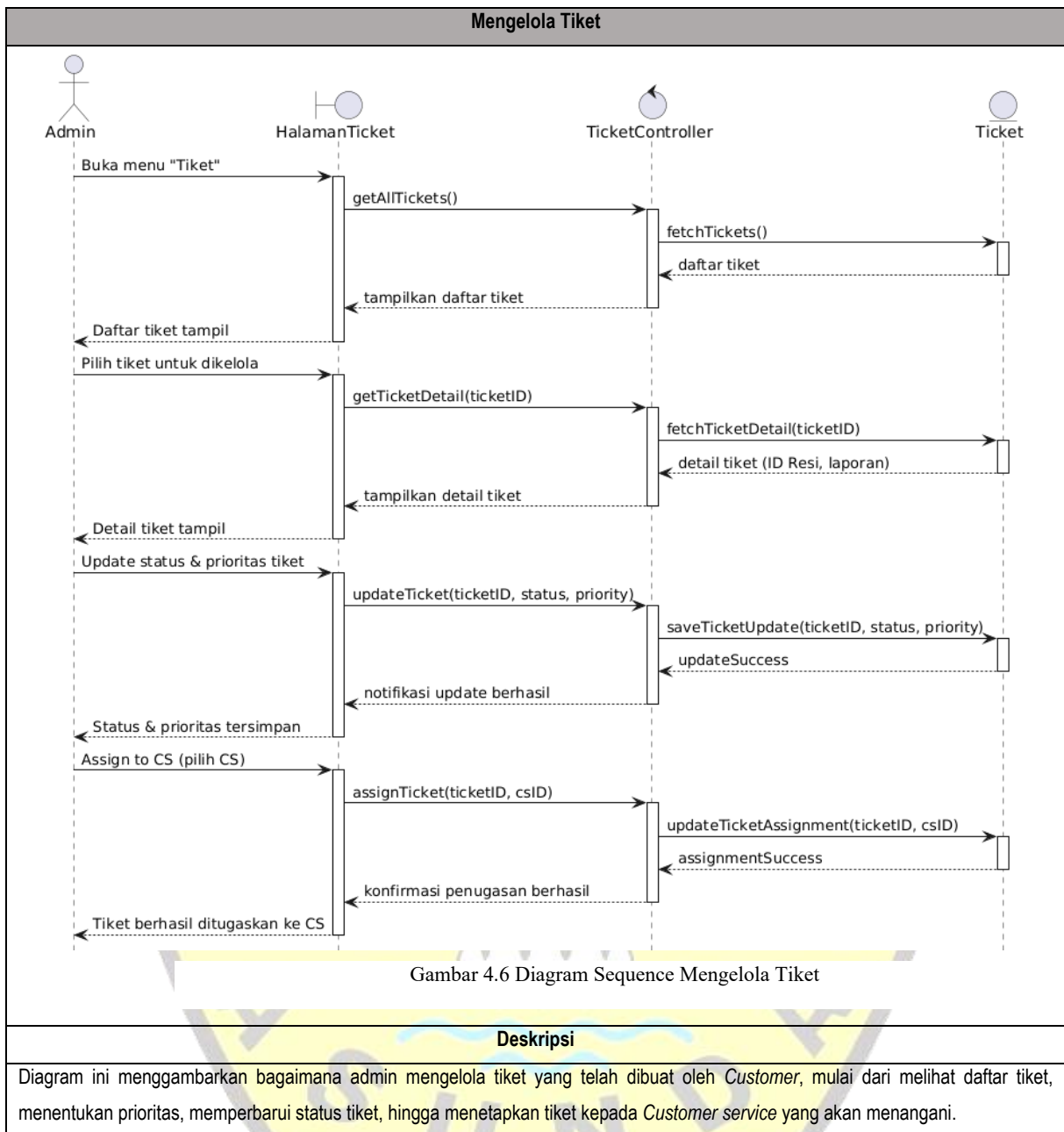


Gambar 4.5 Diagram Sequence Memulai Sesi Live Chat

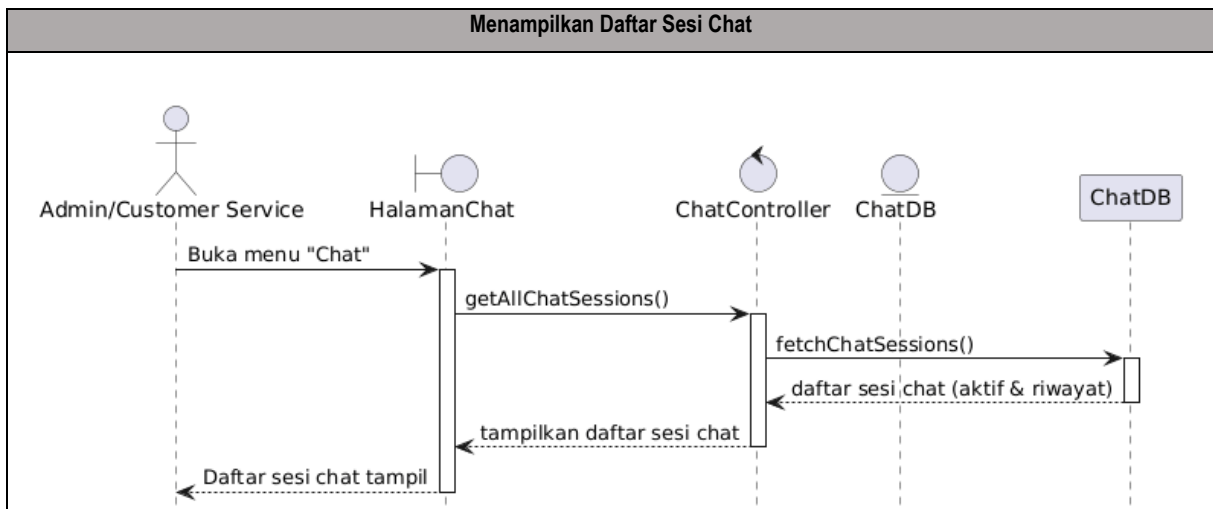
Deskripsi

Diagram ini menggambarkan bagaimana *Customer* memulai sesi live chat, dimulai dari memilih menu bantuan atau layanan chat, kemudian sistem membuat sesi baru sehingga *Customer* dapat terhubung dengan *Customer service* yang tersedia.

Tabel 4.22 Diagram Sequence Mengelola Tiket



Tabel 4.23 Diagram Sequence Menampilkan Daftar Sesi Chat

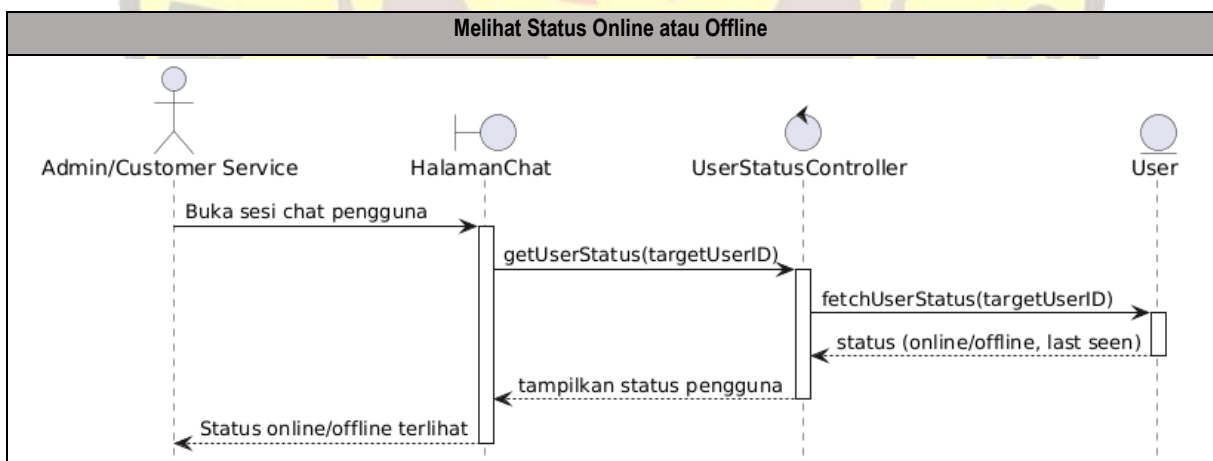


Gambar 4.7 Diagram Sequence Menampilkan Daftar Sesi Chat

Deskripsi

Diagram ini menggambarkan bagaimana admin dan *Customer service* menampilkan daftar sesi percakapan, baik yang sedang berlangsung maupun yang telah selesai, untuk memantau dan mengakses interaksi yang terjadi antara *Customer* dan *Customer service*.

Tabel 4.24 Diagram Sequence Melihat Status Online atau Offline

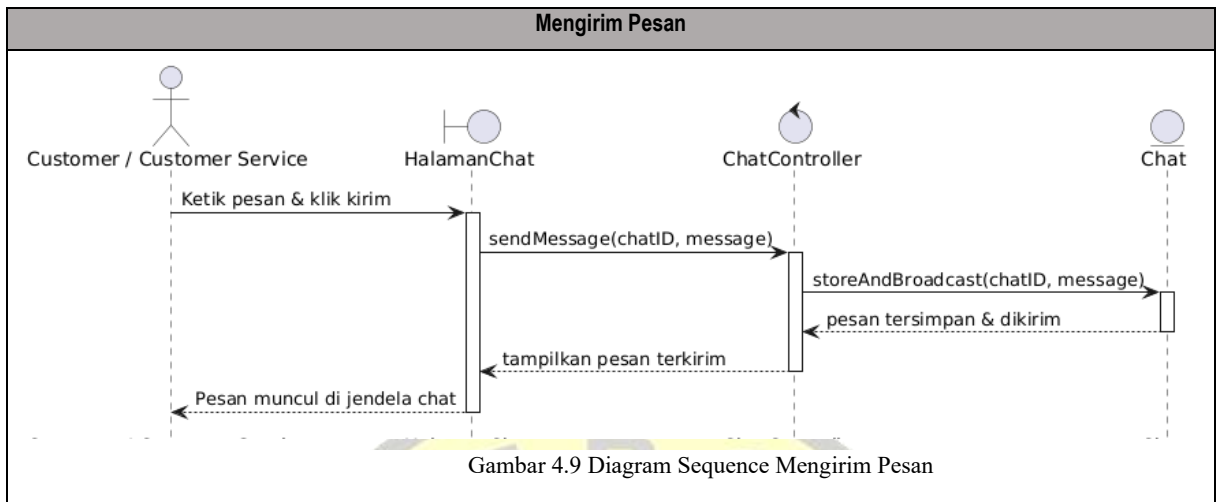


Gambar 4.8 Diagram Sequence Melihat Status Online atau Offline

Deskripsi

Diagram ini menggambarkan bagaimana admin dan *Customer service* melihat status online atau offline pengguna lain dalam sistem, sehingga dapat diketahui siapa yang sedang aktif dan dapat memberikan respon.

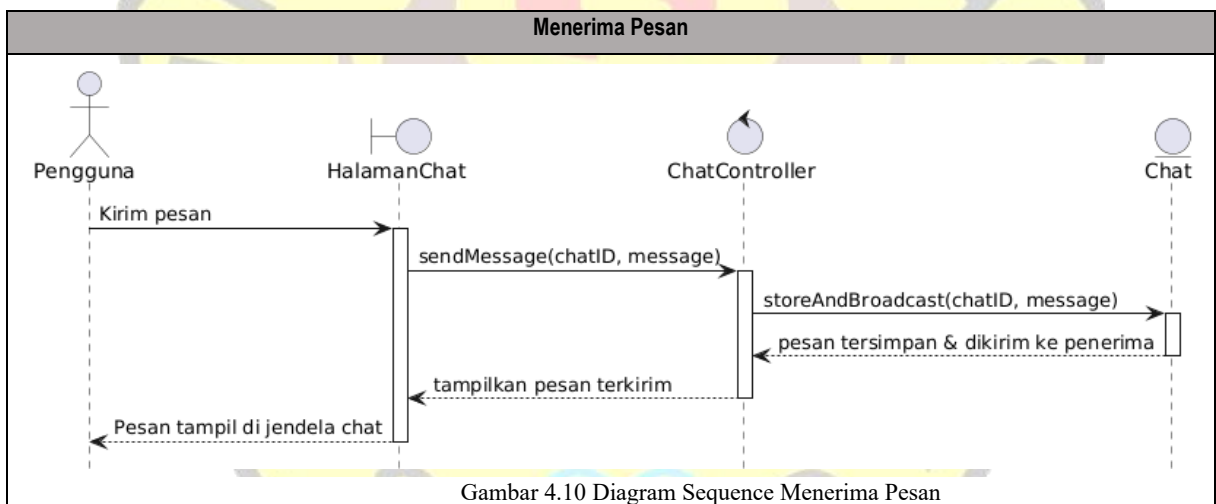
Tabel 4.25 Diagram Sequence Mengirim Pesan



Deskripsi

Diagram ini menggambarkan bagaimana *Customer* dan *Customer service* mengirim pesan selama sesi chat berlangsung, mulai dari penulisan pesan hingga pesan tersebut dikirim dan diterima oleh sistem.

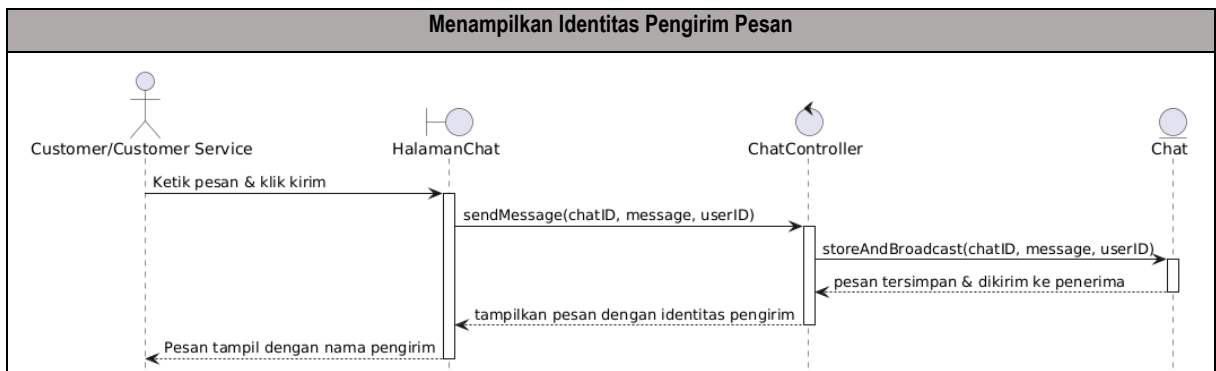
Tabel 4.26 Diagram Sequence Menerima Pesan



Deskripsi

Diagram ini menggambarkan bagaimana admin, *Customer service*, dan *Customer* menerima pesan dalam sistem chat dan admin hanya bersifat memantau, sedangkan *Customer service* dan *Customer* menerima pesan sebagai bagian dari interaksi dua arah.

Table 4.27 Diagram Sequence Menampilkan Identitas Pengirim Pesan

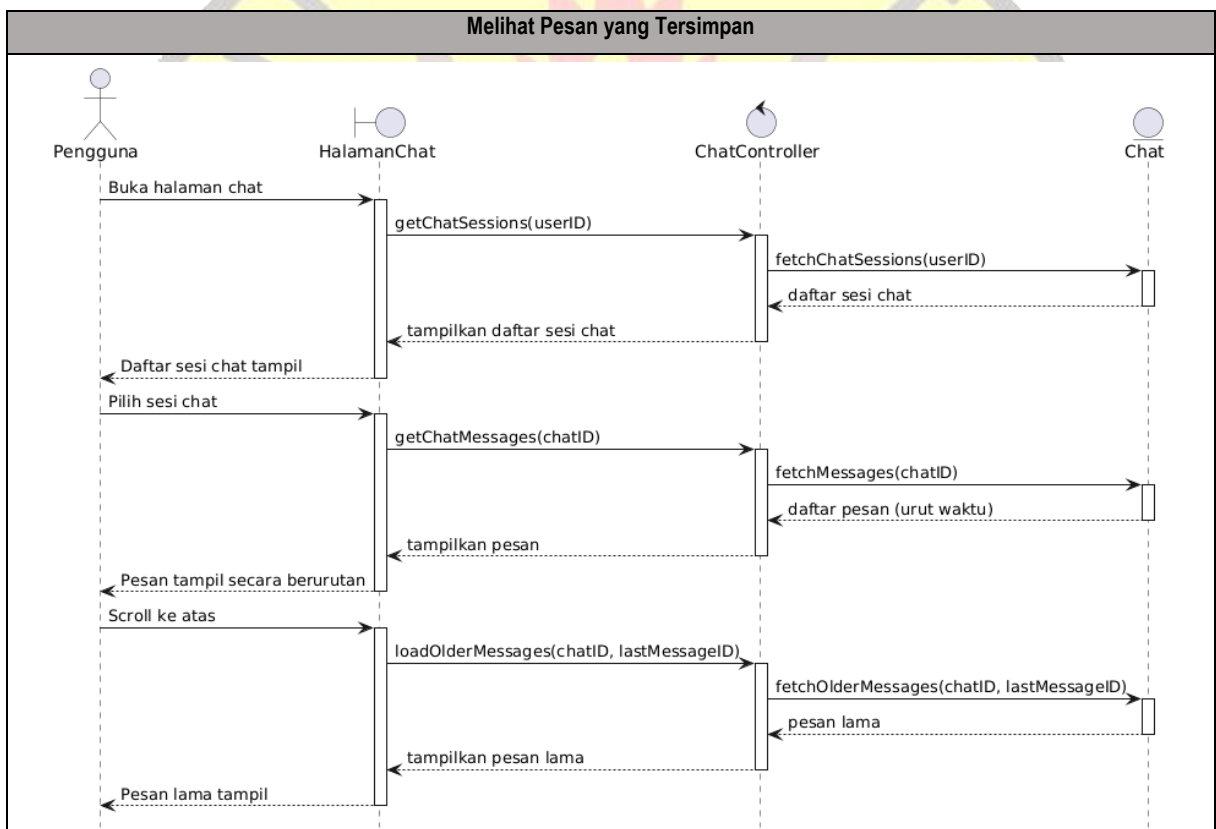


Gambar 4.11 Diagram Sequence Menampilkan Identitas Pengirim Pesan

Deskripsi

Diagram ini menggambarkan bagaimana sistem menampilkan identitas pengirim pada setiap pesan, sehingga penerima dapat mengetahui apakah pesan berasal dari *Customer service* atau *Customer*.

Tabel 4.28 Diagram Sequence Melihat Pesan yang Tersimpan



Gambar 4.12 Diagram Sequence Melihat Pesan yang Tersimpan

Deskripsi

Diagram ini menggambarkan bagaimana admin, *Customer service*, dan *Customer* melihat riwayat percakapan yang telah tersimpan, sehingga percakapan sebelumnya dapat ditelusuri kembali untuk keperluan dokumentasi atau tindak lanjut layanan.

4.4. Pemodelan Berbasis Kelas

Pemodelan kelas berbasis objek digunakan untuk memvisualisasikan objek-objek yang akan dimanipulasi oleh sistem atau perangkat lunak. Pemodelan ini juga menggambarkan operasi-operasi yang diterapkan pada objek, hubungan antar objek, serta kolaborasi yang terjadi antara kelas-kelas yang telah didefinisikan.

4.4.1 Identifikasi Kelas-Kelas Analisis

Proses identifikasi kelas dilakukan dengan memeriksa skenario sistem atau perangkat lunak yang akan dikembangkan, yang dijadikan acuan dalam pembuatan kelas-kelas analisis. Kelas-kelas analisis yang dihasilkan dapat dilihat pada Tabel 4.29 Identifikasi Kelas-kelas Analisis berikut.

Tabel 4.29 Identifikasi Kelas-kelas Analisis

No	Nama kelas	Jenis	Deskripsi
1.	User	Model	Kelas ini digunakan untuk menyimpan data pengguna (<i>Customer</i> dan <i>Customer service</i>), termasuk identitas, status online/offline, dan data login.
2.	Chat	Model	Kelas ini digunakan untuk menyimpan data sesi percakapan antara pengguna dan <i>Customer service</i> , status aktif atau selesai.
3.	Message	Model	Kelas ini digunakan untuk menyimpan isi pesan yang dikirim/diterima, identitas pengirim, waktu, dan status pesan.
4.	AuthController	Controller	Kelas ini digunakan untuk mengelola proses autentikasi dan otorisasi pengguna, meliputi registrasi, login, verifikasi token, pengelolaan sesi, pengambilan profil pengguna, serta pengecekan peran berbasis Role-Based Access Control (RBAC).
5.	ChatController	Controller	Kelas ini digunakan untuk mengelola fitur live chat, termasuk pembuatan sesi chat, pengiriman dan pengambilan pesan secara <i>Real-time</i> , pengelolaan data percakapan, serta pengakhiran sesi chat yang terintegrasi dengan sistem ticketing dan <i>Firebase Cloud Firestore</i> .
6.	TicketController	Controller	Kelas ini digunakan untuk mengelola proses ticketing, meliputi pembuatan, pengambilan, pembaruan, penugasan, penutupan, dan penghapusan tiket sesuai dengan alur bisnis sistem dengan memanfaatkan <i>Firebase Cloud Firestore</i> .
7.	UserController	Controller	Kelas ini digunakan untuk mengelola data pengguna, meliputi pembuatan, pengambilan, pembaruan, dan penghapusan user, serta pengaturan peran dan status pengguna dengan integrasi <i>Firebase Authentication</i> dan <i>Firestore</i> .
8.	LoginPage	View	Kelas ini digunakan untuk menyediakan tampilan halaman login bagi pengguna dengan memasukkan email dan password, dilengkapi validasi data serta fitur lupa password.
9.	RegistrasiPage	View	Kelas ini digunakan untuk menyediakan tampilan halaman registrasi bagi pengguna dalam membuat akun baru dengan validasi data pengguna.
10.	ResetPasswordPage	View	Kelas ini digunakan untuk menyediakan tampilan halaman reset password dengan memasukkan email yang telah terdaftar pada sistem.
11.	DashboardPage	View	Kelas ini digunakan untuk menyediakan halaman utama <i>Customer</i> yang menampilkan informasi layanan, FAQ, fitur live chat, dan tiket.

No	Nama kelas	Jenis	Deskripsi
12.	ListTicketPage	View	Kelas ini digunakan untuk menampilkan daftar tiket dalam bentuk tabel beserta aksi pengelolaan tiket.
13.	DetailTicketPage	View	Kelas ini digunakan untuk menampilkan detail tiket secara read-only, yang memuat informasi lengkap tiket.
14.	EditTicketPage	View	Kelas ini digunakan sebagai halaman pengelolaan tiket untuk memperbarui prioritas dan melakukan penugasan <i>Customer service</i> .
15.	ListTicketRecapPage	View	Kelas ini digunakan untuk menampilkan rekap daftar tiket dalam bentuk tabel.
16.	DetailTicketRecapPage	View	Kelas ini digunakan untuk menampilkan detail rekap tiket secara read-only.
17.	ChatHistoryPage	View	Kelas ini digunakan untuk menampilkan riwayat live chat, termasuk daftar dan isi percakapan yang telah berlangsung.
18.	ChatPage	View	Kelas ini digunakan untuk menyediakan halaman live chat <i>Customer service</i> untuk komunikasi <i>Real-time</i> dengan <i>Customer</i> .

Berdasarkan Tabel 4.29 Identifikasi Kelas-kelas Analisis, kelas-kelas analisis dikelompokkan menjadi tiga jenis utama, yaitu Model, Controller, dan View. Kelas Model berfungsi menyimpan data inti seperti pengguna, sesi chat, dan pesan. Kelas Controller mengelola logika proses seperti pengiriman pesan, pengelolaan tiket, dan status online. Sementara itu, kelas View digunakan untuk menampilkan antarmuka kepada pengguna sesuai fungsinya. Identifikasi ini membantu merancang struktur aplikasi sesuai dengan peran masing-masing kelas dalam sistem.

4.4.2 Menentukan Atribut

Atribut merupakan karakteristik yang membedakan satu kelas dengan kelas lainnya. Daftar atribut dari kelas-kelas yang telah didefinisikan sebelumnya disajikan pada Tabel 4.30 Menentukan Atribut berikut.

Tabel 4.30 Menentukan Atribut

No	Nama kelas	Nama atribut
1.	User	Id
		Name
		Email
		Role
		Status
		createdAt
2	Chat	CustomerId
		CustomerName
		CustomerPhone
		CustomerEmail
		resi
		createdAt
3.	Message	csId
		senderId

No	Nama kelas	Nama atribut
		sender
		receiverId
		text
		createdAt
4.	Ticket	noTicket
		noResi
		description
		priority
		status
		CustomerId
		assignedCsId
		createdAt

Berdasarkan Tabel 4.30 Menentukan Atribut, setiap kelas model dalam sistem memiliki atribut yang merepresentasikan karakteristik dan data utama yang dikelolanya. Kelas User menyimpan informasi identitas dan status pengguna, kelas Chat merepresentasikan data sesi percakapan antara *Customer* dan *Customer service*, kelas Message digunakan untuk menyimpan isi dan informasi pesan yang dikirim dalam sesi chat, sedangkan kelas Ticket menyimpan data tiket layanan yang berkaitan dengan keluhan atau permintaan *Customer*, termasuk status dan prioritas penanganannya.

4.4.3 Menentukan Perilaku

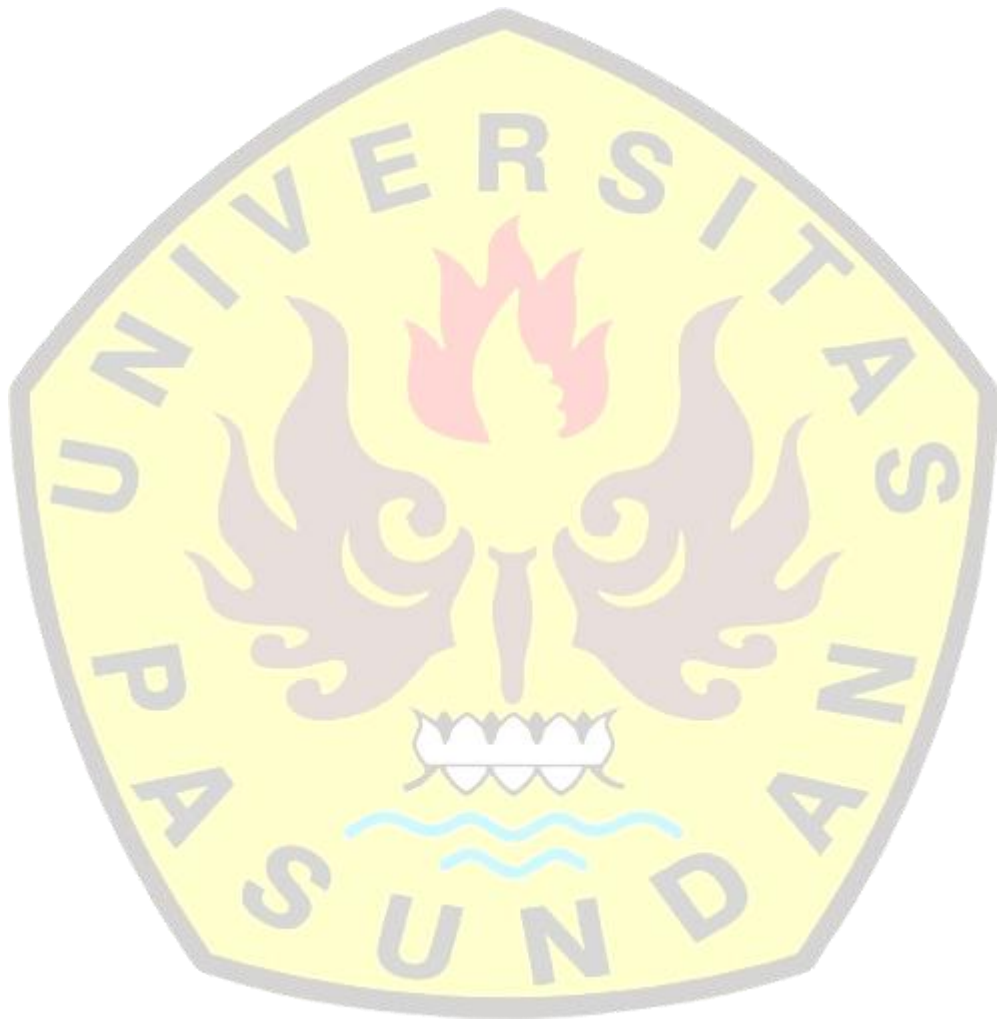
Pada tahap ini, ditetapkan perilaku atau fungsi utama yang dimiliki setiap kelas dalam sistem. Perilaku ini menggambarkan aksi-aksi yang dapat dilakukan oleh masing-masing kelas dalam menjalankan proses pada aplikasi. Rincian perilaku tiap kelas disajikan pada Tabel 4.31 Menentukan Perilaku berikut.

Tabel 4.31 Menentukan Perilaku

No	Nama kelas	Deskripsi	Perilaku
1.	AuthController	Mengelola proses autentikasi dan otorisasi pengguna dalam sistem	registerUser()
		Mendaftarkan pengguna baru dan menyimpan data ke <i>Firebase Authentication</i> dan <i>Firestore</i>	loginUser()
		Melakukan proses login pengguna dan validasi peran (role)	verifyToken()
		Memverifikasi token autentikasi untuk mengamankan akses ke sistem	logoutUser()
		Mengakhiri sesi pengguna dengan menghapus token autentikasi	checkRole()
		Mengambil dan menampilkan data profil pengguna yang sedang aktif	getProfile()
2.	ChatController	Membuat sesi chat baru berdasarkan data <i>Customer</i> dan menyimpannya ke basis data, serta menyimpan pesan pertama jika tersedia.	createChat()
		Mengirim dan menyimpan pesan ke dalam sesi chat tertentu.	sendMessage()
		Mengambil seluruh pesan dari satu sesi chat dan menampilkannya secara berurutan berdasarkan waktu.	getMessages()
		Mengambil daftar seluruh sesi chat beserta pesan terakhir dari masing-masing chat.	getChatsForUser()
		Mengambil detail data sesi chat berdasarkan identifier chat.	getChatById()
		Mengakhiri sesi chat dengan memperbarui status chat dan menyesuaikan status tiket terkait menjadi selesai.	endChat()
3.	TicketController	Membuat tiket layanan baru dengan nomor tiket unik, menyimpan data tiket ke dalam basis data, serta mengaitkannya dengan sesi chat jika tersedia.	createTicket()
		Mengambil dan menampilkan seluruh data tiket yang tersimpan di sistem, diurutkan berdasarkan waktu pembuatan tiket.	getAllTickets()
		Mengambil dan menampilkan detail data tiket berdasarkan identifier tiket tertentu.	getTicketDetail()
		Memperbarui informasi tiket berdasarkan data yang dikirimkan, seperti deskripsi, status, atau atribut lainnya.	updateTicket()
		Memperbarui tingkat prioritas penanganan tiket sesuai dengan kebutuhan proses layanan.	updateTicketPriority()
		Menetapkan tiket kepada <i>Customer service</i> tertentu dan memperbarui status tiket menjadi ditugaskan.	assignTicketToCS()

No	Nama kelas	Deskripsi	Perilaku
		Mengambil daftar tiket baru yang belum ditangani oleh <i>Customer service</i> .	getListTicketsForCS()
		Menutup tiket layanan dengan memperbarui status tiket menjadi <i>closed</i> setelah permasalahan selesai ditangani.	closeTicket()
		Menghapus data tiket dari sistem berdasarkan identifier tiket.	deleteTicket()
4.	UserController	Mengambil dan menampilkan seluruh data pengguna yang tersimpan di dalam sistem.	getUsers()
		Mengambil dan menampilkan data pengguna berdasarkan identifier (ID) tertentu.	getUserById()
		Membuat akun pengguna baru dengan menyimpan data pengguna ke <i>Firebase Authentication</i> dan <i>Firestore</i> .	createUser()
		Memperbarui data pengguna, seperti nama, peran (role), dan status, baik pada <i>Firebase Authentication</i> maupun <i>Firestore</i> .	updateUser()
		Menghapus data pengguna dari sistem, termasuk menghapus akun dari <i>Firebase Authentication</i> dan <i>Firestore</i> .	deleteUser()
		Mengambil seluruh data pengguna sebagai fungsi alternatif untuk kebutuhan tampilan atau manajemen data.	getAllUsers()
		Memperbarui status online atau offline pengguna serta mencatat waktu terakhir aktivitas pengguna (<i>last seen</i>).	setUserOnlineStatus()

Berdasarkan Tabel 4.31, perilaku tiap kelas controller merepresentasikan fungsi utama sistem sesuai dengan perannya. *AuthController* menangani proses autentikasi dan otorisasi pengguna, *ChatController* mengelola sesi dan pesan percakapan, *TicketController* mengelola proses tiket layanan, serta *UserController* bertanggung jawab terhadap pengelolaan data dan status pengguna dalam sistem.



4.5. Perancangan Perangkat Lunak

Perancangan perangkat lunak merupakan tahap proses yang bertujuan menerapkan berbagai teknik dan prinsip untuk mewujudkan bentuk fisik dari perangkat lunak yang akan dikembangkan. Beberapa tahapan dalam perancangan perangkat lunak yang dilakukan meliputi perancangan data dan perancangan antarmuka.

4.5.1 Perancangan Data

Perancangan data merupakan tahap untuk memperjelas hasil pemodelan data yang telah mendefinisikan objek-objek data, atribut, serta relasi antar objek, sekaligus menyusun ERD (Entity Relationship Diagram). Perancangan data yang dihasilkan dapat dilihat pada tabel berikut.

Tabel 4.32 Perancangan Data

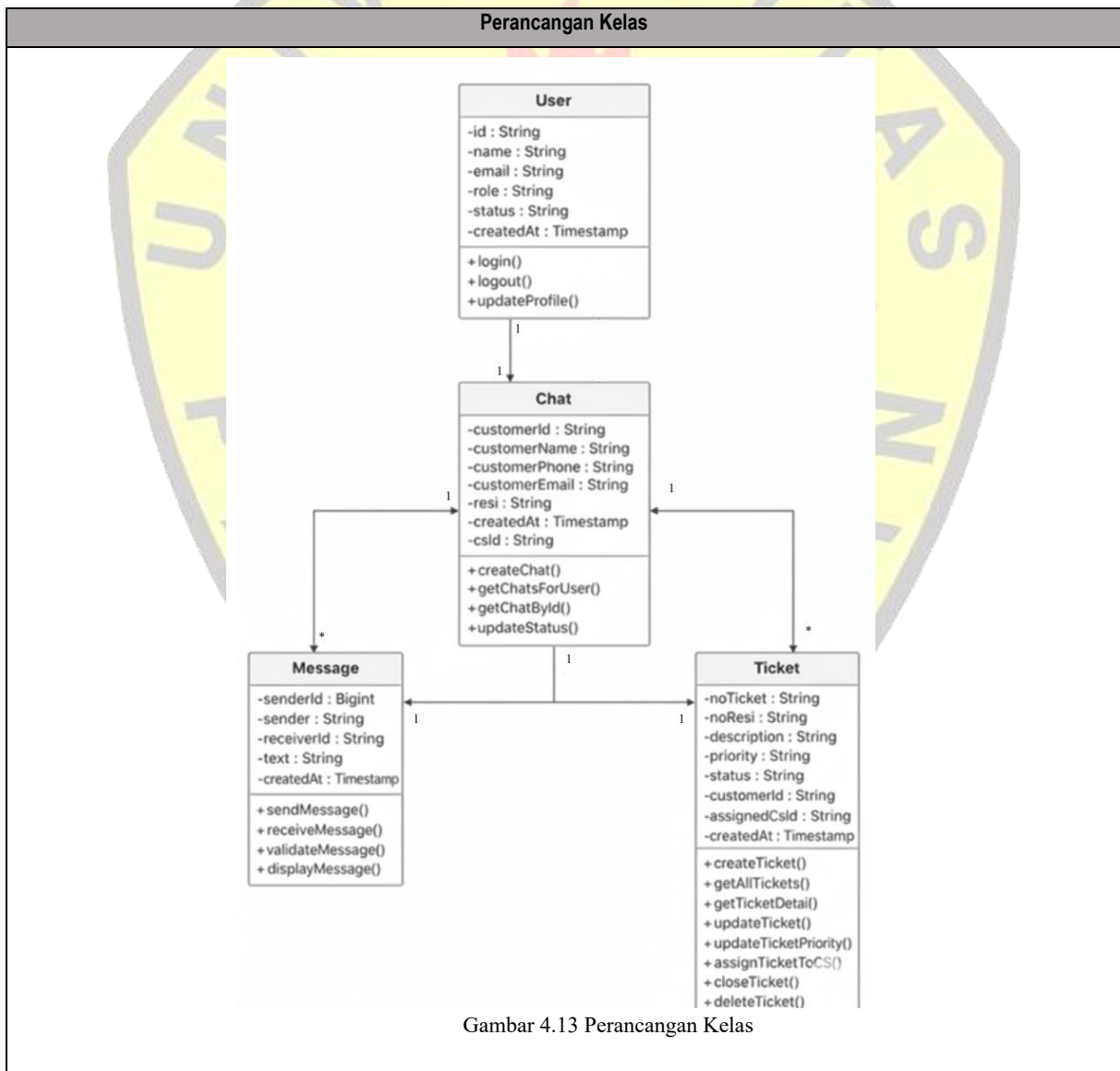
No	Nama objek	Atribut	Tipe data
1.	User	Id	String
		Name	String
		Email	String
		Role	String
		Status	String
		createdAt	Timestamp
2.	Chat	CustomerId	String
		CustomerName	String
		CustomerPhone	String
		CustomerEmail	String
		resi	String
		createdAt	Timestamp
		csId	String
3.	Message	senderId	Bigint
		sender	String
		receiverId	String
		text	String
		createdAt	Timestamp
4.	Ticket	noTicket	String
		noResi	String
		description	String
		priority	String
		status	String
		CustomerId	String
		assignedCsId	String
		createdAt	Timestamp

Berdasarkan Tabel 4.32, sistem ini memiliki empat objek data utama, yaitu User, Chat, Message, dan Ticket. Setiap objek memiliki atribut yang spesifik sesuai fungsinya. User menyimpan informasi identitas, email, peran, status, dan waktu pembuatan. Chat merepresentasikan sesi percakapan dengan atribut *CustomerId*, *CustomerName*, *CustomerPhone*, *CustomerEmail*, *resi*, *createdAt*, dan *csId*. Message menyimpan informasi pesan dengan atribut *senderId*, *sender*, *receiverId*, *text*, dan *createdAt*. Sedangkan Ticket mencatat tiket layanan yang diajukan oleh *Customer* dengan atribut *noTicket*, *noResi*, *description*, *priority*, *status*, *CustomerId*, *assignedCsId*, dan *createdAt*.

4.5.2 Perancangan Kelas

Perancangan kelas adalah tahap pembuatan diagram kelas berdasarkan kelas-kelas yang telah didefinisikan sebelumnya, beserta atribut dan operasinya. Diagram kelas yang dihasilkan dapat dilihat pada Tabel 4.33 Perancangan Kelas berikut.

Tabel 4.33 Perancangan Kelas

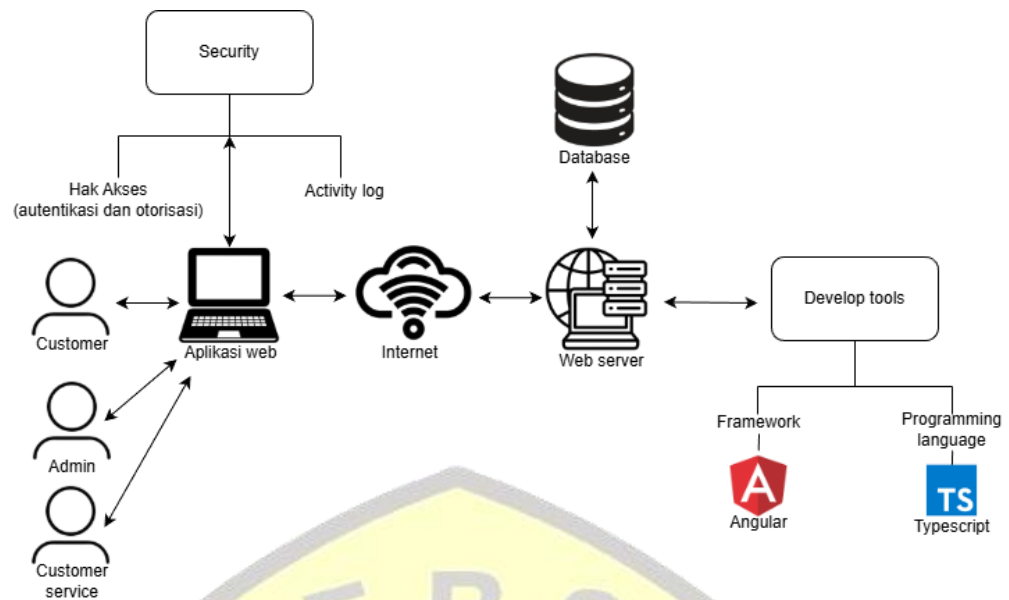


Berdasarkan Gambar 4.13 menggambarkan struktur sistem manajemen layanan pelanggan yang mengintegrasikan kelas User, Chat, Message, dan Ticket ke dalam satu kesatuan fungsional. Sistem ini memungkinkan User untuk memulai sesi Chat yang menampung rincian data pelanggan dan nomor resi, di mana setiap interaksi pesan dikelola secara mendetail melalui kelas Message. Apabila permasalahan pelanggan memerlukan tindak lanjut formal, sistem akan menghasilkan Ticket yang berfungsi untuk menetapkan prioritas, menugaskan petugas, dan melacak status penyelesaian masalah secara sistematis.

4.5.3 Perancangan Arsitektur

Perancangan arsitektur merupakan proses awal dalam pembangunan perangkat lunak yang bertujuan untuk menetapkan kerangka kerja untuk kontrol antar komponen yang terlibat. Arsitektur mencakup struktur sistem yang terdiri dari komponen dan hubungannya serta mempertimbangkan persyaratan fungsional dan non – fungsional. Dalam konteks desain arsitektur, komponen perangkat lunak dapat berupa sesuatu yang sederhana seperti modul program tetapi juga dapat diperluas untuk menyertakan basis data yang memungkinkan konfigurasi jaringan klien dan server [PRE20].

Perancangan arsitektur perangkat lunak pada sistem ini menggunakan pendekatan arsitektur client–server dengan tiga lapisan utama (3-tier architecture), yaitu lapisan presentasi, lapisan logika aplikasi, dan lapisan data. Lapisan presentasi direpresentasikan oleh aplikasi frontend berbasis web yang dibangun menggunakan framework *Angular* dan bahasa pemrograman TypeScript, yang berfungsi sebagai antarmuka pengguna dalam mengakses fitur live chat dan ticketing. Lapisan logika aplikasi diimplementasikan pada backend berbasis Node.js yang bertugas mengelola proses bisnis sistem, mengatur alur komunikasi antara frontend dan basis data, serta menerapkan mekanisme keamanan dan pengendalian akses pengguna. Lapisan ini menjadi penghubung utama antara pengguna dan data sistem. Lapisan data pada arsitektur ini menggunakan *Firebase Cloud Firestore* sebagai basis data yang menyimpan data pengguna, data tiket, dan riwayat percakapan live chat. Untuk mendukung aspek keamanan, sistem juga terintegrasi dengan *Firebase Authentication* dan aturan keamanan (security rules) yang berfungsi membatasi akses data sesuai dengan peran pengguna. Selain itu, proses pengembangan sistem didukung oleh berbagai development tools yang menunjang efisiensi dan kemudahan dalam pengelolaan aplikasi. Perancangan arsitektur dapat dilihat pada gambar 4.14 perancangan arsitektur berikut.

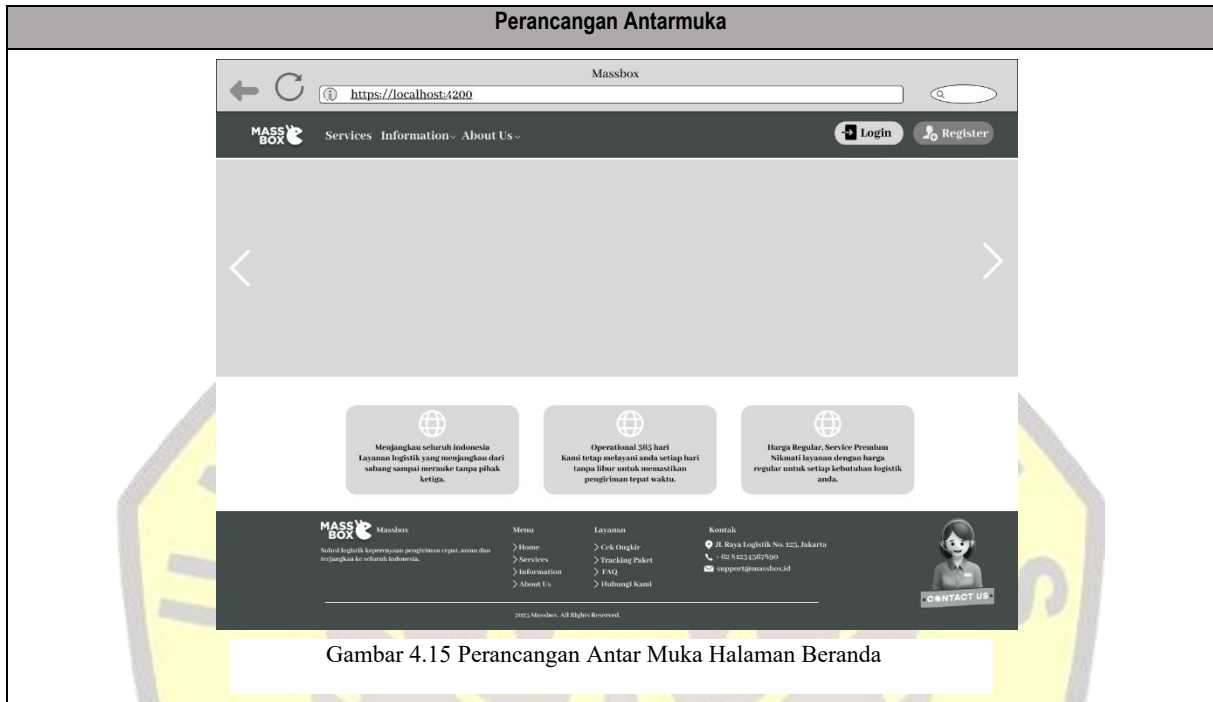


Berdasarkan Gambar 4.14, perancangan arsitektur sistem menggambarkan alur kerja dan hubungan antar komponen utama dalam sistem Live Chat dan Ticketing. Gambar tersebut menunjukkan bahwa pengguna mengakses sistem melalui aplikasi web berbasis *Angular* sebagai lapisan presentasi. Selanjutnya, permintaan pengguna diproses oleh backend Node.js pada lapisan logika aplikasi, yang berfungsi mengelola proses bisnis serta pengendalian hak akses. Data sistem disimpan pada *Firebase* Cloud Firestore sebagai lapisan data, dengan dukungan *Firebase* Authentication dan Security Rules untuk menjaga keamanan dan pembatasan akses.

4.6. Perancangan Antarmuka Pengguna

Perancangan antarmuka merupakan tahap untuk menggambarkan bagaimana antarmuka sistem atau perangkat lunak dirancang, dengan tujuan menciptakan komunikasi yang efektif yang memfasilitasi interaksi antara manusia dan komputer. Antarmuka pengguna dari perangkat lunak yang dikembangkan dapat dilihat pada tabel berikut.

Tabel 4.34 Perancangan Antarmuka Halaman Beranda

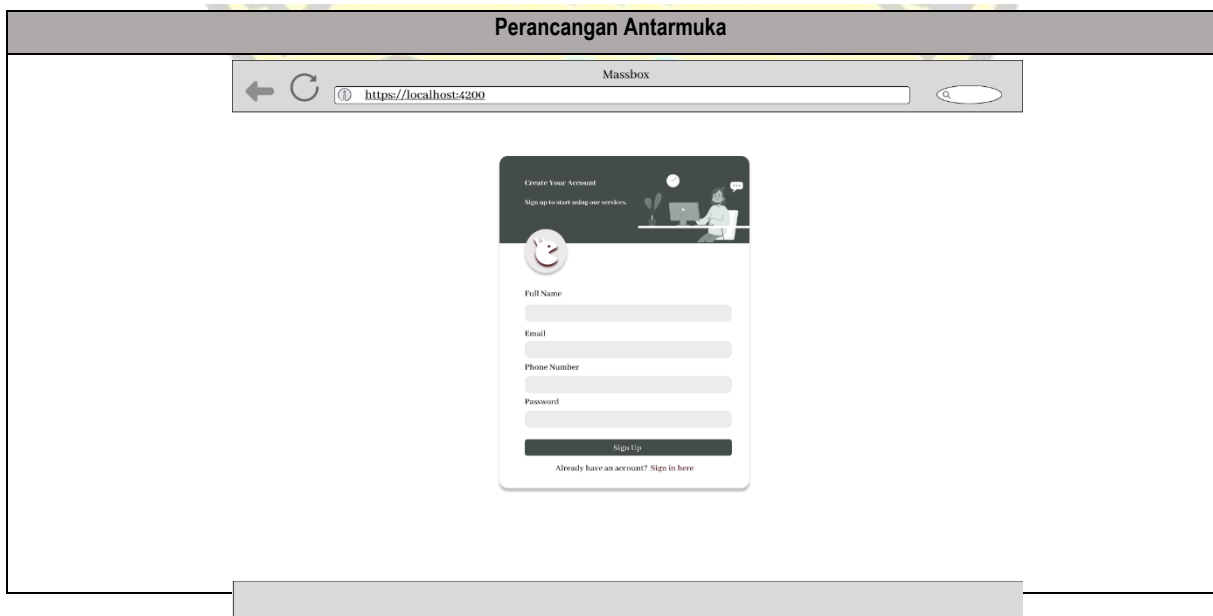


Gambar 4.15 Perancangan Antar Muka Halaman Beranda

Deskripsi

Halaman ini merupakan rancangan tampilan utama pada situs web Massbox yang berfungsi sebagai platform penyedia layanan pengiriman barang.

Tabel 4.35 Perancangan Antar Muka Halaman Register



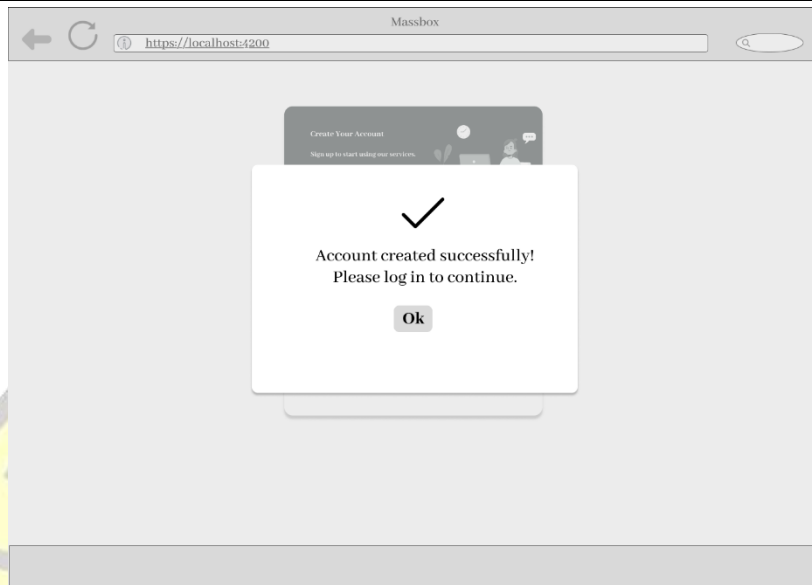
Gambar 4.16 Perancangan Antar Muka Halaman Register

Deskripsi

Halaman ini merupakan rancangan tampilan pendaftaran yang memuat formulir registrasi, yang terdiri atas kolom nama lengkap (*full name*), alamat surel (*email*), no telepon dan kata sandi (*password*) yang wajib diisi oleh pengguna.

Tabel 4.36 Perancangan Antar Muka Halaman konfirmasi Pendaftaran

Perancangan Antarmuka



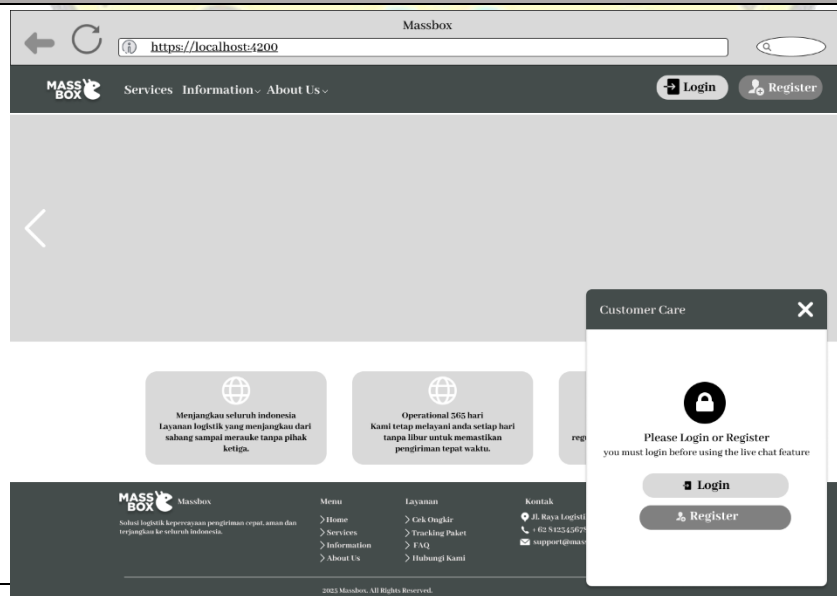
Gambar 4.17 Perancangan Antar Muka Halaman konfirmasi Pendaftaran

Deskripsi

Halaman ini merupakan rancangan tampilan konfirmasi pendaftaran yang menampilkan informasi bahwa akun pengguna telah berhasil dibuat, serta memberikan pemberitahuan bahwa pengguna dapat melanjutkan proses masuk (*login*) menggunakan kredensial yang telah didaftarkan.

Tabel 4.37 Perancangan Antar Muka Fitur Live Chat Sebelum Customer Melakukan Login

Perancangan Antarmuka

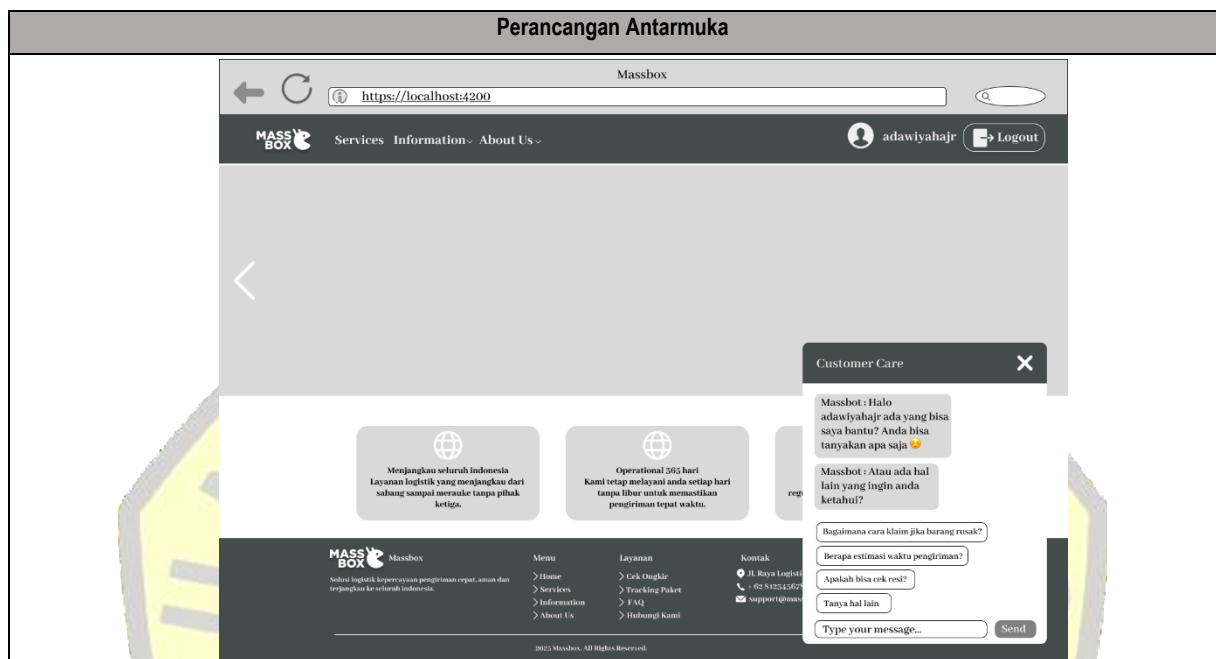


Gambar 4.18 Perancangan Antar Muka Fitur Live Chat Sebelum Customer Melakukan Login

Deskripsi

Halaman ini merupakan rancangan antarmuka awal yang ditampilkan kepada pengguna sebelum mengakses fitur *live chat*, yang berfungsi untuk menyampaikan informasi bahwa pengguna diwajibkan melakukan proses autentikasi (*login*) terlebih dahulu sebelum dapat menggunakan layanan tersebut.

Tabel 4.38 Perancangan Antar Muka Halaman Jendela Chat

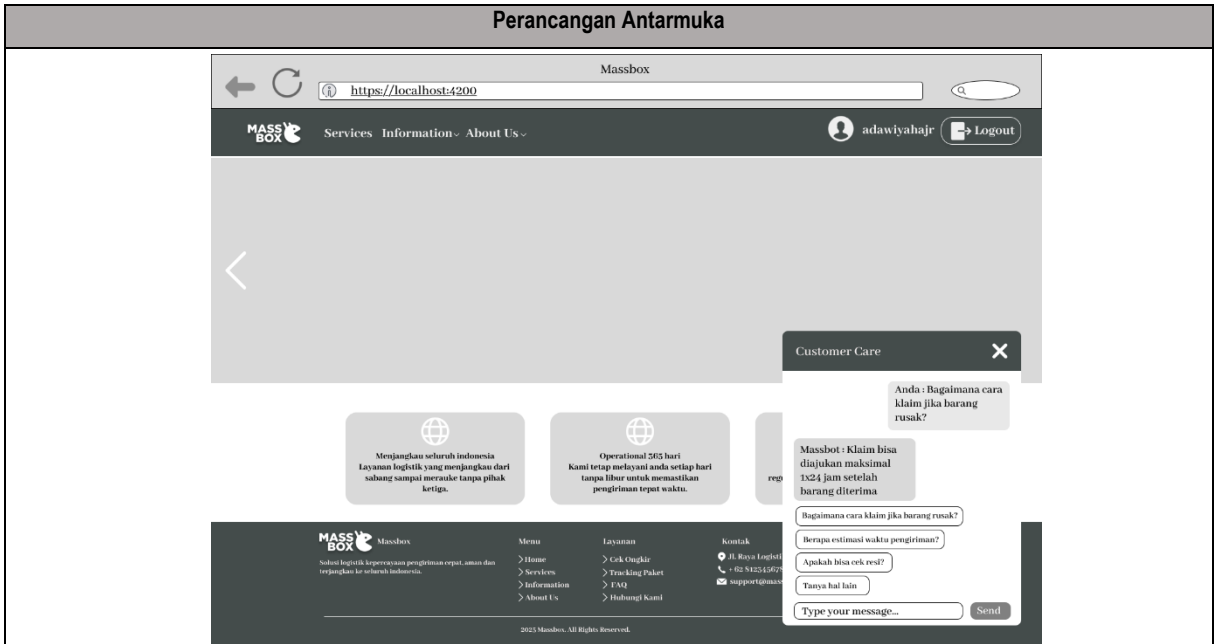


Gambar 4.19 Perancangan Antar Muka Halaman Jendela Chat

Deskripsi

Halaman ini merupakan rancangan tampilan yang ditampilkan setelah proses pendaftaran dan autentikasi (*login*) berhasil, di mana saat ikon *chat* diakses akan muncul jendela percakapan yang menampilkan sapaan dari MassBot serta beberapa tombol *quick question* yang berfungsi untuk memudahkan pengguna dalam mengajukan pertanyaan secara cepat.

Tabel 4.39 Perancangan Antar Muka Halaman Quick Question Chat

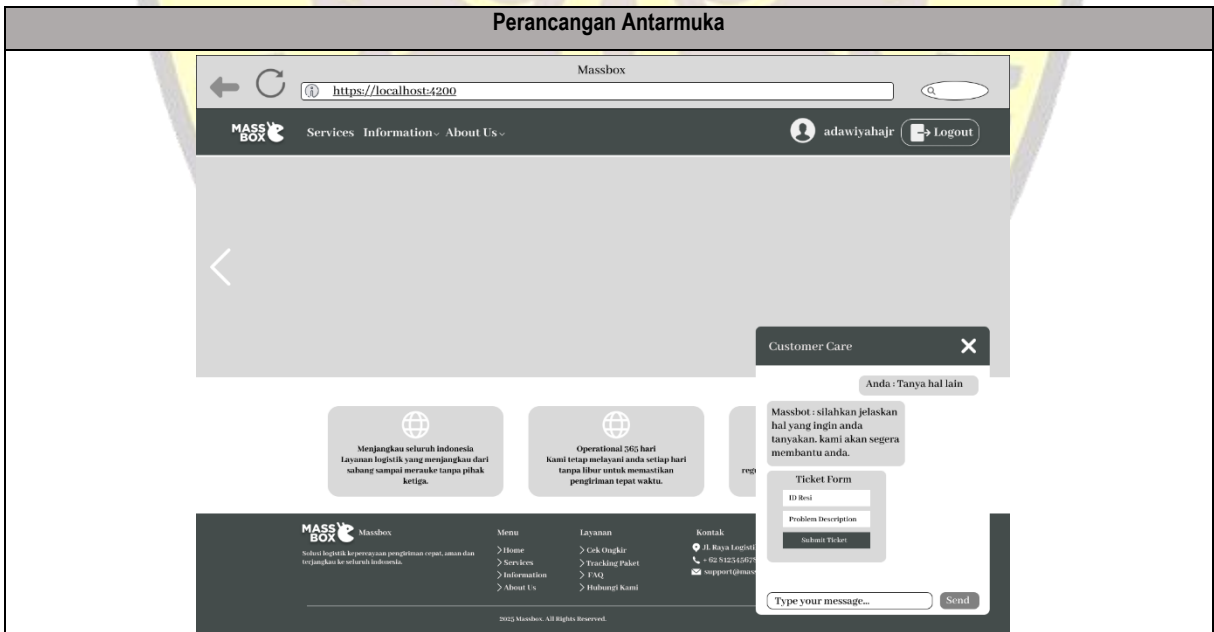


Gambar 4.20 Perancangan Antar Muka Halaman Quick Question Chat

Deskripsi

Halaman ini merupakan rancangan tampilan percakapan yang ditampilkan setelah pengguna menekan tombol *quick question*, di mana MassBot menampilkan pertanyaan yang dipilih beserta respons otomatis, sehingga interaksi antara pengguna dan sistem dapat berlangsung secara dinamis dan efisien.

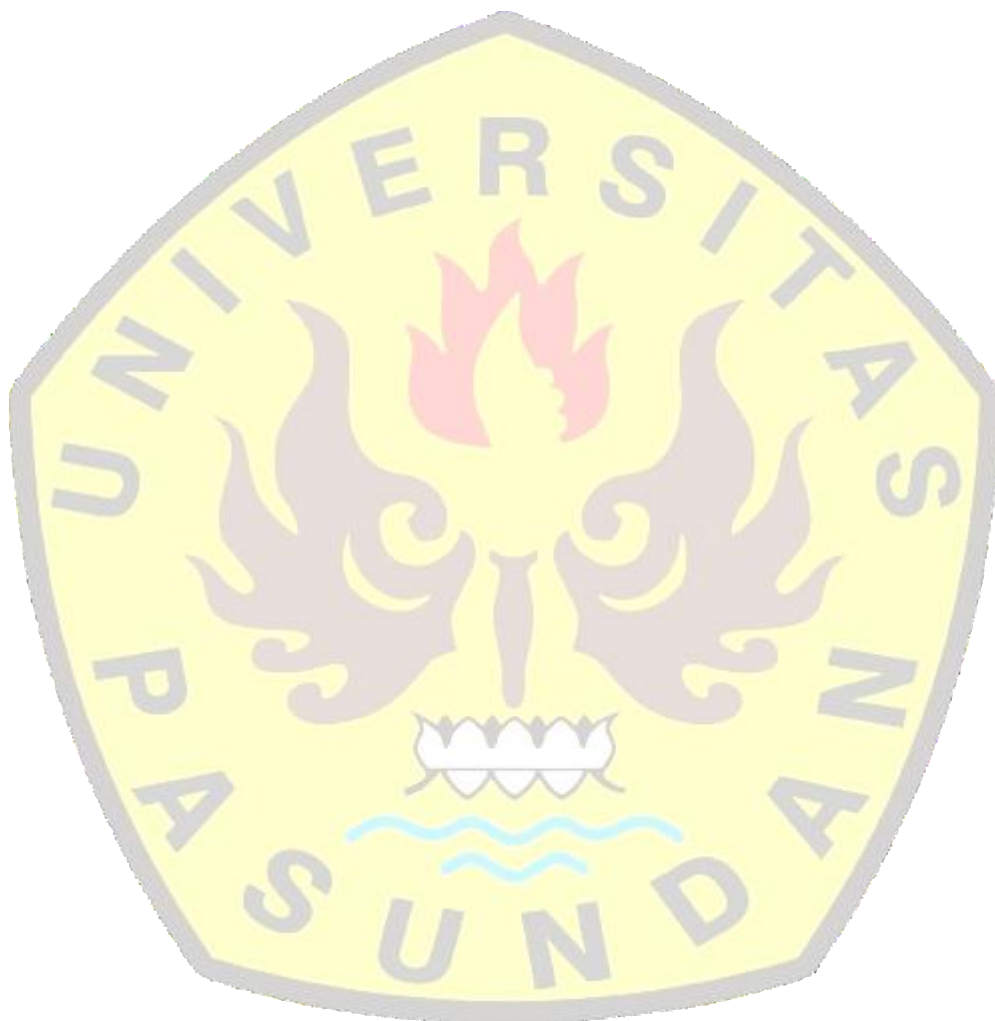
Tabel 4.40 Perancangan Antar Muka Halaman Pelaporan Kendala



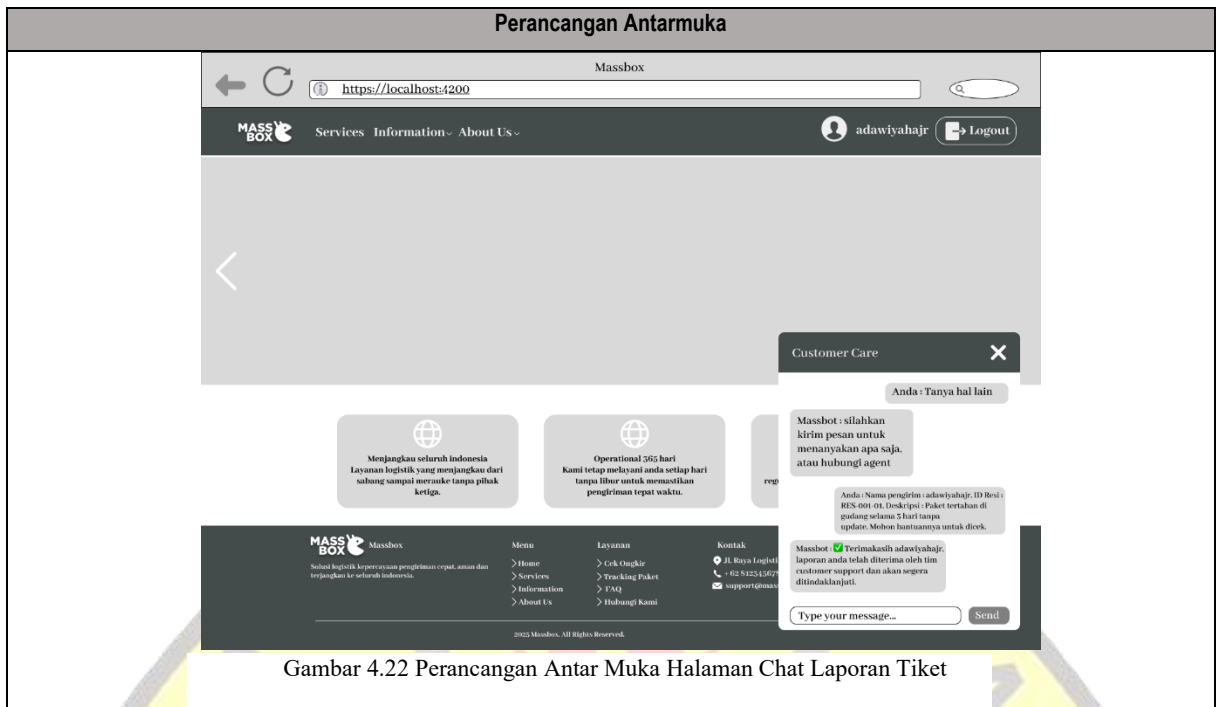
Gambar 4.21 Perancangan Antar Muka Halaman Pelaporan Kendala

Deskripsi

Halaman ini merupakan rancangan tampilan percakapan ketika pengguna memilih opsi "Tanya Hal Lain" pada menu *quick question*, di mana sistem menampilkan formulir percakapan yang memungkinkan pengguna untuk menyampaikan kendala atau permasalahan yang dialami secara langsung kepada layanan pelanggan.



Tabel 4.41 Perancangan Antar Muka Halaman Chat Laporan Tiket

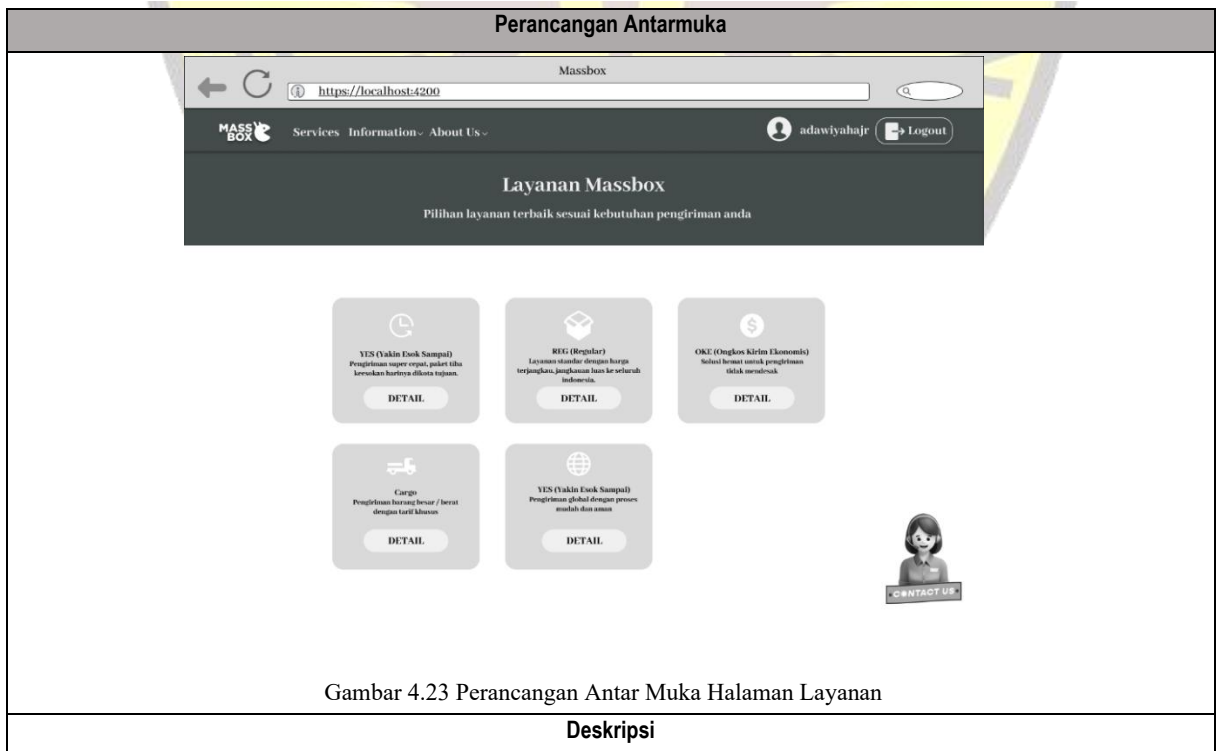


Gambar 4.22 Perancangan Antar Muka Halaman Chat Laporan Tiket

Deskripsi

Halaman ini merupakan rancangan tampilan percakapan setelah pengguna mengirimkan tiket laporan, di mana MassBot memberikan respons otomatis yang menyatakan bahwa laporan telah diterima dan akan segera diproses oleh pihak layanan pelanggan (*Customer service*).

Tabel 4.42 Perancangan Antar Muka Halaman Layanan

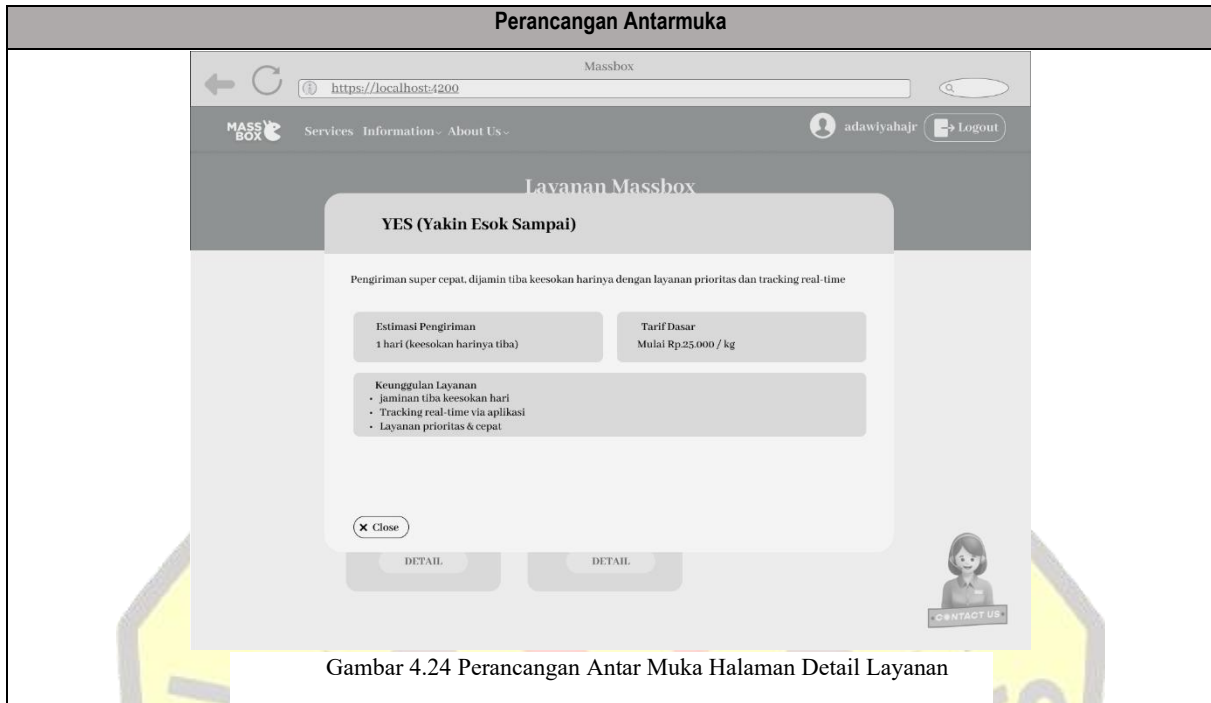


Gambar 4.23 Perancangan Antar Muka Halaman Layanan

Deskripsi

Halaman ini merupakan rancangan tampilan layanan Massbox yang menampilkan daftar layanan pengiriman dalam bentuk *card*. Pengguna dapat melihat informasi singkat dari setiap layanan yang tersedia serta menekan tombol "Detail" untuk menampilkan jendela *modal* yang memuat deskripsi layanan secara lengkap, estimasi waktu pengiriman, tarif dasar, serta keunggulan masing-masing layanan.

Tabel 4.43 Perancangan Antar Muka Halaman Detail Layanan

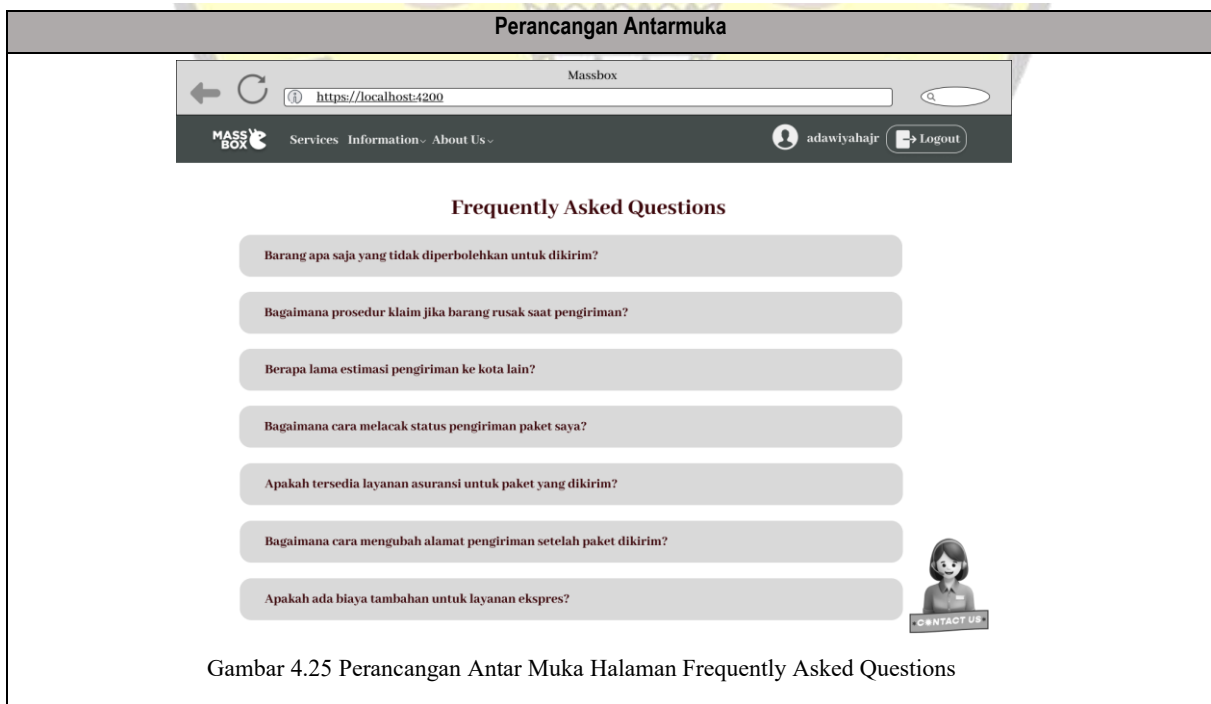


Gambar 4.24 Perancangan Antar Muka Halaman Detail Layanan

Deskripsi

Halaman ini merupakan rancangan tampilan detail layanan yang ditampilkan setelah pengguna menekan tombol "Detail", di mana jendela *modal* menyajikan informasi lengkap mengenai layanan, yang meliputi deskripsi, estimasi waktu pengiriman, tarif dasar, serta keunggulan layanan yang ditawarkan.

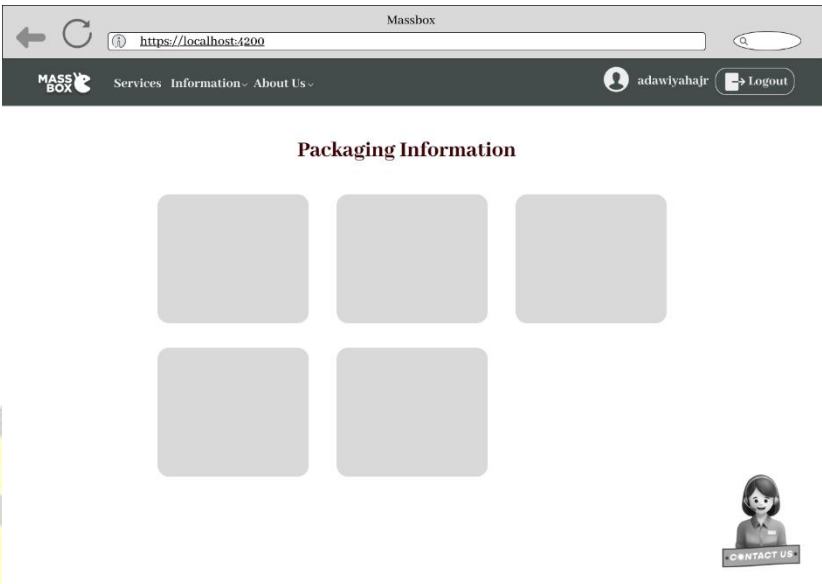
Tabel 4.44 Perancangan Antar Muka Halaman Frequently Asked Questions



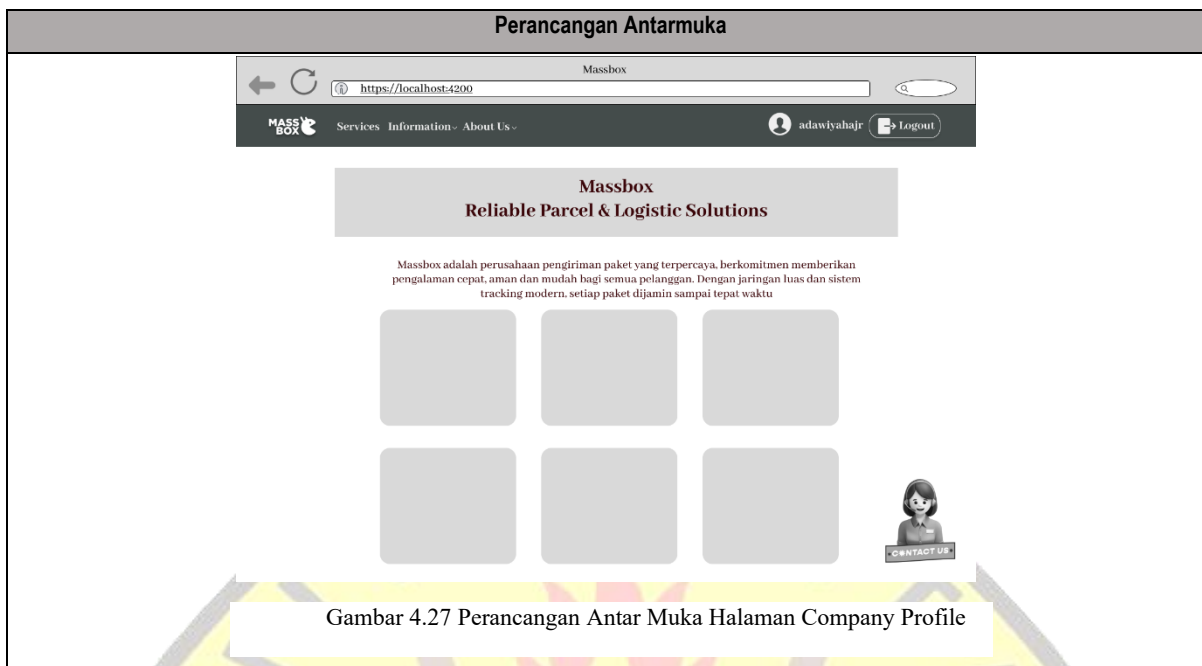
Gambar 4.25 Perancangan Antar Muka Halaman Frequently Asked Questions

Deskripsi
Halaman ini merupakan rancangan tampilan <i>Frequently Asked Questions</i> (FAQ) pada situs web Massbox yang menampilkan daftar pertanyaan umum dalam bentuk <i>accordion</i> . Pengguna dapat memilih setiap pertanyaan untuk menampilkan jawaban terkait secara interaktif.

Tabel 4.45 Perancangan Antar Muka Halaman Packaging Information

Perancangan Antarmuka
 <p>The screenshot shows a web browser window with the URL 'https://localhost:4200'. The page title is 'Massbox'. The navigation bar includes 'Services', 'Information', and 'About Us'. A user profile 'adawiyahjr' and a 'Logout' button are present. The main content area is titled 'Packaging Information' and displays five placeholder cards in a grid. A 'CONTACT US' button with a user icon is located in the bottom right corner.</p>
Gambar 4.26 Perancangan Antar Muka Halaman Packaging Information
Deskripsi
Halaman ini merupakan rancangan tampilan informasi <i>Packaging</i> pada Massbox yang menyajikan daftar jenis kemasan dalam bentuk <i>card</i> . Setiap <i>card</i> menampilkan informasi berupa nama kemasan, ukuran, berat maksimum, serta tips penggunaan yang bertujuan untuk membantu pengguna dalam memilih kemasan yang sesuai dengan kebutuhan pengiriman.

Tabel 4.46 Perancangan Antar Muka Halaman Company Profile

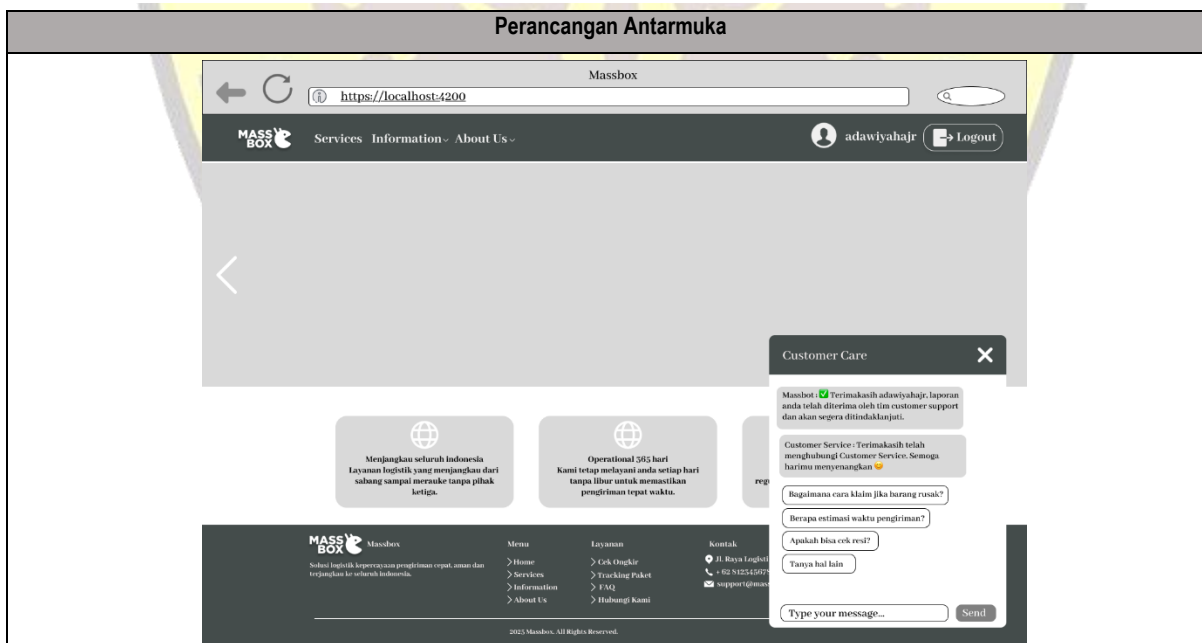


Gambar 4.27 Perancangan Antar Muka Halaman Company Profile

Deskripsi

Halaman ini merupakan rancangan tampilan *Company Profile* pada Massbox yang memuat deskripsi perusahaan serta layanan utama yang ditawarkan. Halaman ini menyajikan informasi mengenai komitmen Massbox dalam menyediakan layanan pengiriman paket yang cepat, aman, dan terpercaya, serta berbagai layanan unggulan yang dirancang untuk mendukung pengalaman pengiriman pelanggan.

Tabel 4.47 Perancangan Antarmuka Halaman Pesan Penutup Percakapan

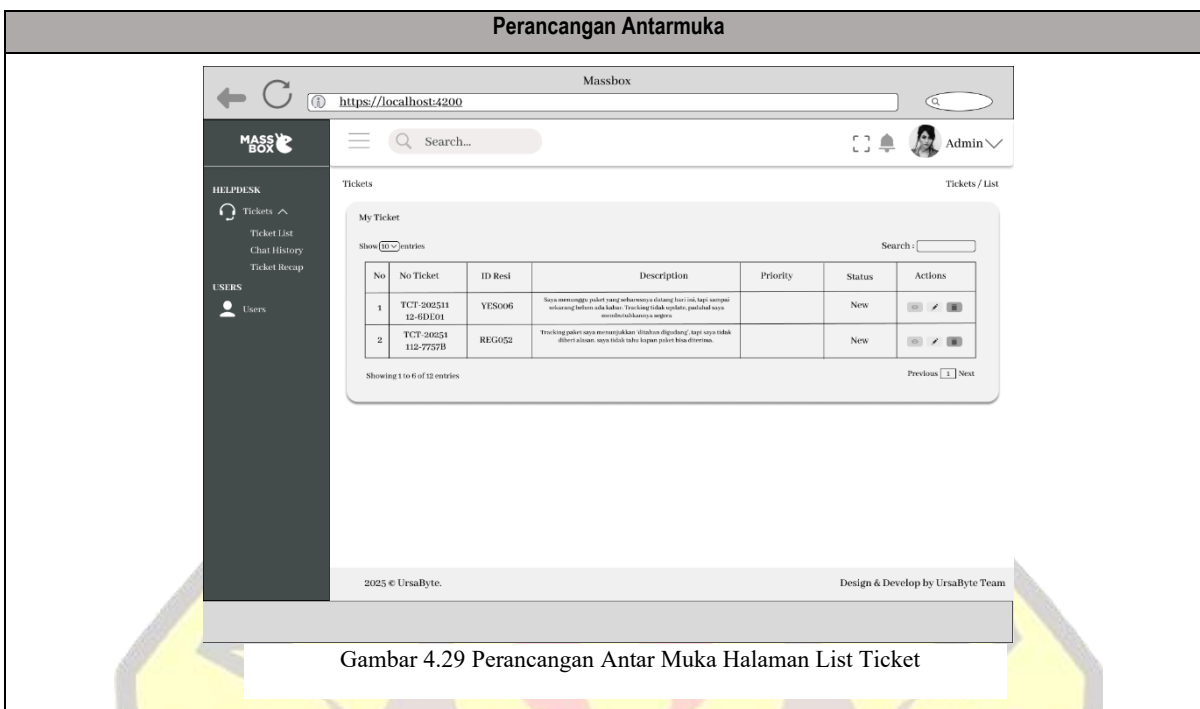


Gambar 4.28 Perancangan Antarmuka Halaman Pesan Penutup Percakapan

Deskripsi

Halaman ini merupakan rancangan tampilan percakapan setelah pihak layanan pelanggan (*Customer service*) mengakhiri interaksi dengan pengguna, di mana sistem menampilkan pesan penutup sebagai notifikasi akhir dari percakapan.

Tabel 4.48 Perancangan Antar Muka Halaman List Ticket

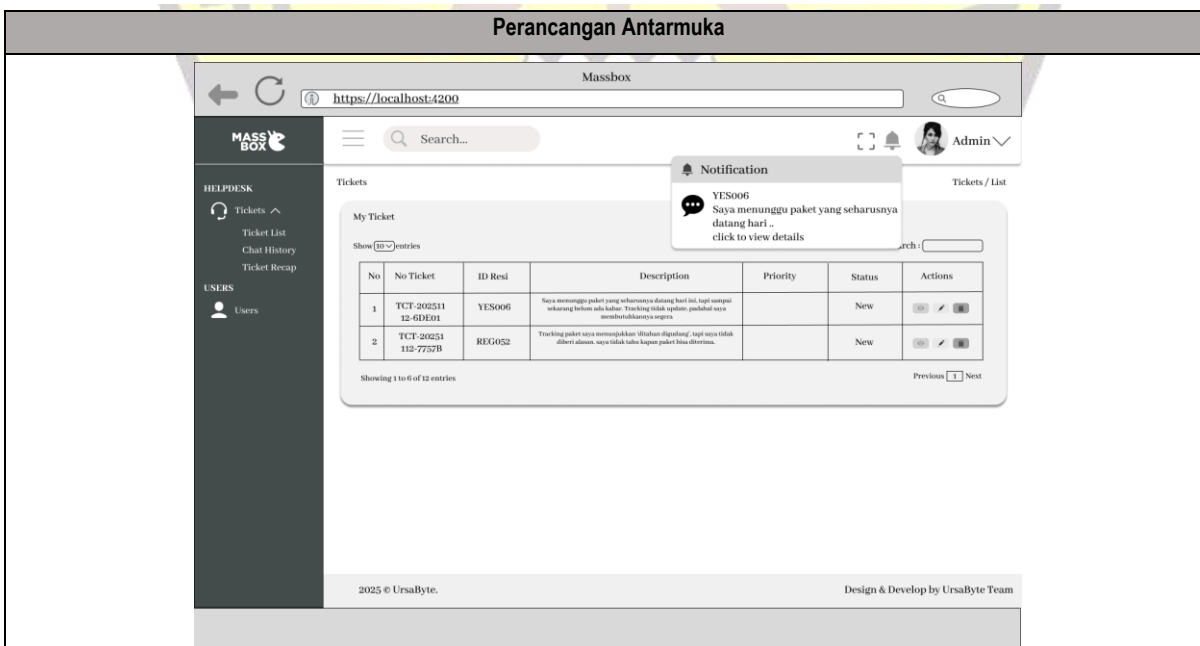


Gambar 4.29 Perancangan Antar Muka Halaman List Ticket

Deskripsi

Halaman ini merupakan rancangan tampilan daftar tiket pengguna yang menyajikan tabel berisi informasi tiket, meliputi nomor tiket, ID resi, deskripsi permasalahan, tingkat prioritas, dan status. Melalui halaman ini, admin dapat melihat detail tiket, melakukan pengeditan, atau menghapus tiket melalui tombol aksi yang tersedia pada setiap baris data.

Tabel 4.49 Perancangan Antar Muka Halaman Notifications

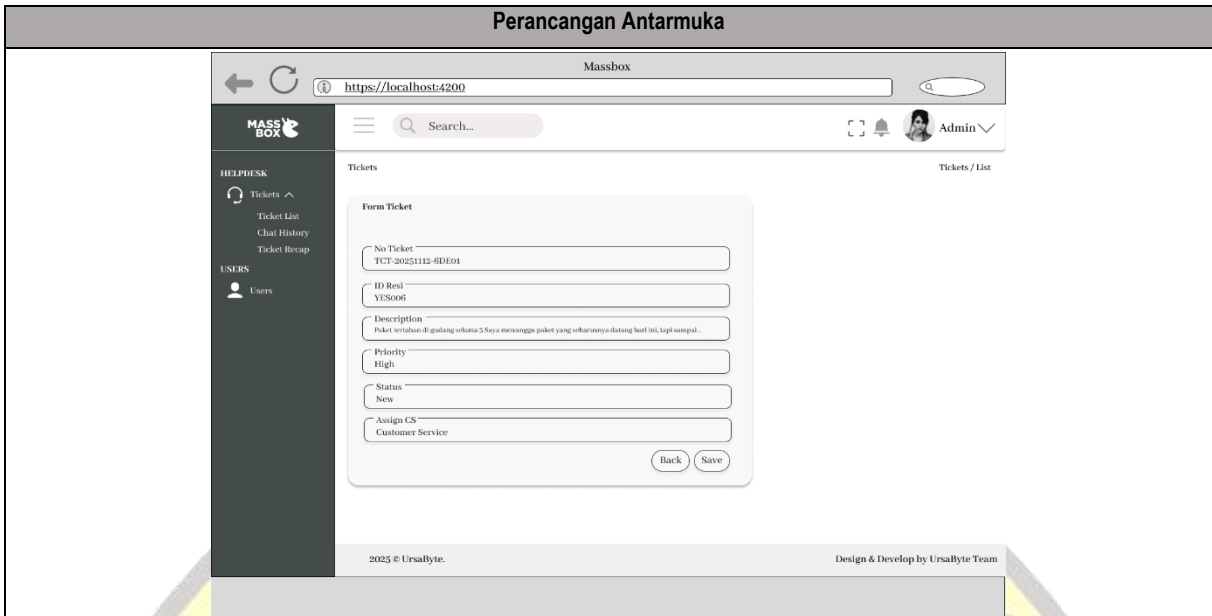


Gambar 4.30 Perancangan Antar Muka Halaman Notifications

Deskripsi

Halaman ini merupakan rancangan tampilan notifikasi yang ditampilkan setelah pelanggan mengirimkan kendala atau permasalahan, yang menunjukkan bahwa laporan telah berhasil diterima oleh sistem dan akan segera ditindaklanjuti.

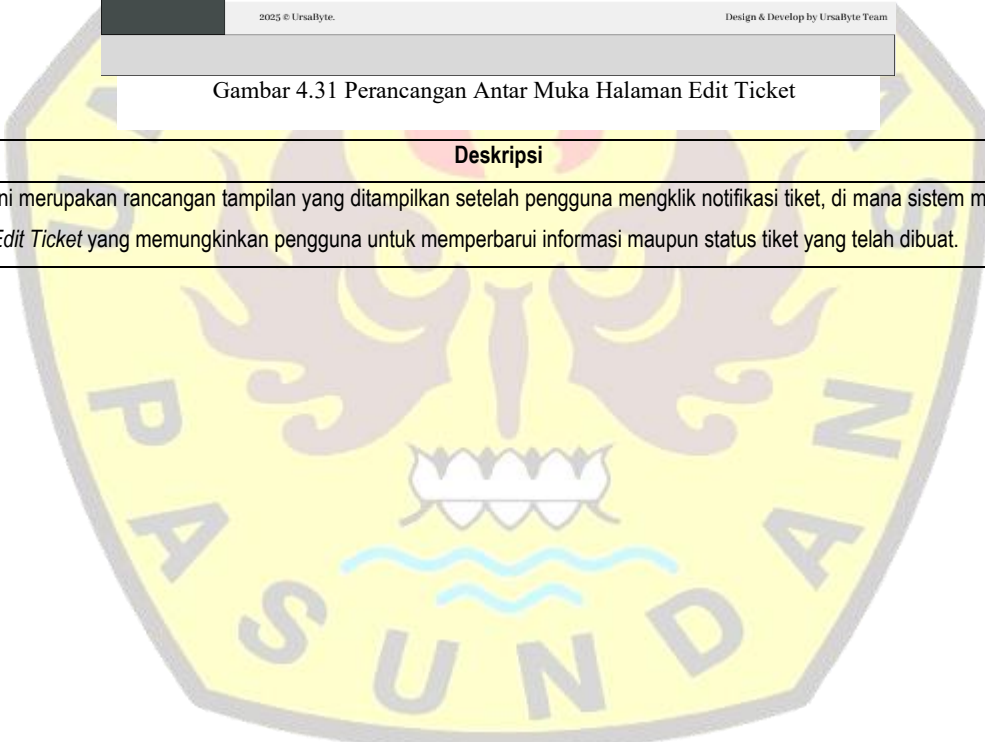
Tabel 4.50 Perancangan Antar Muka Halaman Edit Ticket



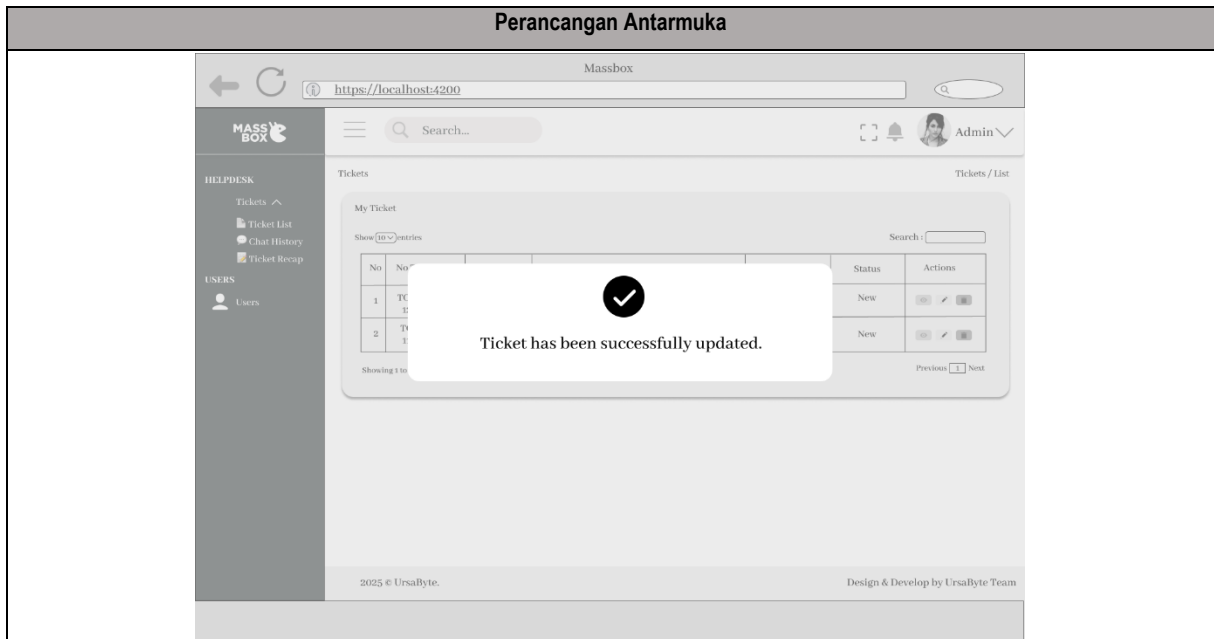
Gambar 4.31 Perancangan Antar Muka Halaman Edit Ticket

Deskripsi

Halaman ini merupakan rancangan tampilan yang ditampilkan setelah pengguna mengklik notifikasi tiket, di mana sistem menampilkan halaman *Edit Ticket* yang memungkinkan pengguna untuk memperbarui informasi maupun status tiket yang telah dibuat.



Tabel 4.51 Perancangan Antar Muka Halaman Konfirmasi Update Tiket

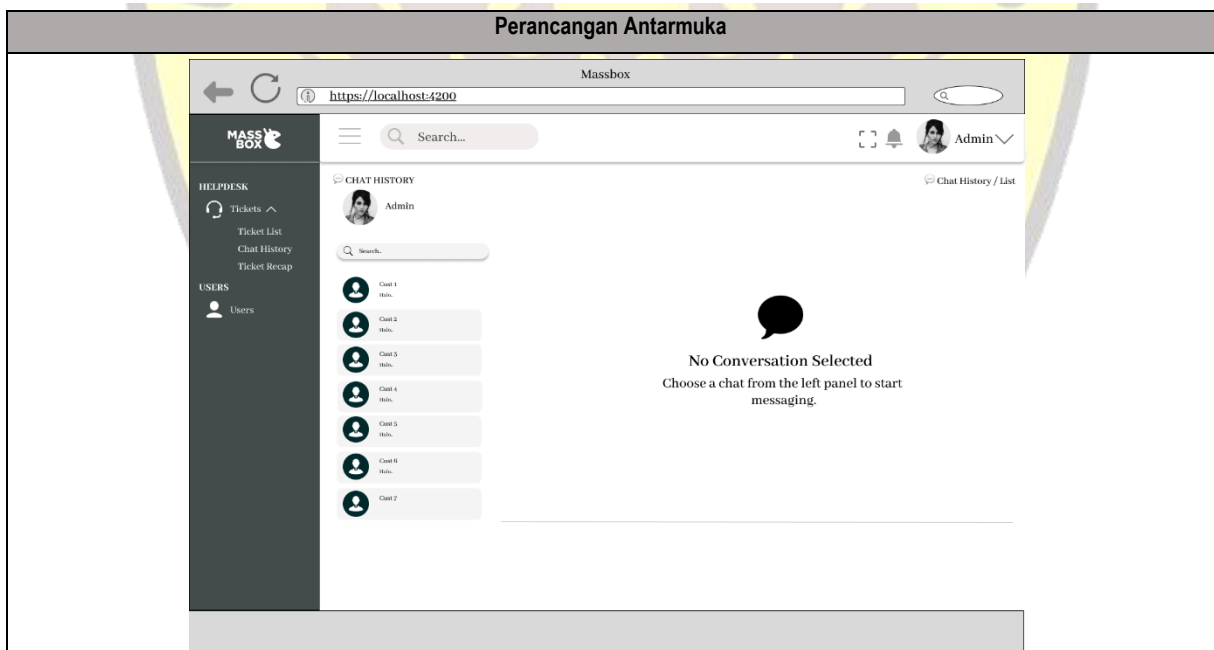


Gambar 4.32 Perancangan Antar Muka Halaman Konfirmasi Update Tiket

Deskripsi

Halaman ini merupakan rancangan tampilan notifikasi konfirmasi yang ditampilkan setelah pengguna berhasil memperbarui status tiket, yang memberikan informasi bahwa perubahan tersebut telah berhasil disimpan.

Tabel 4.52 Perancangan Antar Muka Halaman Pemberitahuan Ketika Tidak Ada Percakapan yang Dipilih

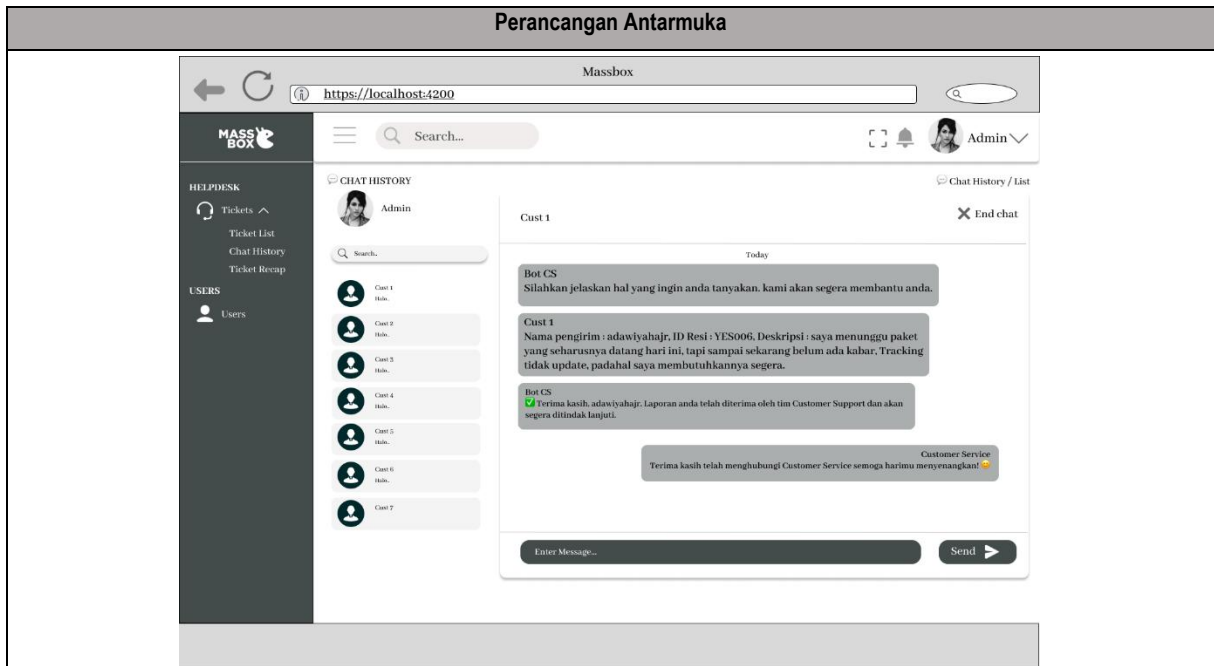


Gambar 4.33 Perancangan Antar Muka Halaman Pemberitahuan Ketika Tidak Ada Percakapan yang Dipilih

Deskripsi

Halaman ini merupakan rancangan antarmuka pemberitahuan yang ditampilkan ketika belum ada percakapan yang dipilih oleh pengguna, dengan tujuan memberikan arahan agar pengguna memilih percakapan sebelum memulai pengiriman pesan.

Tabel 4.53 Perancangan Antar Muka Halaman Chat History

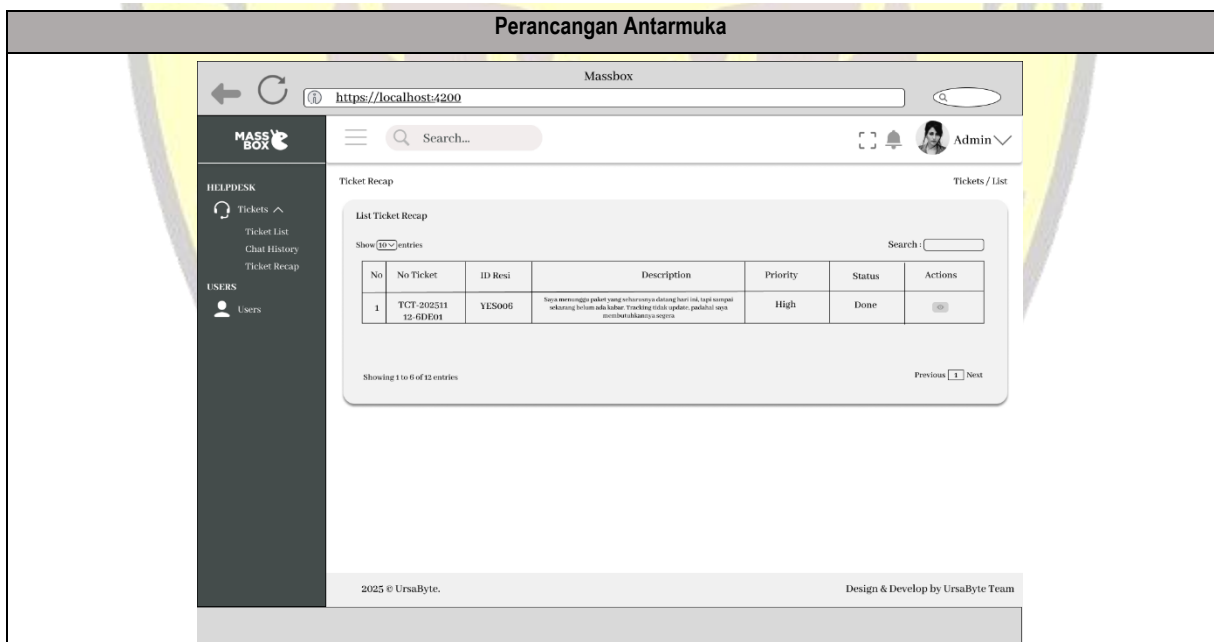


Gambar 4.34 Perancangan Antar Muka Halaman Chat History

Deskripsi

Halaman ini merupakan rancangan tampilan percakapan yang menyajikan riwayat pesan antara pihak layanan pelanggan dan pengirim tiket, yang berfungsi untuk memudahkan proses tindak lanjut serta menjadi referensi terhadap interaksi yang telah dilakukan sebelumnya.

Tabel 4.54 Perancangan Antar Muka Halaman Ticket Recap

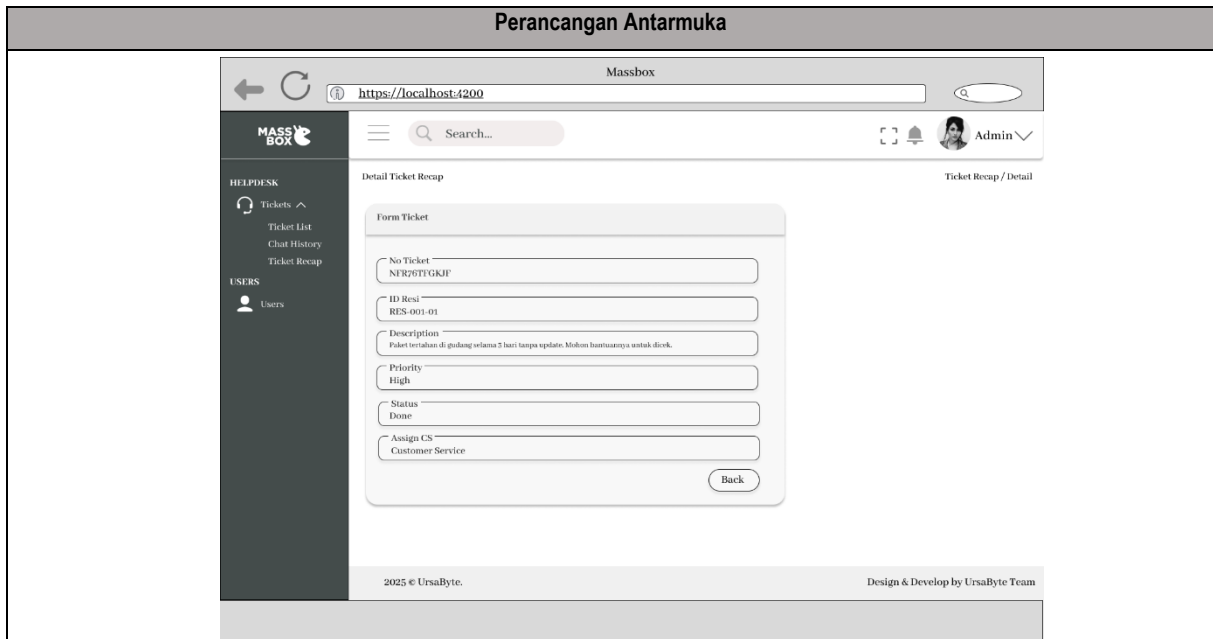


Gambar 4.35 Perancangan Antar Muka Halaman Ticket Recap

Deskripsi

Halaman ini merupakan rancangan tampilan rekap tiket yang menampilkan daftar tiket pengguna dengan status selesai. Halaman ini menyajikan informasi berupa nomor tiket, ID resi, deskripsi, prioritas, dan status, serta menyediakan tombol "View" yang memungkinkan admin meninjau detail setiap tiket.

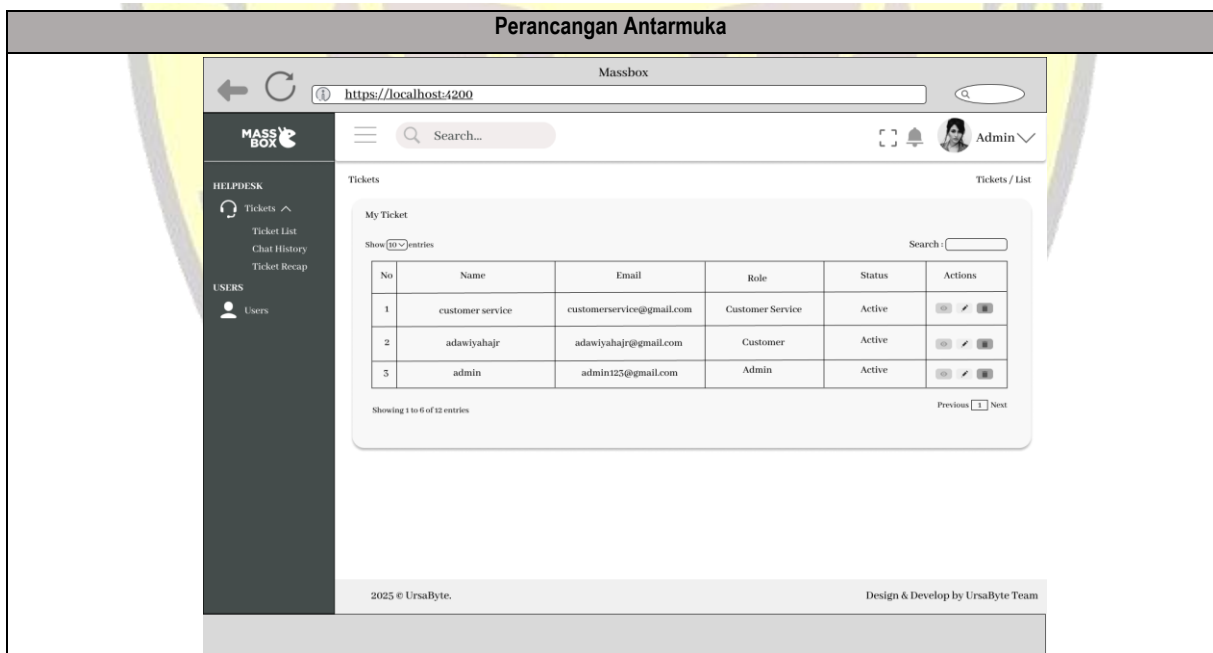
Tabel 4.55 Perancangan Antar Muka Halaman Detail Ticket Recap



Deskripsi

Halaman ini merupakan rancangan tampilan detail tiket yang ditampilkan setelah admin menekan tombol “View” pada daftar tiket, di mana sistem menyajikan informasi lengkap terkait tiket tersebut untuk keperluan peninjauan lebih lanjut.

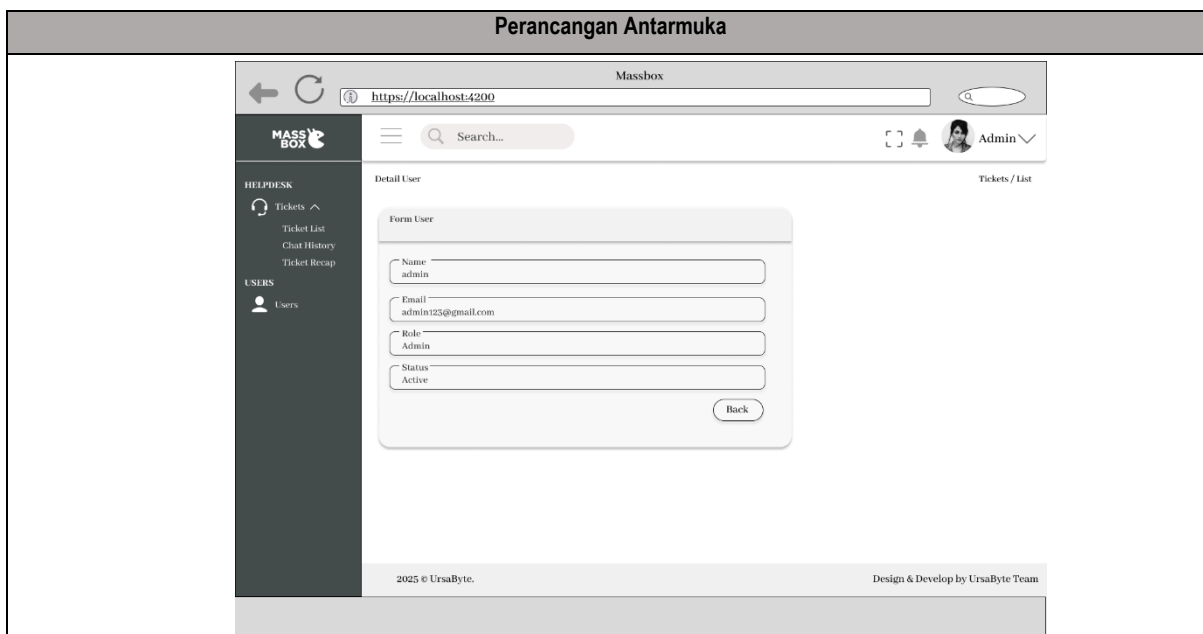
Tabel 4.56 Perancangan Antar Muka Halaman Users



Deskripsi

Halaman ini merupakan rancangan tampilan daftar pengguna yang menyajikan tabel berisi informasi setiap pengguna, meliputi nama, email, role, dan status. Admin dapat mengelola data pengguna dengan melihat detail, melakukan pengeditan, atau menghapus data melalui tombol aksi yang tersedia pada setiap baris.

Tabel 4.57 Perancangan Antar Muka Halaman Detail Users

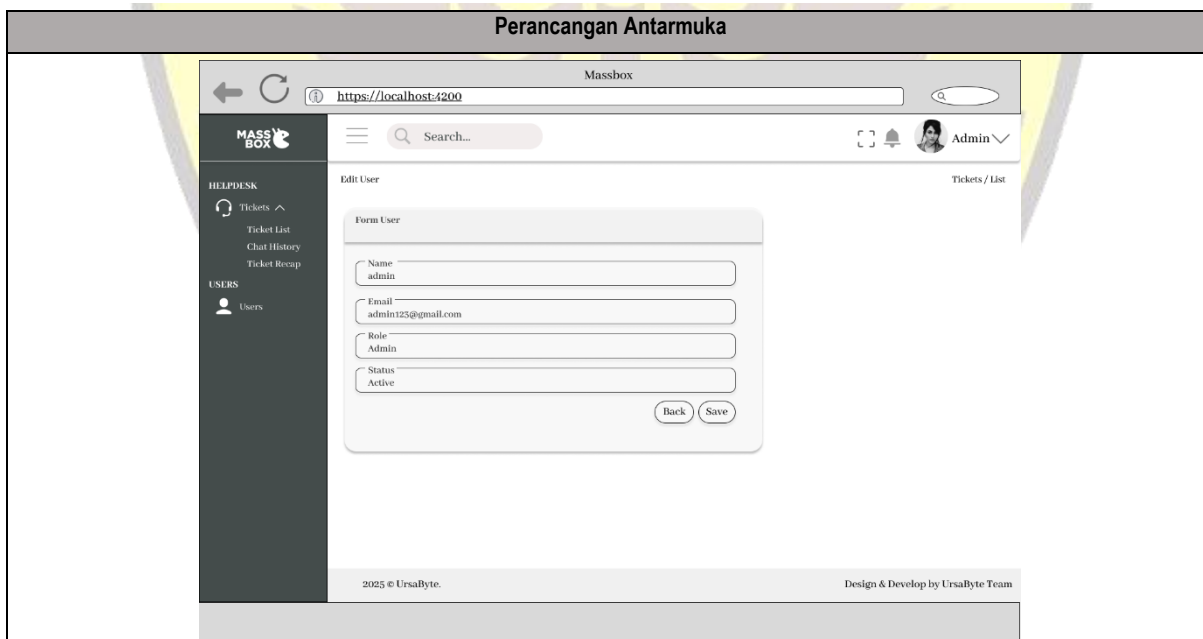


Gambar 4.38 Perancangan Antar Muka Halaman Detail Users

Deskripsi

Halaman ini merupakan rancangan tampilan detail pengguna yang berisi kolom input untuk nama, alamat surel, peran, dan status pengguna. Setiap kolom dilengkapi dengan validasi wajib, serta terdapat tombol untuk menyimpan perubahan data atau kembali ke halaman daftar pengguna.

Tabel 4.58 Perancangan Antar Muka Halaman Edit Users

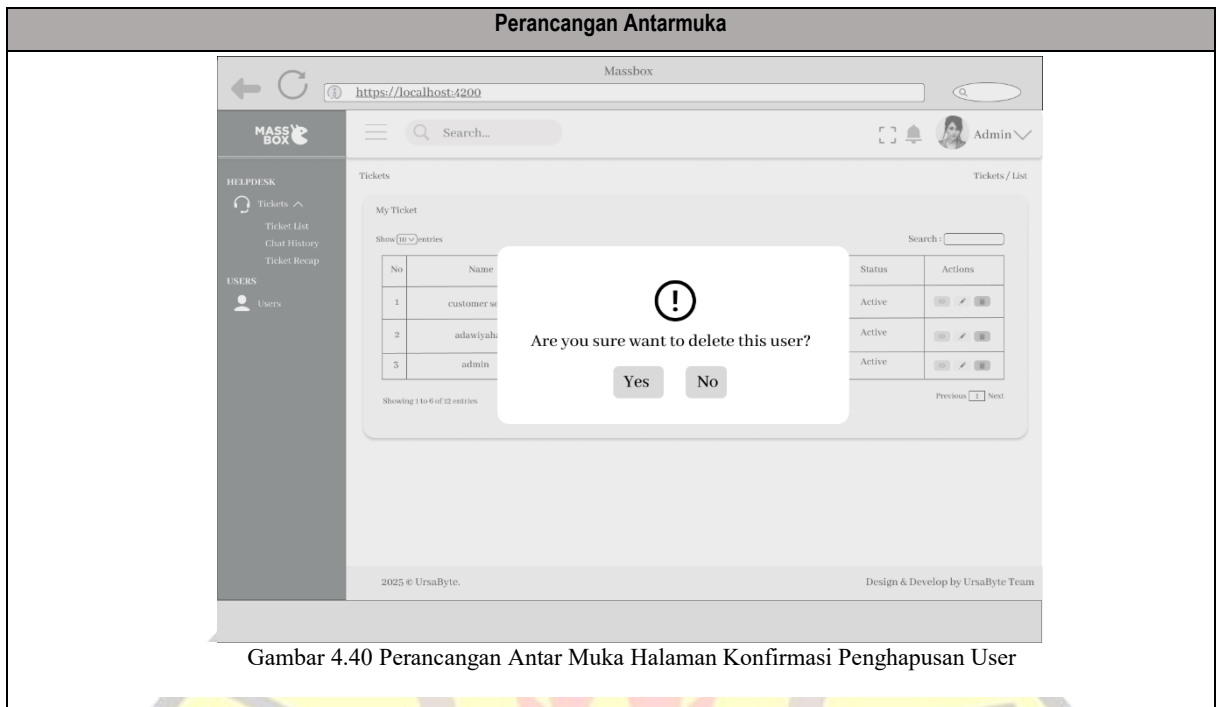


Gambar 4.39 Perancangan Antar Muka Halaman Edit Users

Deskripsi

Halaman ini merupakan rancangan tampilan pengeditan pengguna, di mana admin dapat memperbaiki informasi pengguna dengan validasi untuk memastikan data lengkap dan benar sebelum disimpan.

Tabel 4.59 Perancangan Antar Muka Halaman Konfirmasi Penghapusan User

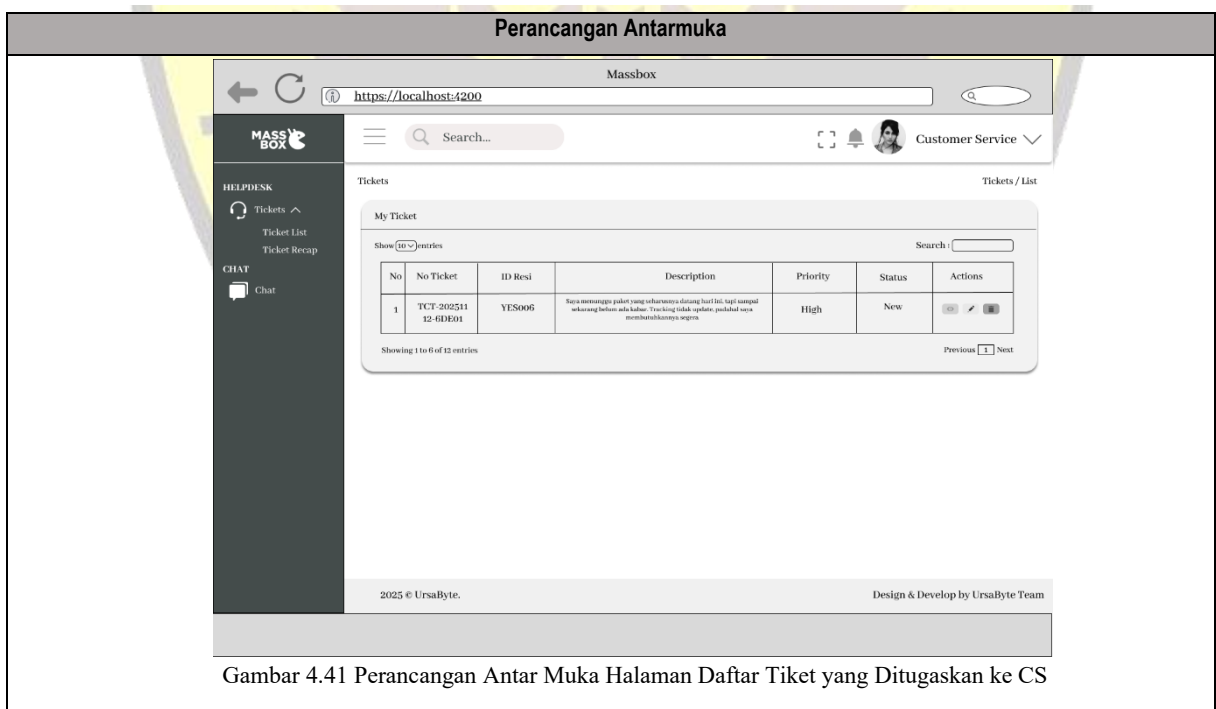


Gambar 4.40 Perancangan Antar Muka Halaman Konfirmasi Penghapusan User

Deskripsi

Halaman ini merupakan rancangan tampilan notifikasi konfirmasi penghapusan pengguna, yang berfungsi untuk meminta persetujuan admin sebelum sistem melakukan penghapusan data pengguna secara permanen.

Tabel 4.60 Perancangan Antar Muka Halaman Daftar Tiket yang Ditugaskan ke CS

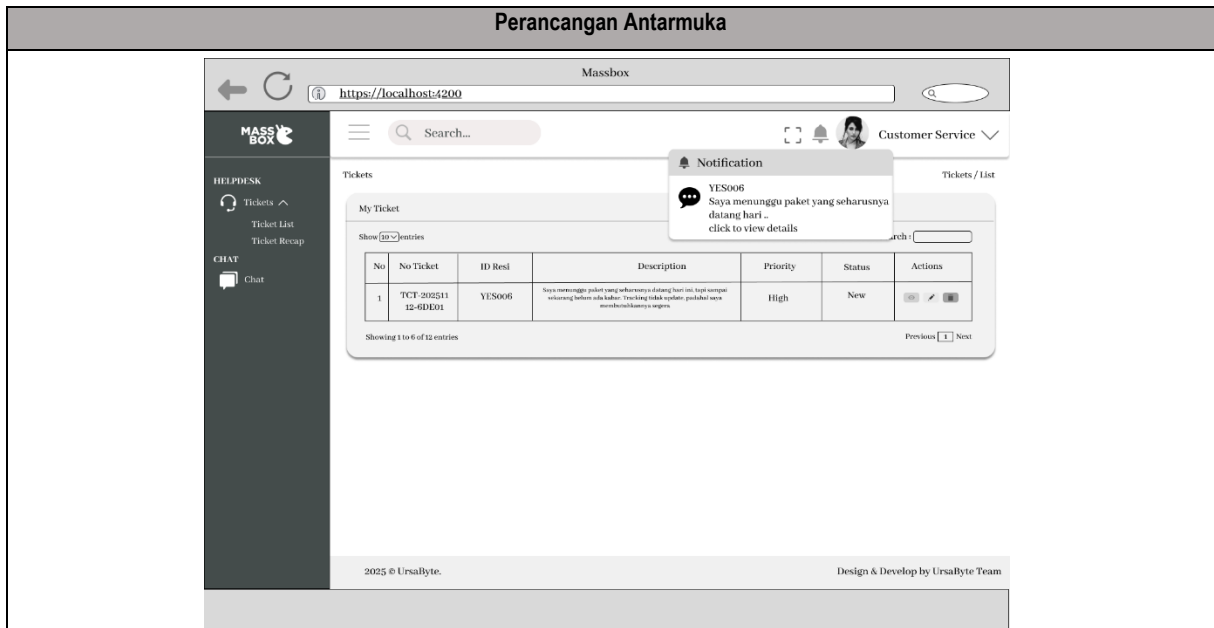


Gambar 4.41 Perancangan Antar Muka Halaman Daftar Tiket yang Ditugaskan ke CS

Deskripsi

Halaman ini merupakan rancangan tampilan daftar ticket yang telah diassign ke *Customer service*, yang menampilkan informasi ticket seperti nomor ticket, ID resi, deskripsi, prioritas, dan status. *Customer service* dapat memulai percakapan dengan pelanggan melalui tombol *Chat*, melihat detail ticket melalui tombol *View*, atau memperbarui ticket menggunakan tombol *Edit*.

Tabel 4.61 Perancangan Antar Muka Halaman Notifications

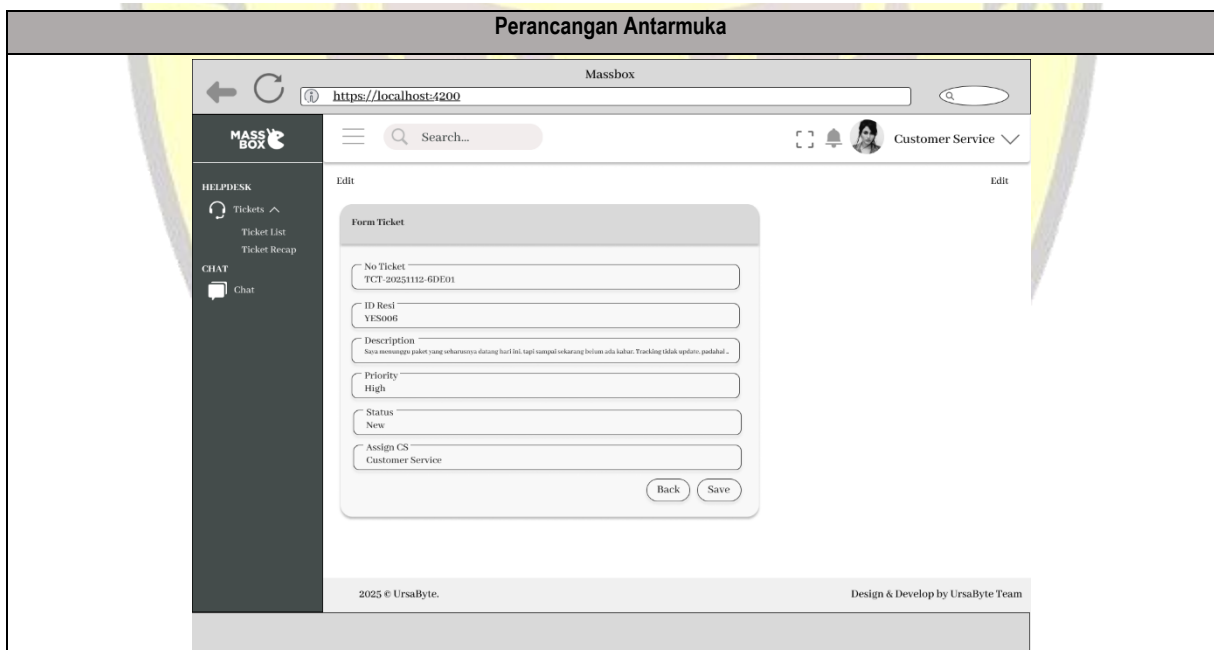


Gambar 4.42 Perancangan Antar Muka Halaman Notifications

Deskripsi

Halaman ini merupakan rancangan tampilan notifikasi untuk *Customer service*, yang berfungsi memberitahukan adanya ticket baru atau pembaruan ticket yang memerlukan perhatian, sehingga *Customer service* dapat segera menindaklanjuti dengan cepat dan tepat.

Tabel 4.62 Perancangan Antar Muka Halaman Edit Ticket

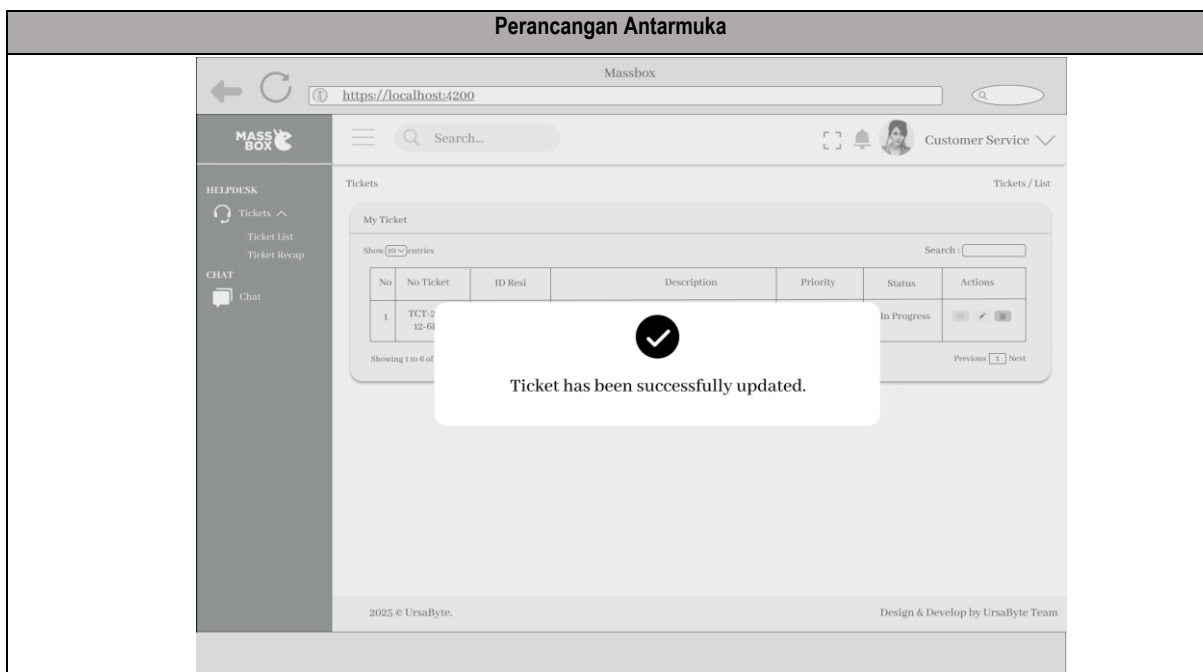


Gambar 4.43 Perancangan Antar Muka Halaman Edit Ticket

Deskripsi

Halaman ini merupakan rancangan tampilan Edit Ticket, yang muncul setelah pengguna menekan notifikasi ticket. Pada halaman ini, sistem menampilkan formulir untuk memperbarui informasi atau status ticket yang telah dibuat, guna memastikan data ticket tetap akurat dan terkini.

Tabel 4.63 Perancangan Antar Muka Halaman Konfirmasi Update Ticket

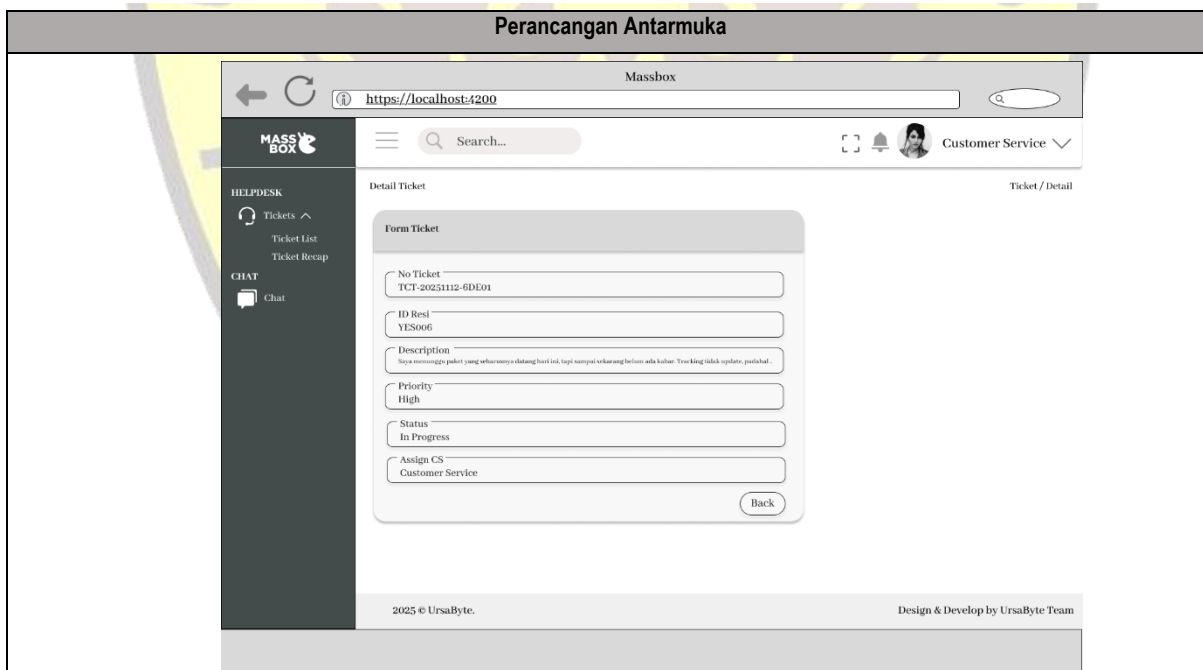


Gambar 4.44 Perancangan Antar Muka Halaman Konfirmasi Update Ticket

Deskripsi

Halaman ini merupakan rancangan tampilan notifikasi konfirmasi pembaruan ticket, yang muncul setelah pengguna berhasil memperbarui status ticket. Notifikasi ini berfungsi untuk memberikan umpan balik bahwa perubahan data telah berhasil disimpan oleh sistem.

Tabel 4.64 Perancangan Antar Muka Halaman Detail Tickets

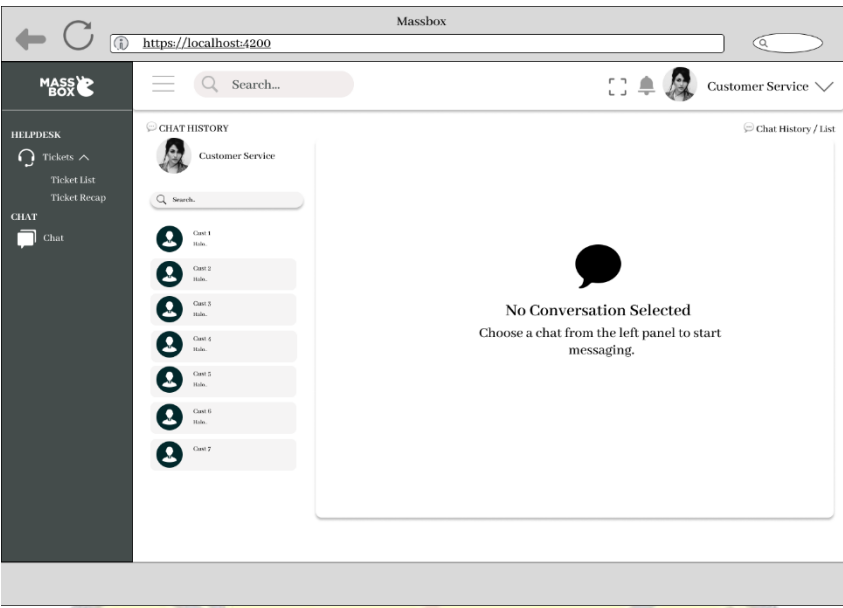


Gambar 4.45 Perancangan Antar Muka Halaman Detail Tickets

Deskripsi

Halaman ini merupakan rancangan tampilan detail ticket, yang menampilkan informasi lengkap mengenai ticket tertentu untuk memudahkan *Customer service* dalam meninjau dan menindaklanjuti laporan pengguna.

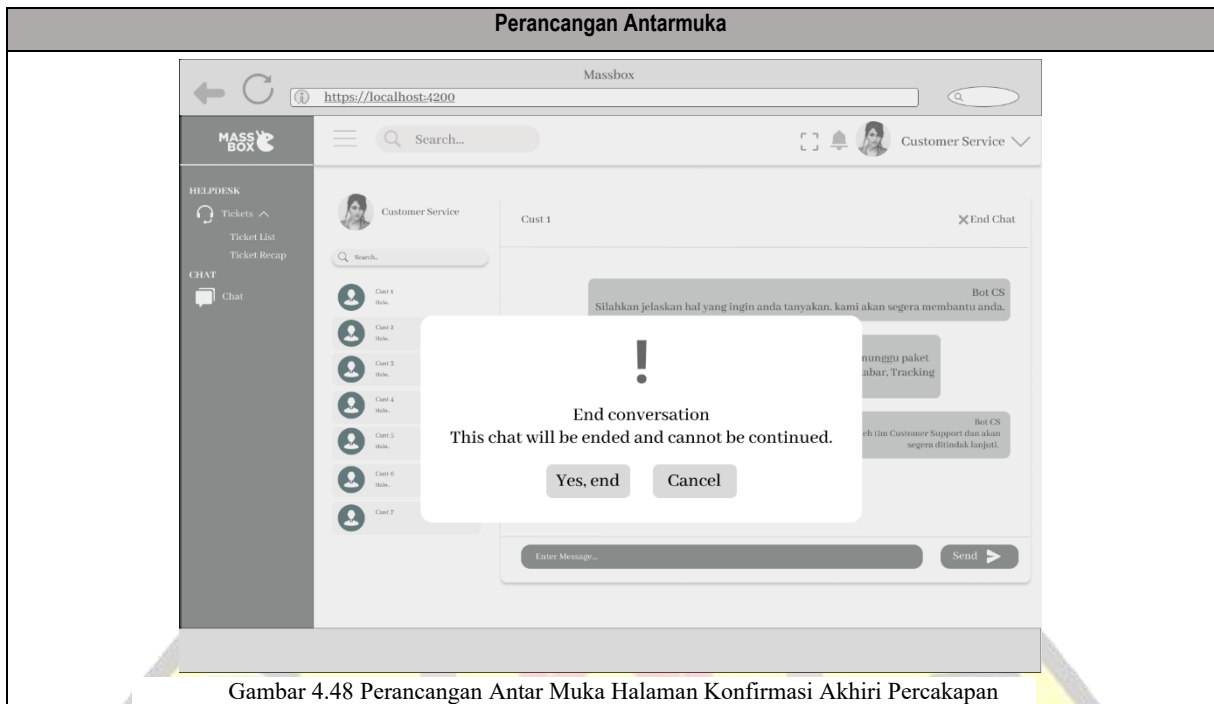
Tabel 4.65 Perancangan Antar Muka Pemberitahuan Ketika Tidak Ada Percakapan yang Dipilih

Perancangan Antarmuka	
	
Gambar 4.46 Perancangan Antar Muka Pemberitahuan Ketika Tidak	
Deskripsi	
Halaman ini merupakan rancangan antarmuka pemberitahuan yang ditampilkan ketika belum ada percakapan yang dipilih oleh pengguna, dengan tujuan memberikan arahan agar pengguna memilih percakapan sebelum memulai pengiriman pesan.	

Tabel 4.66 Perancangan Antar Muka Halaman Percakapan

Perancangan Antarmuka	
	
Gambar 4.47 Perancangan Antar Muka Halaman Percakapan	
Deskripsi	
Halaman ini merupakan rancangan tampilan percakapan ticket, yang menampilkan riwayat komunikasi antara <i>Customer service</i> dan pelanggan setelah <i>Customer service</i> menekan ikon <i>Chat</i> pada daftar ticket. Tampilan ini memudahkan <i>Customer service</i> dalam meninjau konteks percakapan sebelumnya serta melanjutkan komunikasi secara langsung dan responsif.	

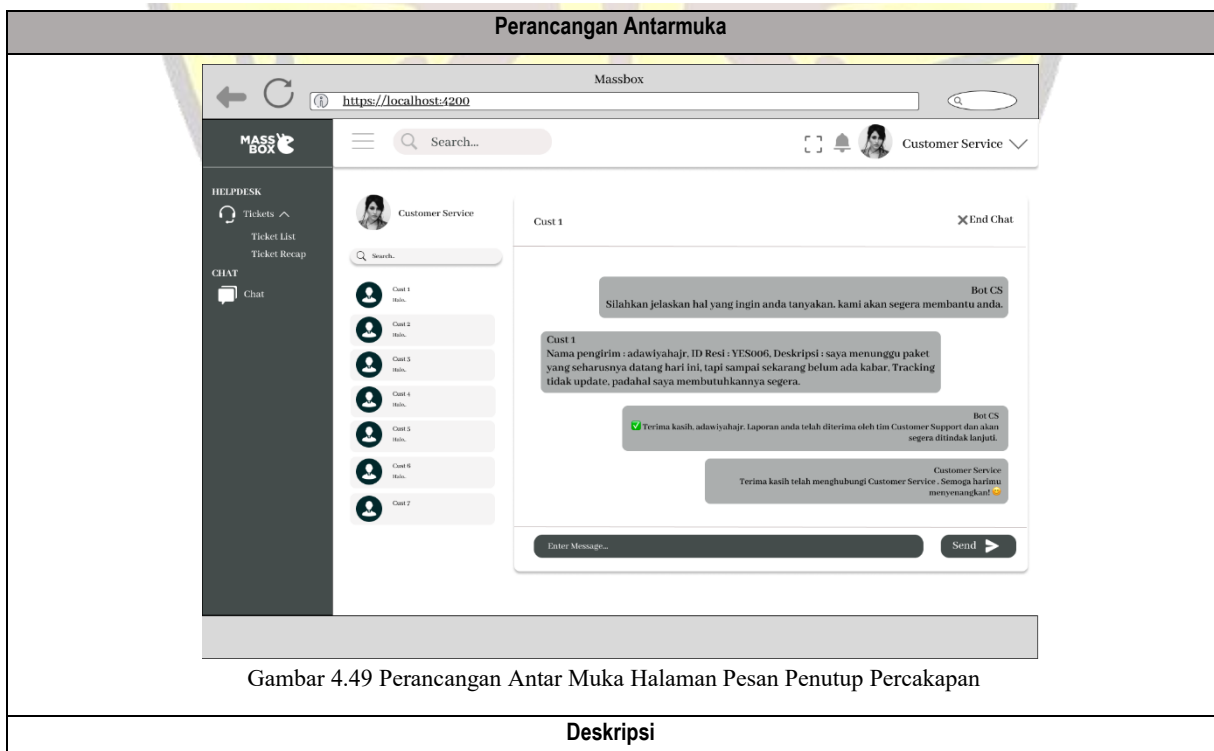
Tabel 4.67 Perancangan Antar Muka Halaman Konfirmasi Akhir Percakapan



Deskripsi

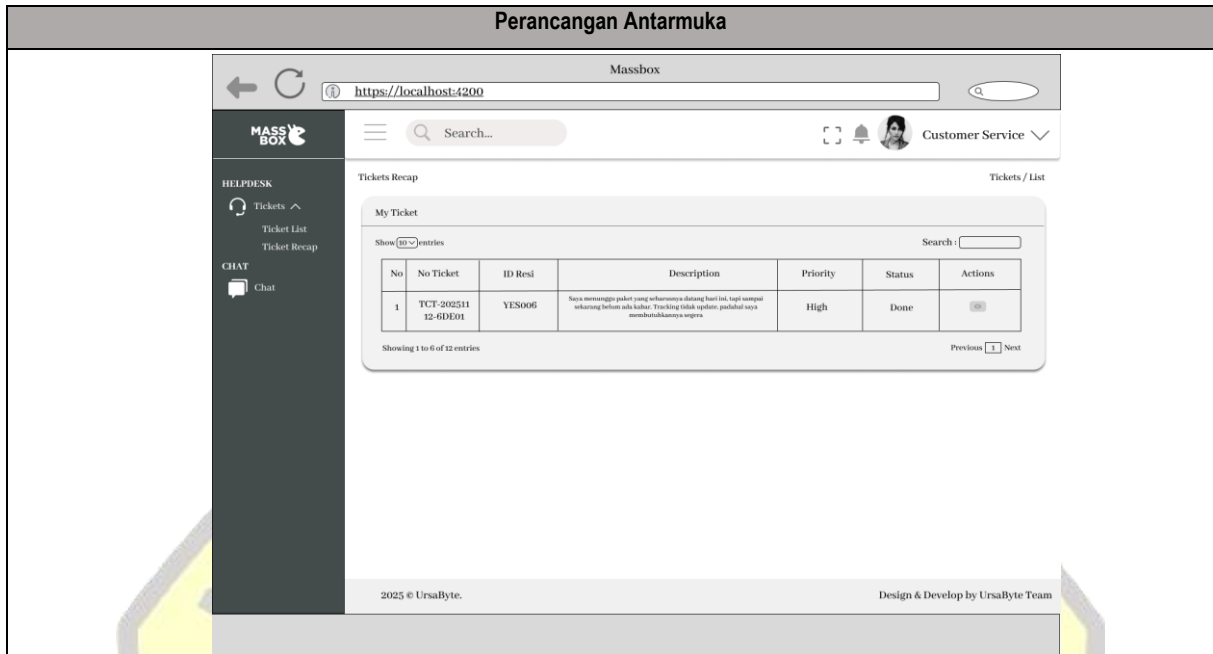
Halaman ini merupakan rancangan tampilan notifikasi konfirmasi akhir percakapan, yang muncul ketika *Customer service* menekan tombol 'End Chat' pada halaman percakapan. Notifikasi ini berfungsi untuk memastikan bahwa *Customer service* benar-benar ingin mengakhiri percakapan secara permanen sebelum tindakan tersebut dikonfirmasi.

Tabel 4.68 Perancangan Antar Muka Halaman Pesan Penutup Percakapan



Halaman ini merupakan rancangan tampilan pesan penutup percakapan, yang muncul setelah *Customer service* berhasil mengakhiri sesi chat. Pesan ini berfungsi sebagai pemberitahuan akhir bahwa percakapan telah selesai dan sebagai bentuk ucapan terima kasih atas interaksi yang telah dilakukan.

Tabel 4.69 Perancangan Antar Muka Halaman List Ticket Recap

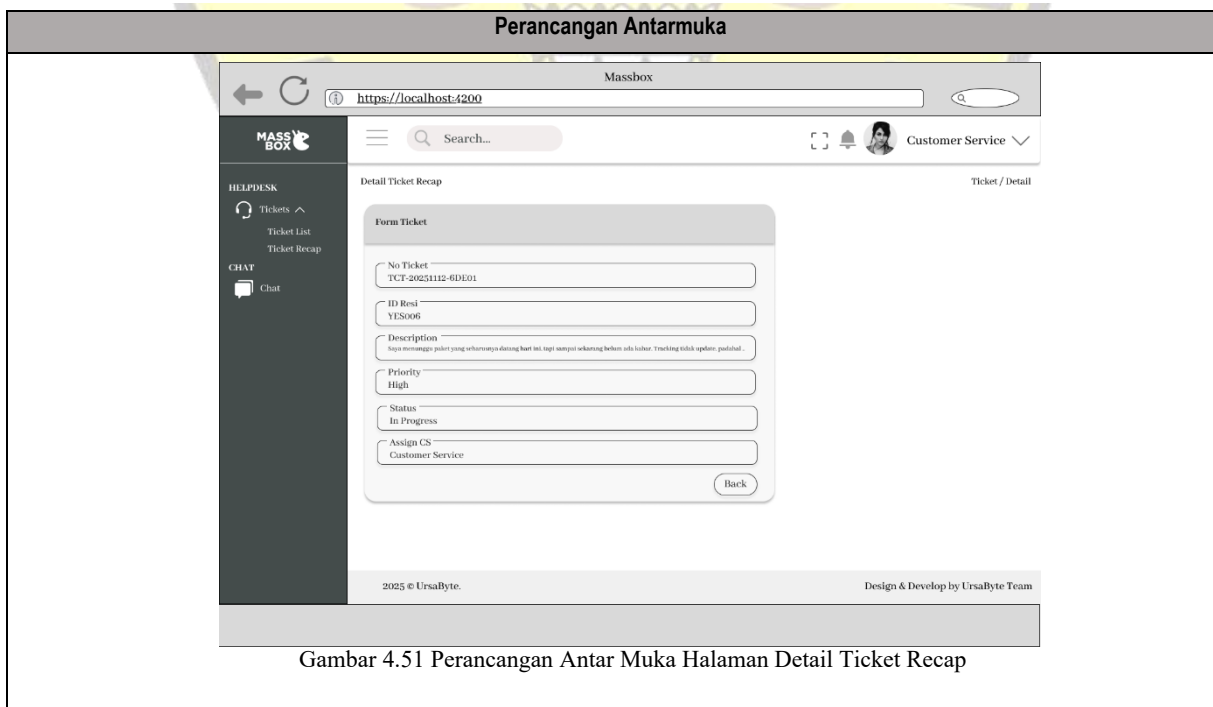


Gambar 4.50 Perancangan Antar Muka Halaman List Ticket Recap

Deskripsi

Halaman ini merupakan rancangan tampilan detail user, yang menampilkan informasi lengkap terkait data user, seperti nama, email, peran (role), dan status. Setiap kolom memiliki validasi wajib untuk memastikan kelengkapan data, serta menyediakan tombol untuk menyimpan perubahan atau kembali ke daftar pengguna.

Tabel 4.70 Perancangan Antar Muka Halaman Detail Ticket Recap



Gambar 4.51 Perancangan Antar Muka Halaman Detail Ticket Recap

Deskripsi

Halaman ini merupakan rancangan tampilan detail ticket recap, yang muncul setelah *Customer service* menekan tombol 'View' pada daftar ticket. Pada halaman ini, sistem menampilkan informasi lengkap mengenai ticket, termasuk ID ticket, deskripsi kendala, prioritas, status, serta data pelanggan terkait.



BAB 5 IMPLEMENTASI DAN PENGUJIAN

5.1. Kontruksi Perangkat Lunak

Konstruksi perangkat lunak merujuk pada proses pembuatan perangkat lunak secara detail, yang mencakup penulisan kode, integrasi, dan debugging. Tahap ini dilakukan berdasarkan rancangan-rancangan yang telah disusun sebelumnya.

5.1.1 Kebutuhan Pembangunan Perangkat Lunak

Berikut ini adalah kebutuhan perangkat lunak dan kakas untuk pengembangan perangkat lunak yang akan dibangun :

- a. Spesifikasi perangkat untuk pembangunan

Tabel 5.1 Spesifikasi perangkat lunak untuk pembangunan

No	Nama Perangkat	Spesifikasi Perangkat untuk Pembangunan
1.	Sistem Operasi	Windows 11 Home Single Language 64-bit
2.	Processor	AMD Ryzen 7 7435HS (16 CPUs), ~3.1 GHz
3.	RAM	16 GB
4.	Ruang Penyimpanan	512 GB

- b. Perangkat lunak yang digunakan dalam pembangunan

Tabel 5.2 Perangkat Lunak yang digunakan dalam pembangunan

No	Nama Perangkat	Sumber	Deskripsi
1.	Draw.io	https://app.diagrams.net/	Software yang digunakan untuk membuat diagram.
2.	Visual studio code	https://code.visualstudio.com/	Software yang digunakan untuk penulisan kode perangkat lunak.
3.	Github	https://github.com/	Software yang digunakan untuk manajemen penyimpanan kode sumber dan melacak perubahan pada repositori.
4.	Figma	https://www.figma.com/	Software berbasis web yang digunakan untuk merancang antarmuka pengguna (UI) dan pengalaman pengguna (UX) secara kolaboratif dalam pengembangan aplikasi maupun website.

No	Nama Perangkat	Sumber	Deskripsi
5.	Postman	https://www.postman.com/	Software yang digunakan untuk melakukan testing, dokumentasi, dan manajemen API.

5.1.2 Pengkodean

Pada tahap ini dilakukan proses penerjemahan rancangan sistem ke dalam bentuk kode program. Aplikasi dibangun menggunakan Node.js sebagai platform backend untuk mengelola logika aplikasi dan layanan API. Pada sisi frontend, sistem dikembangkan menggunakan framework *Angular* dengan bahasa pemrograman TypeScript untuk membangun antarmuka pengguna. Sebagai basis data, sistem menggunakan *Firebase* Cloud Firestore yang mendukung pengelolaan data secara *Real-time*, serta *Firebase* Authentication sebagai mekanisme autentikasi pengguna.

5.1.2.1 Struktur Aplikasi Web

Pada pembangunan aplikasi ini diterapkan pendekatan arsitektur *Model View Controller* (MVC). Bagian Model berfungsi untuk merepresentasikan dan mengelola data sistem seperti data pengguna, tiket, dan percakapan. Bagian View diimplementasikan pada aplikasi frontend berbasis *Angular* yang bertanggung jawab menampilkan antarmuka pengguna. Sementara itu, bagian Controller diimplementasikan pada backend Node.js yang berperan sebagai penghubung antara Model dan View serta mengatur alur logika aplikasi.

5.1.2.2 Model Aplikasi Web

Model dalam arsitektur MVC berfungsi untuk mengelola data sistem yang berkomunikasi dengan basis data serta melakukan pemrosesan data yang diterima dari controller, seperti menyimpan, mengambil, dan menghapus data. Pada sistem ini, pengelolaan data dilakukan melalui model yang diimplementasikan menggunakan Node.js dan terintegrasi dengan *Firebase* Cloud Firestore sebagai basis data. Model-model tersebut digunakan untuk merepresentasikan data pengguna, tiket, dan percakapan, sehingga memudahkan pengelolaan data secara terstruktur. Daftar model yang digunakan pada sistem ini dapat dilihat pada Tabel 5.3 berikut ini

Tabel 5.3 Model Aplikasi Web

No	Nama Model	Deskripsi
1.	Chat.model.ts	Model ini digunakan untuk merepresentasikan data percakapan antara pelanggan dan <i>Customer service</i> . Di dalamnya terdapat model Chat yang menyimpan informasi identitas pelanggan, nomor resi, dan waktu pembuatan percakapan, serta model Message yang menyimpan informasi pengirim, penerima, isi pesan, dan waktu pengiriman pesan dalam proses live chat.

No	Nama Model	Deskripsi
2.	Ticket.model.ts	Model ini digunakan untuk merepresentasikan data tiket layanan pelanggan. Model Ticket mencakup informasi nomor tiket, nomor resi, deskripsi permasalahan, prioritas, status tiket, serta keterkaitannya dengan pelanggan dan <i>Customer service</i> .
3.	User.model.ts	Model ini digunakan untuk merepresentasikan data pengguna sistem. Model User menyimpan informasi identitas pengguna, status dan peran.

Berdasarkan model-model pada tabel 5.3 digunakan sebagai representasi struktur data dalam sistem Live Chat dan Ticketing. Setiap model dirancang untuk mendukung proses pengelolaan data pengguna, percakapan, pesan, dan tiket layanan pelanggan secara terstruktur serta mendukung penerapan Role-Based Access Control (RBAC).

5.1.2.3 Controller Aplikasi Web

Controller dalam arsitektur MVC berperan sebagai penghubung antara Model dan View. Controller bertugas menerima permintaan dari pengguna melalui aplikasi frontend berbasis *Angular*, kemudian memproses permintaan tersebut sesuai dengan logika aplikasi yang dijalankan pada backend *Node.js*. Permintaan yang memerlukan pengelolaan data akan diteruskan oleh controller ke model untuk melakukan operasi data pada *Firebase Cloud Firestore*. Hasil dari proses tersebut selanjutnya dikirim kembali ke aplikasi frontend untuk ditampilkan kepada pengguna. Daftar controller yang digunakan dalam sistem ini dapat dilihat pada Tabel 5.4 berikut ini

Tabel 5.4 Controller Aplikasi Web

No	Nama Controller	Deskripsi
1.	Auth.controller.ts	Controller ini digunakan untuk mengelola proses autentikasi dan otorisasi pengguna, meliputi registrasi, login, verifikasi token, pengelolaan sesi, pengambilan profil pengguna, serta pengecekan peran berbasis <i>Role-Based Access Control (RBAC)</i> .
2.	Chat.controller.ts	Controller ini digunakan untuk mengelola fitur <i>live chat</i> , termasuk pembuatan sesi chat, pengiriman dan pengambilan pesan, pengelolaan data chat, serta pengakhiran sesi chat yang terintegrasi dengan sistem ticketing dan <i>Firebase Cloud Firestore</i> .
3.	Ticket.controller.ts	Controller ini digunakan untuk mengelola proses ticketing, meliputi pembuatan, pengambilan, pembaruan, penugasan, penutupan, dan penghapusan tiket sesuai alur bisnis sistem dengan memanfaatkan <i>Firebase Cloud Firestore</i> .
4.	User.controller.ts	Controller ini digunakan untuk mengelola data pengguna, meliputi pembuatan, pengambilan, pembaruan, dan penghapusan user, serta pengaturan peran dan status pengguna dengan integrasi <i>Firebase Authentication</i> dan <i>Firestore</i> .

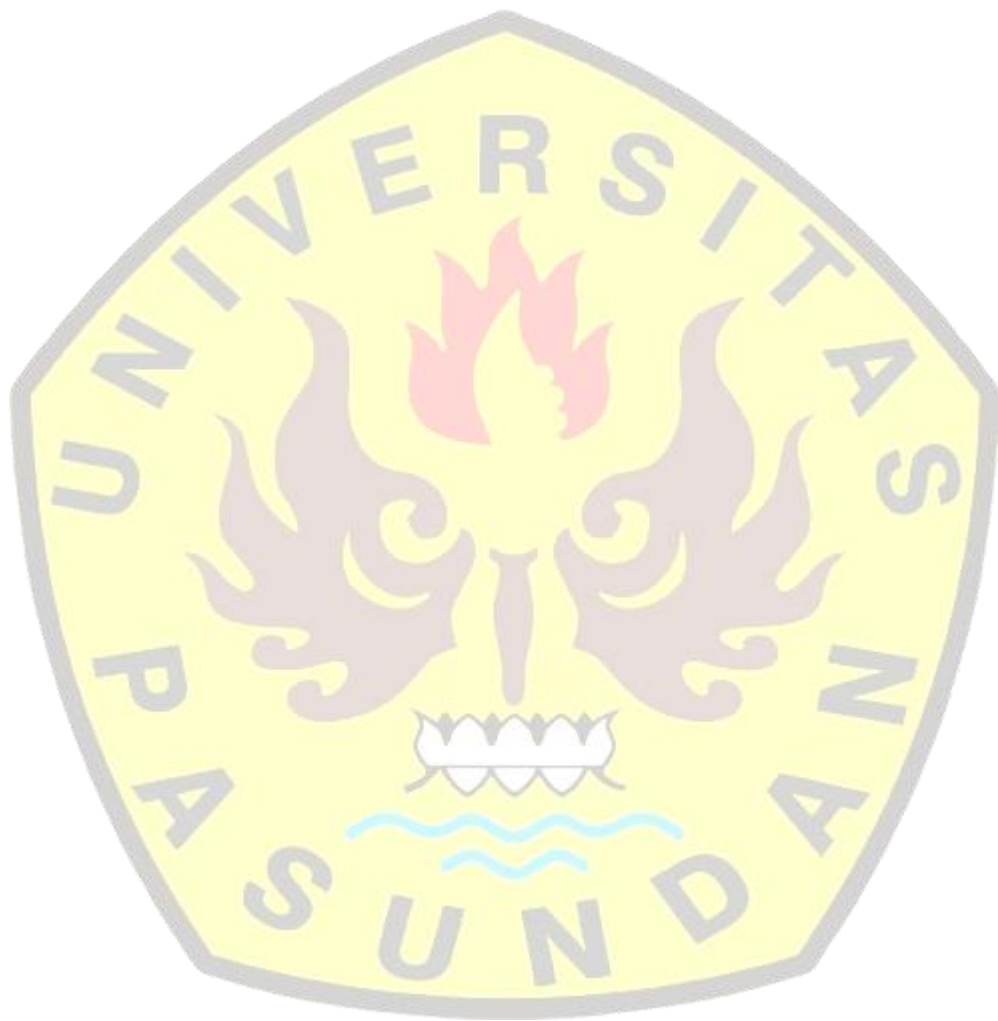
Berdasarkan tabel 5.4, sistem ini menggunakan beberapa controller yang berfungsi mengelola autentikasi pengguna, layanan *live chat*, proses ticketing, serta pengelolaan data pengguna. Seluruh controller berperan sebagai penghubung antara frontend dan backend, serta berinteraksi dengan *Firebase Authentication* dan *Firebase Cloud Firestore* untuk mendukung proses bisnis sistem secara terintegrasi.

5.1.2.4 View Aplikasi Web

View dalam arsitektur MVC berfungsi untuk menampilkan antarmuka pengguna serta menyajikan data yang diterima dari controller dalam bentuk yang mudah dipahami. Pada sistem ini, View diimplementasikan menggunakan framework frontend *Angular*, yang bertugas menampilkan halaman live chat, ticketing, dan manajemen pengguna. View juga menerima input dari pengguna, seperti pengiriman pesan, pembuatan tiket, dan pengelolaan akun, yang selanjutnya diteruskan ke controller untuk diproses. Daftar View yang digunakan dalam sistem ini dapat dilihat pada Tabel 5.5 berikut ini.

Tabel 5.5 View Aplikasi Web

No	Nama View	Deskripsi
1.	Login.component.html	Tampilan ini merupakan halaman login untuk memasukkan email dan password dengan validasi serta fitur lupa password.
2.	Registrasi.component.html	Tampilan ini merupakan halaman registrasi untuk membuat akun baru dengan validasi data pengguna.
3.	Reset-password.component.html	Tampilan ini merupakan halaman reset password dengan memasukkan email terdaftar.
4.	Dashboard.component.html	Tampilan ini merupakan halaman utama <i>Customer</i> yang menampilkan informasi layanan, FAQ, live chat, dan tiket.
5.	List-ticket.component.html	Tampilan ini merupakan halaman daftar tiket dalam bentuk tabel beserta aksi pengelolaan tiket.
6.	Detail-ticket.component.html	Tampilan ini merupakan halaman detail tiket yang menampilkan informasi tiket secara read-only.
7.	Create-edit-ticket.component.html	Tampilan ini merupakan halaman pengelolaan tiket untuk memperbarui prioritas dan penugasan <i>Customer service</i> .
8.	List-ticket-recap.component.html	Tampilan ini merupakan halaman rekap daftar tiket dalam bentuk tabel.
9.	Detail-ticket-recap.component.html	Tampilan ini merupakan halaman detail rekap tiket yang ditampilkan secara read-only.



No	Nama View	Deskripsi
10.	Chat-history-component.html	Tampilan ini merupakan halaman riwayat live chat yang menampilkan daftar dan isi percakapan.
11.	Chat.component.html	Tampilan ini merupakan halaman live chat <i>Customer service</i> untuk komunikasi <i>Real-time</i> dengan <i>Customer</i> .

Berdasarkan Tabel 5.5, tampilan antarmuka pada aplikasi Live Chat dan Ticketing terdiri dari halaman login, registrasi, dan reset password yang digunakan untuk proses autentikasi pengguna. Selain itu, tersedia halaman dashboard *Customer* sebagai pusat akses layanan, live chat, dan pengajuan tiket. Sistem juga menyediakan halaman pengelolaan tiket yang meliputi daftar tiket, detail tiket, serta halaman edit tiket dan rekap tiket untuk mendukung proses monitoring dan pembaruan data. Fitur komunikasi *Real-time* diimplementasikan melalui halaman chat history dan chat *Customer service* yang menampilkan daftar percakapan, riwayat pesan, serta pengiriman pesan secara langsung.

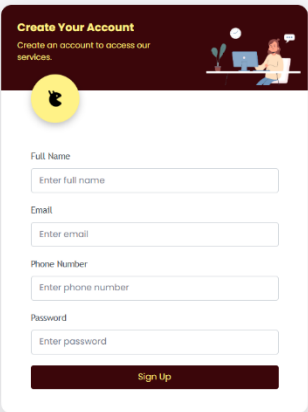
5.1.3 Implementasi Antarmuka Pengguna

Implementasi antarmuka pengguna merupakan hasil implementasi dari perancangan antarmuka di bab sebelumnya. Berikut merupakan implementasi antarmuka pada aplikasi website pengiriman barang.

Tabel 5.6 Implementasi Antar Muka Halaman Beranda

Implementasi Antar Muka Halaman Beranda	
	
Gambar 5.1 Implementasi Antar Muka Halaman Beranda	
Deskripsi	
Gambar diatas menampilkan halaman utama pada situs web Massbox yang berfungsi sebagai platform penyedia layanan pengiriman barang.	

Tabel 5.7 Implementasi Antar Muka Halaman Register

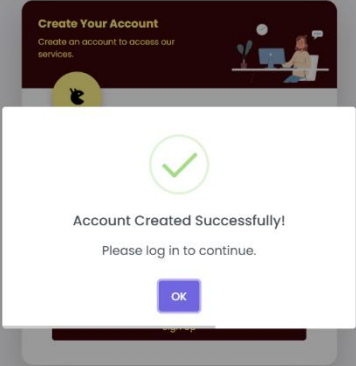
Implementasi Antar Muka Halaman Register


Gambar 5.2 Implementasi Antar Muka Halaman Register

Deskripsi

Gambar di atas menampilkan halaman pendaftaran yang memuat formulir registrasi, yang terdiri atas kolom nama lengkap (full name), alamat surel (email), no telepon dan kata sandi (password) yang wajib diisi oleh pengguna.

Tabel 5.8 Implementasi Antar Muka Halaman konfirmasi Pendaftaran

Implementasi Antar Muka Halaman Register


Gambar 5.3 Implementasi Antar Muka Halaman konfirmasi Pendaftaran

Deskripsi

Gambar di atas menampilkan halaman konfirmasi pendaftaran yang menampilkan informasi bahwa akun pengguna telah berhasil dibuat, serta memberikan pemberitahuan bahwa pengguna dapat melanjutkan proses masuk (login) menggunakan kredensial yang telah didaftarkan.

Tabel 5.9 Implementasi Antar Muka Fitur Live Chat Sebelum Customer Melakukan Login



Gambar 5.4 Implementasi Antar Muka Fitur Live Chat Sebelum Customer Melakukan Login

Deskripsi

Gambar diatas menampilkan halaman awal yang ditampilkan kepada pengguna sebelum mengakses fitur live chat, yang berfungsi untuk menyampaikan informasi bahwa pengguna diwajibkan melakukan proses autentikasi (login) terlebih dahulu sebelum dapat menggunakan layanan tersebut.

Tabel 5.10 Implementasi Antar Muka Halaman Jendela Chat



Gambar 5.5 Implementasi Antar Muka Halaman Jendela Chat

Deskripsi

Gambar di atas menampilkan halaman setelah proses pendaftaran dan autentikasi (login) berhasil, di mana saat ikon chat diakses akan muncul jendela percakapan yang menampilkan sapaan dari MassBot serta beberapa tombol quick question yang berfungsi untuk memudahkan pengguna dalam mengajukan pertanyaan secara cepat.

Tabel 5.11 Implementasi Antar Muka Halaman Quick Question Chat

Implementasi Antar Muka Halaman Quick Question Chat

<p style="text-align: center;">Gambar 5.6 Implementasi Antar Muka Halaman Quick Question Chat</p>
<p style="text-align: center;">Deskripsi</p>
<p>Gambar di atas menampilkan halaman setelah pengguna menekan tombol quick question, di mana MassBot menampilkan pertanyaan yang dipilih beserta respons otomatis, sehingga interaksi antara pengguna dan sistem dapat berlangsung secara dinamis dan efisien.</p>

Tabel 5.12 Implementasi Antar Muka Halaman Pelaporan Kendala

Implementasi Antar Muka Halaman Pelaporan Kendala

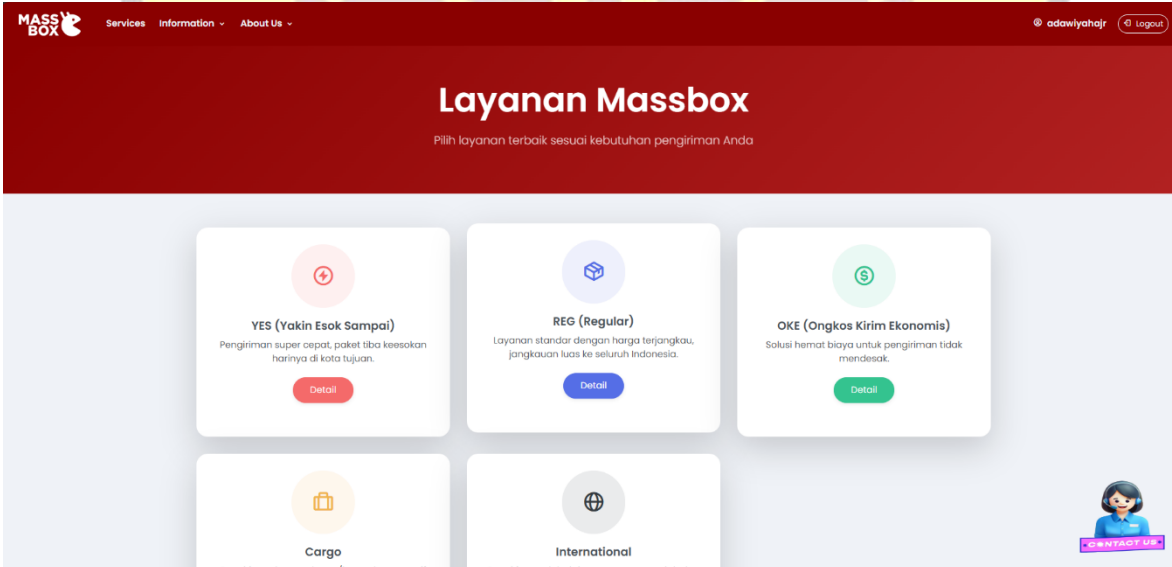
<p style="text-align: center;">Gambar 5.7 Implementasi Antar Muka Halaman Pelaporan Kendala</p>
<p style="text-align: center;">Deskripsi</p>
<p>Gambar di atas menampilkan halaman percakapan ketika pengguna memilih opsi "Tanya Hal Lain" pada menu quick question, di mana sistem menampilkan formulir percakapan yang memungkinkan pengguna untuk menyampaikan kendala atau permasalahan yang dialami secara langsung kepada layanan pelanggan.</p>

Tabel 5.13 Implementasi Antar Muka Halaman Chat Laporan Tiket

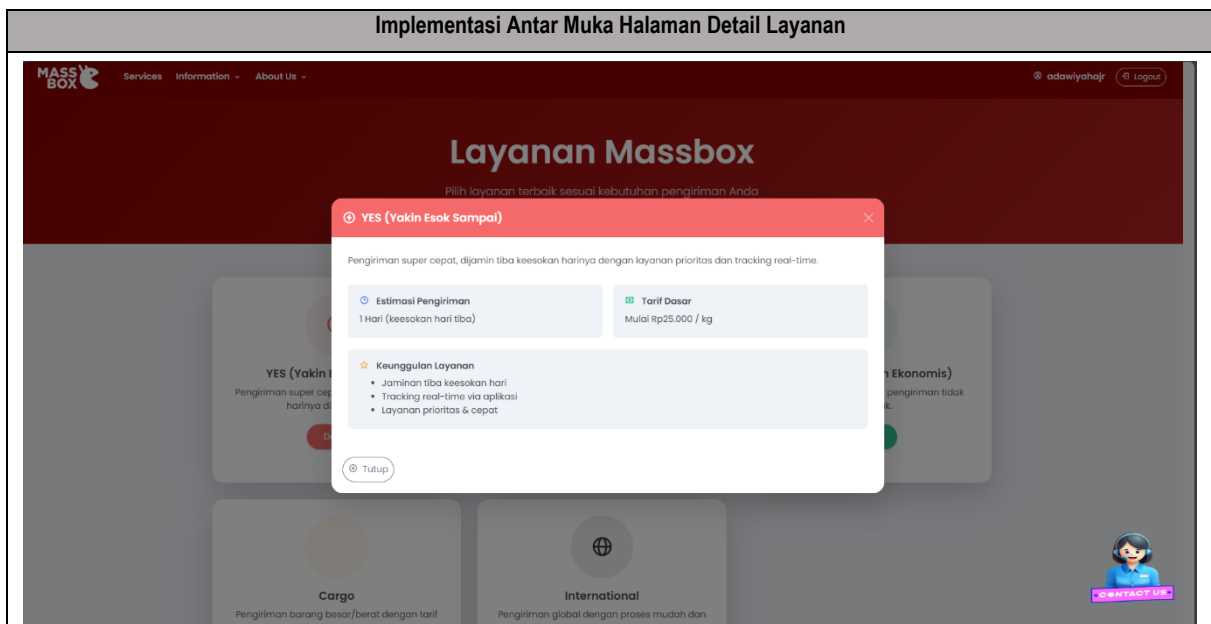
Implementasi Antar Muka Halaman Chat Laporan Tiket

<p style="text-align: center;">Gambar 5.8 Implementasi Antar Muka Halaman Chat Laporan Tiket</p>
<p style="text-align: center;">Deskripsi</p>
<p>Gambar di atas menampilkan halaman percakapan setelah pengguna mengirimkan tiket laporan, di mana MassBot memberikan respons otomatis yang menyatakan bahwa laporan telah diterima dan akan segera diproses oleh pihak layanan pelanggan (<i>Customer service</i>).</p>

Tabel 5.14 Implementasi Antar Muka Halaman Layanan

Implementasi Antar Muka Halaman Layanan

<p style="text-align: center;">Gambar 5.9 Implementasi Antar Muka Halaman Layanan</p>
<p style="text-align: center;">Deskripsi</p>
<p>Gambar di atas menampilkan halaman layanan Massbox yang menampilkan daftar layanan pengiriman dalam bentuk card. Pengguna dapat melihat informasi singkat dari setiap layanan yang tersedia serta menekan tombol “Detail” untuk menampilkan jendela modal yang memuat deskripsi layanan secara lengkap, estimasi waktu pengiriman, tarif dasar, serta keunggulan masing-masing layanan.</p>

Tabel 5.15 Implementasi Antar Muka Halaman Detail Layanan

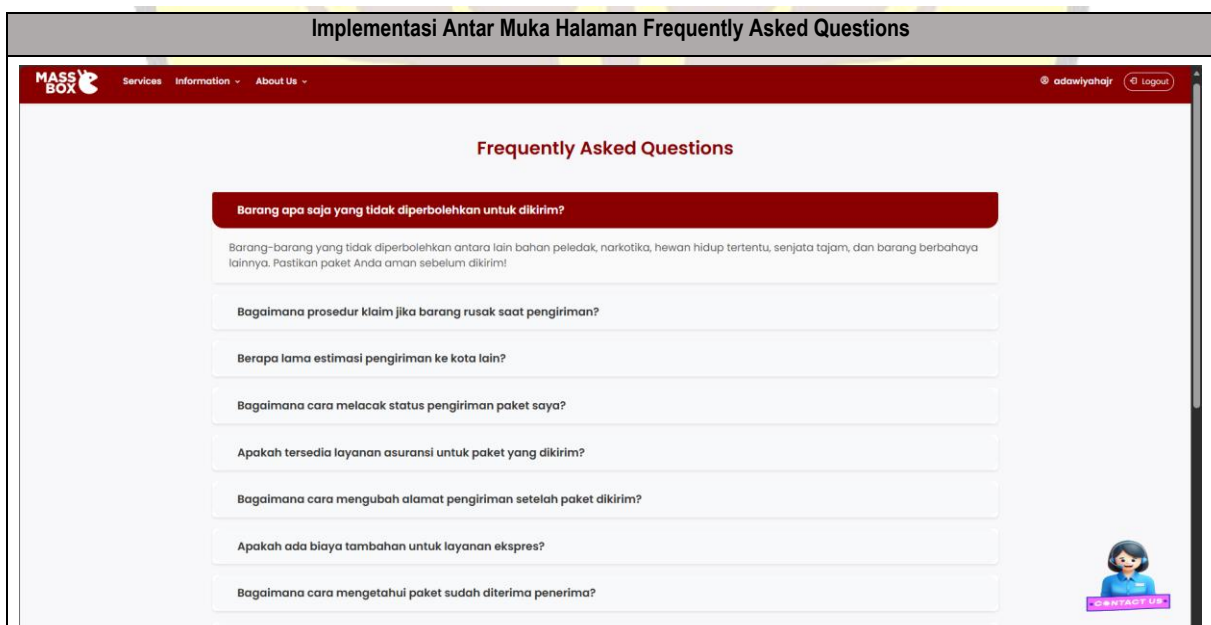


Gambar 5.10 Implementasi Antar Muka Halaman Detail Layanan

Deskripsi

Gambar di atas menampilkan halaman detail layanan yang ditampilkan setelah pengguna menekan tombol "Detail", di mana jendela modal menyajikan informasi lengkap mengenai layanan, yang meliputi deskripsi, estimasi waktu pengiriman, tarif dasar, serta keunggulan layanan yang ditawarkan.

Tabel 5.16 Implementasi Antar Muka Halaman Frequently Asked Questions

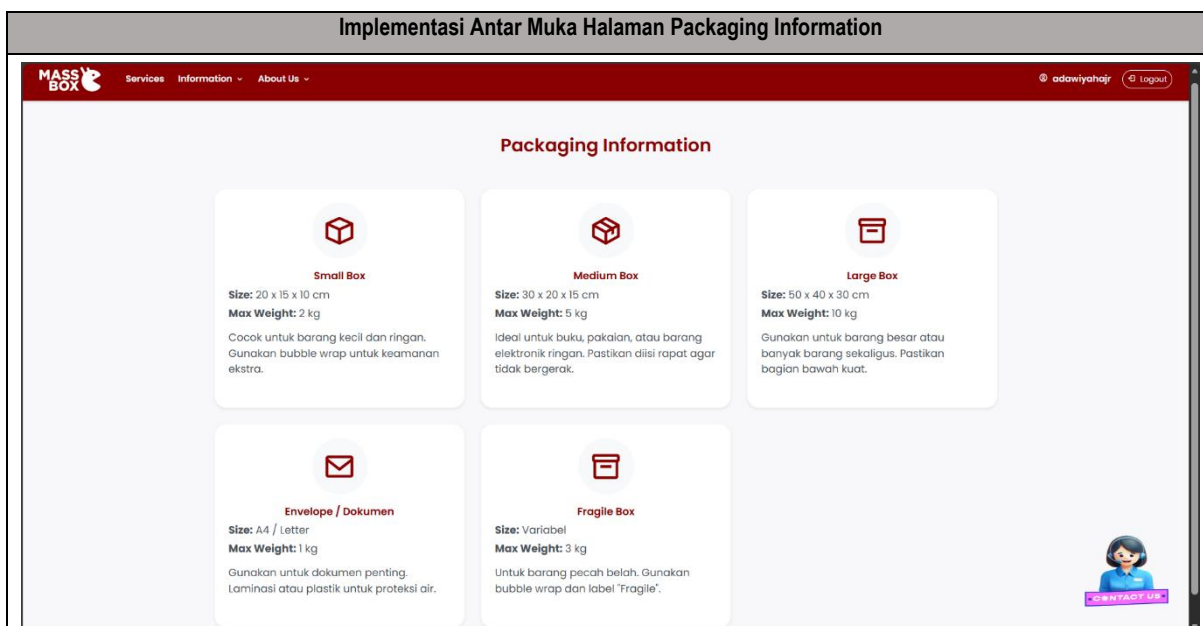


Gambar 5.11 Implementasi Antar Muka Halaman Frequently Asked Questions

Deskripsi

Gambar di atas menampilkan halaman Frequently Asked Questions (FAQ) Massbox, yang memuat daftar pertanyaan umum dalam bentuk accordion. Pengguna dapat menekan setiap pertanyaan untuk menampilkan jawaban terkait secara interaktif.

Tabel 5.17 Implementasi Antar Muka Halaman Packaging Information

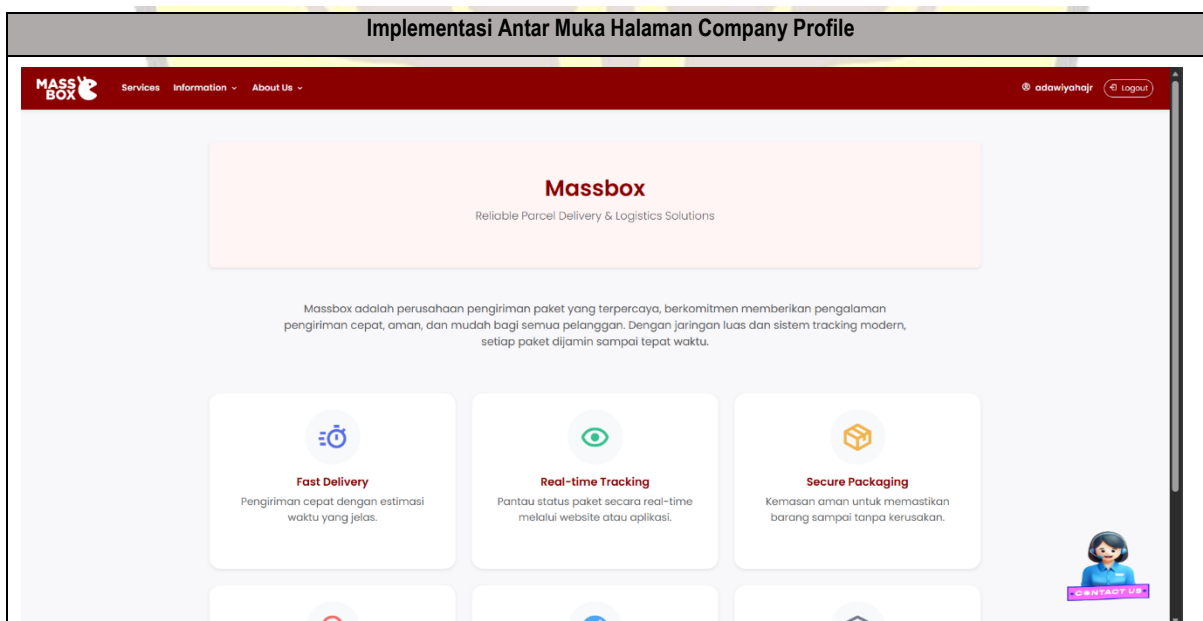


Gambar 5.12 Implementasi Antar Muka Halaman Packaging Information

Deskripsi

Gambar di atas menampilkan halaman informasi Packaging pada Massbox yang menyajikan daftar jenis kemasan dalam bentuk card. Setiap card menampilkan informasi berupa nama kemasan, ukuran, berat maksimum, serta tips penggunaan yang bertujuan untuk membantu pengguna dalam memilih kemasan yang sesuai dengan kebutuhan pengiriman.

Tabel 5.18 Implementasi Antar Muka Halaman Company Profile



Gambar 5.13 Implementasi Antar Muka Halaman Company Profile

Deskripsi

Gambar di atas menampilkan halaman Company Profile pada Massbox yang memuat deskripsi perusahaan serta layanan utama yang ditawarkan. Halaman ini menyajikan informasi mengenai komitmen Massbox dalam menyediakan layanan pengiriman paket yang cepat, aman, dan terpercaya, serta berbagai layanan unggulan yang dirancang untuk mendukung pengalaman pengiriman pelanggan.

Tabel 5.19 Perancangan Antarmuka Halaman Notifikasi Penutup Percakapan

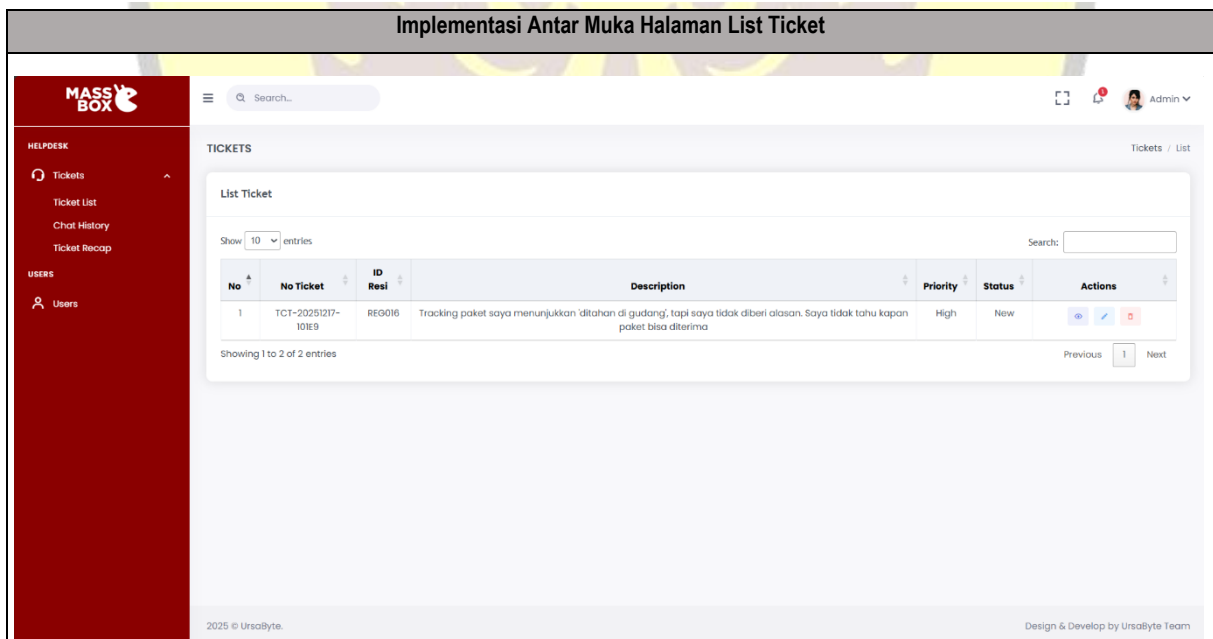


Gambar 5.14 Perancangan Antarmuka Halaman Notifikasi Penutup Percakapan

Deskripsi

Gambar di atas menampilkan halaman percakapan setelah pihak layanan pelanggan (*Customer service*) mengakhiri interaksi dengan pengguna, di mana sistem menampilkan pesan penutup sebagai notifikasi akhir dari percakapan.

Tabel 5.20 Implementasi Antar Muka Halaman List Ticket

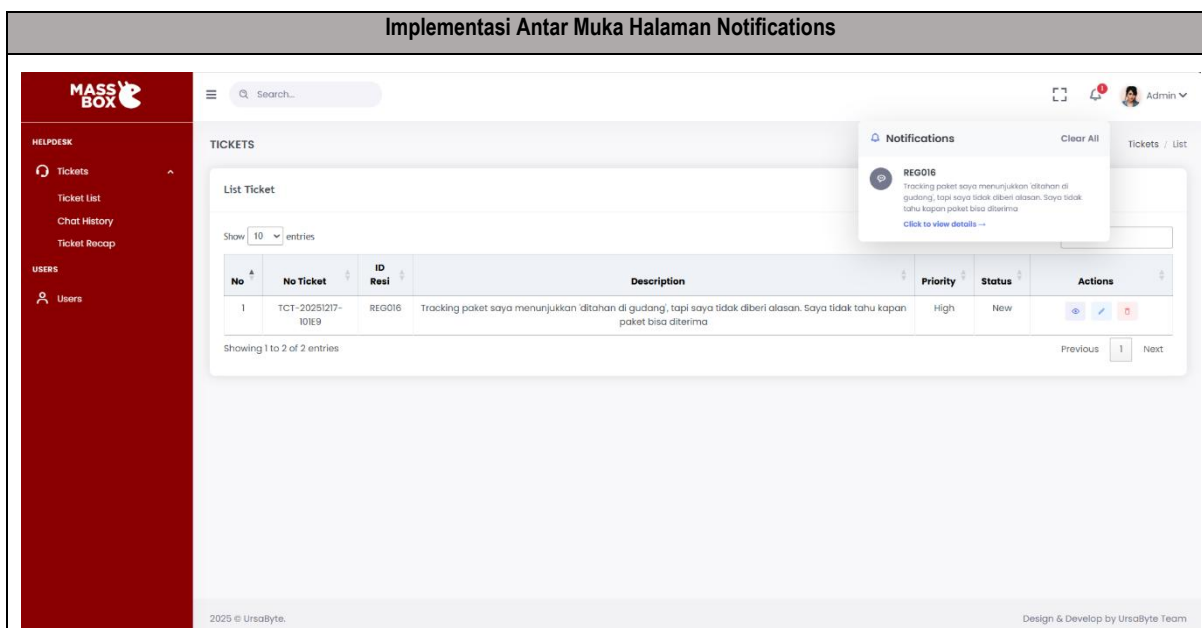


Gambar 5.15 Implementasi Antar Muka Halaman List Ticket

Deskripsi

Gambar di atas menampilkan halaman daftar tiket pengguna yang menyajikan tabel berisi informasi tiket, meliputi nomor tiket, ID resi, deskripsi permasalahan, tingkat prioritas, dan status. Melalui halaman ini, admin dapat melihat detail tiket, melakukan pengeditan, atau menghapus tiket melalui tombol aksi yang tersedia pada setiap baris data.

Tabel 5.21 Implementasi Antar Muka Halaman Notifications

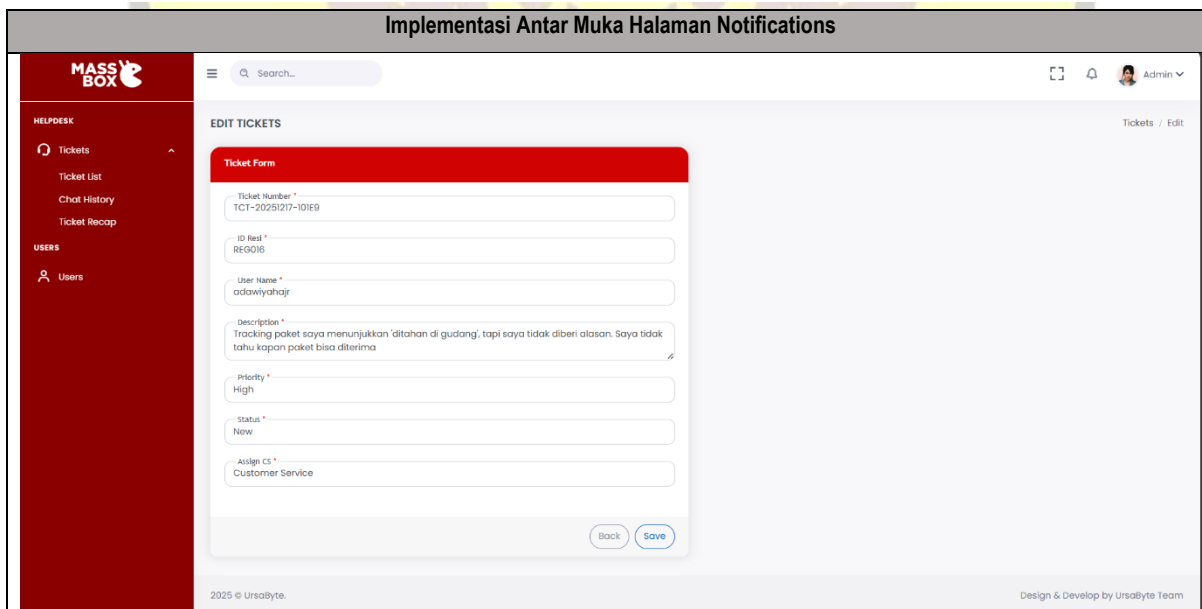


Gambar 5.16 Implementasi Antar Muka Halaman Notifications

Deskripsi

Gambar di atas menampilkan notifikasi setelah pelanggan mengirimkan kendala atau permasalahan, yang menunjukkan bahwa laporan telah berhasil diterima oleh sistem dan akan segera ditindaklanjuti.

Tabel 5.22 Implementasi Antar Muka Halaman Edit Ticket

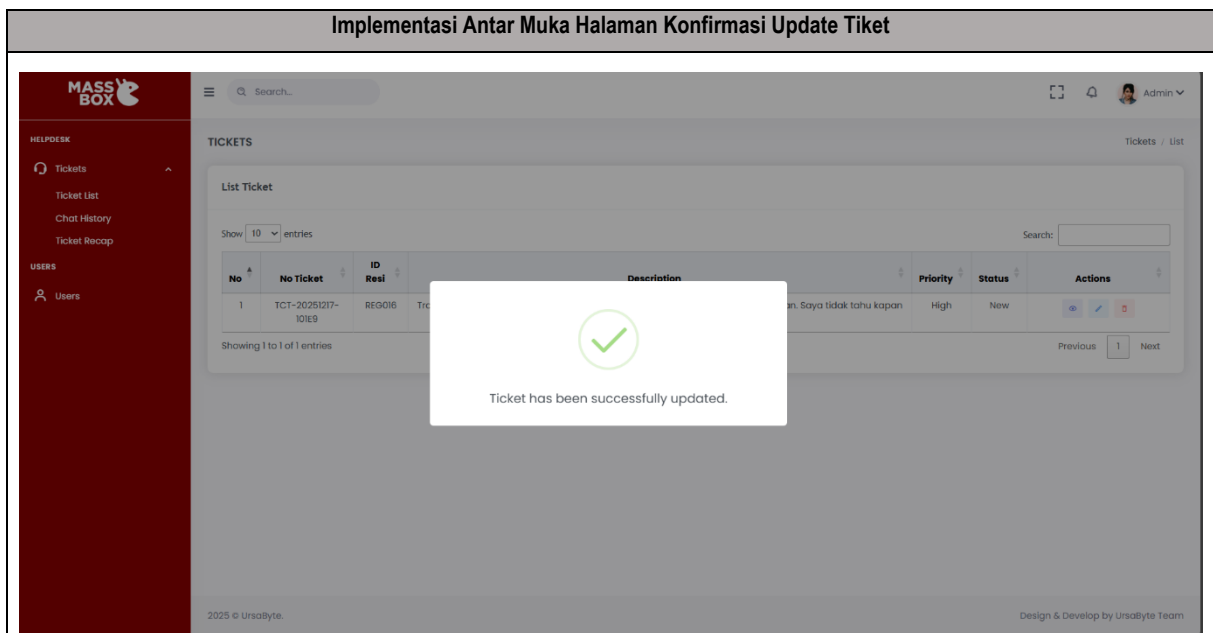


Gambar 5.17 Implementasi Antar Muka Halaman Edit Ticket

Deskripsi

Gambar di atas menampilkan halaman setelah pengguna mengklik notifikasi tiket, di mana sistem menampilkan halaman Edit Ticket yang memungkinkan pengguna untuk memperbarui informasi maupun status tiket yang telah dibuat.

Tabel 5.23 Implementasi Antar Muka Halaman Konfirmasi Update Tiket

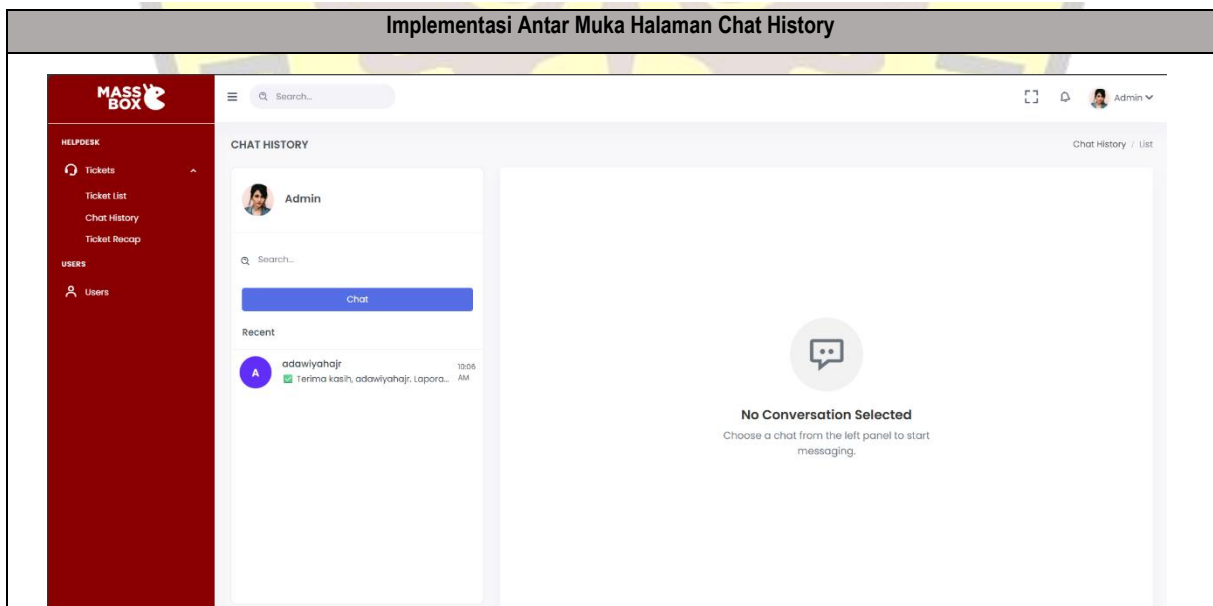


Gambar 5.18 Implementasi Antar Muka Halaman Konfirmasi Update Tiket

Deskripsi

Gambar di atas menampilkan notifikasi konfirmasi setelah pengguna berhasil memperbarui status tiket, yang memberikan informasi bahwa perubahan tersebut telah berhasil disimpan.

Tabel 5.24 Implementasi Antar Muka Halaman Pemberitahuan Ketika Tidak Ada Percakapan yang Dipilih

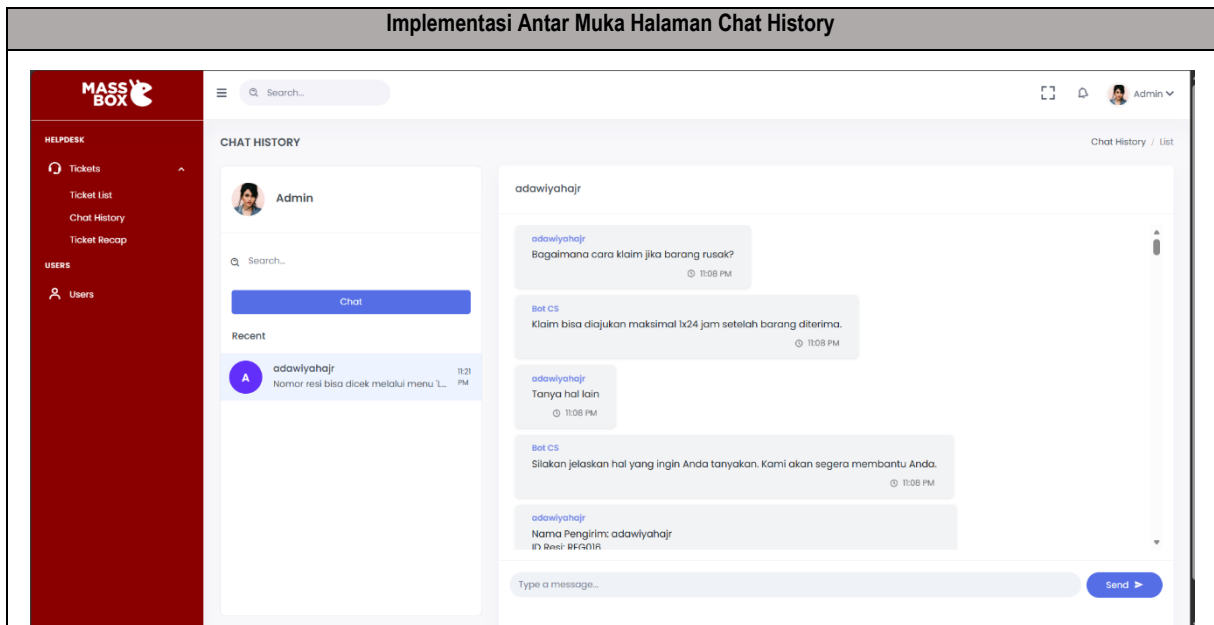


Gambar 5.19 Implementasi Antar Muka Halaman Pemberitahuan Ketika Tidak Ada Percakapan yang Dipilih

Deskripsi

Gambar diatas menampilkan pemberitahuan yang ditampilkan ketika belum ada percakapan yang dipilih oleh pengguna, dengan tujuan memberikan arahan agar pengguna memilih percakapan sebelum memulai pengiriman pesan.

Tabel 5.25 Implementasi Antar Muka Halaman Chat History

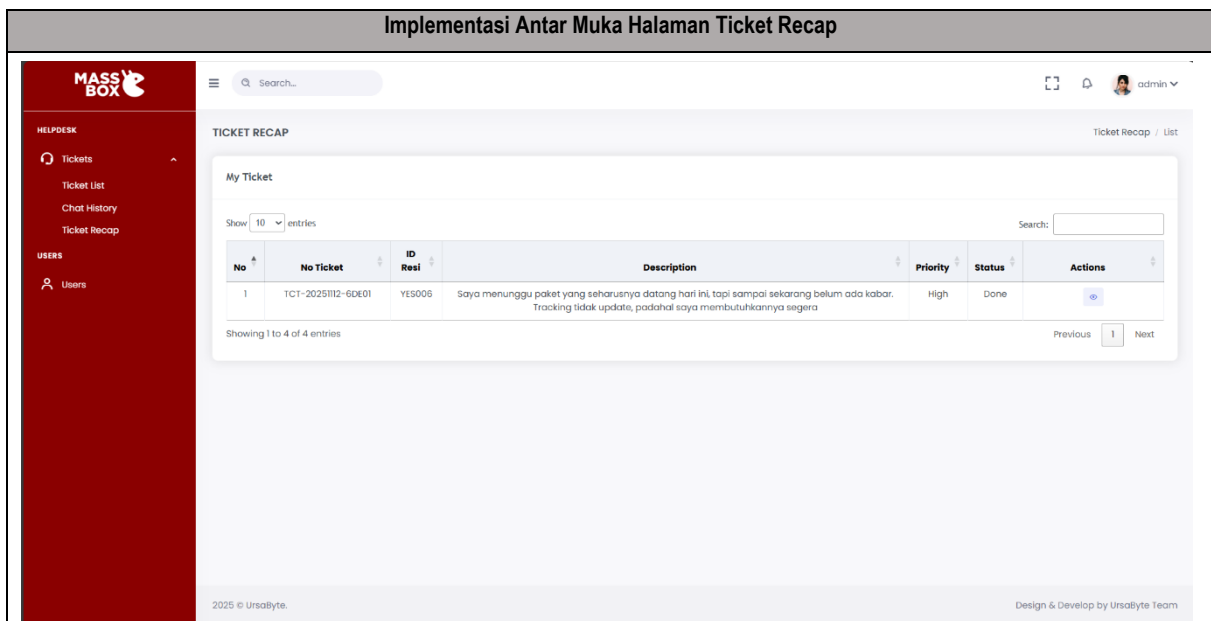


Gambar 5.20 Implementasi Antar Muka Halaman Chat History

Deskripsi

Gambar di atas menampilkan halaman percakapan yang menyajikan riwayat pesan antara pihak layanan pelanggan dan pengirim tiket, yang berfungsi untuk memudahkan proses tindak lanjut serta menjadi referensi terhadap interaksi yang telah dilakukan sebelumnya.

Tabel 5.26 Implementasi Antar Muka Halaman Ticket Recap

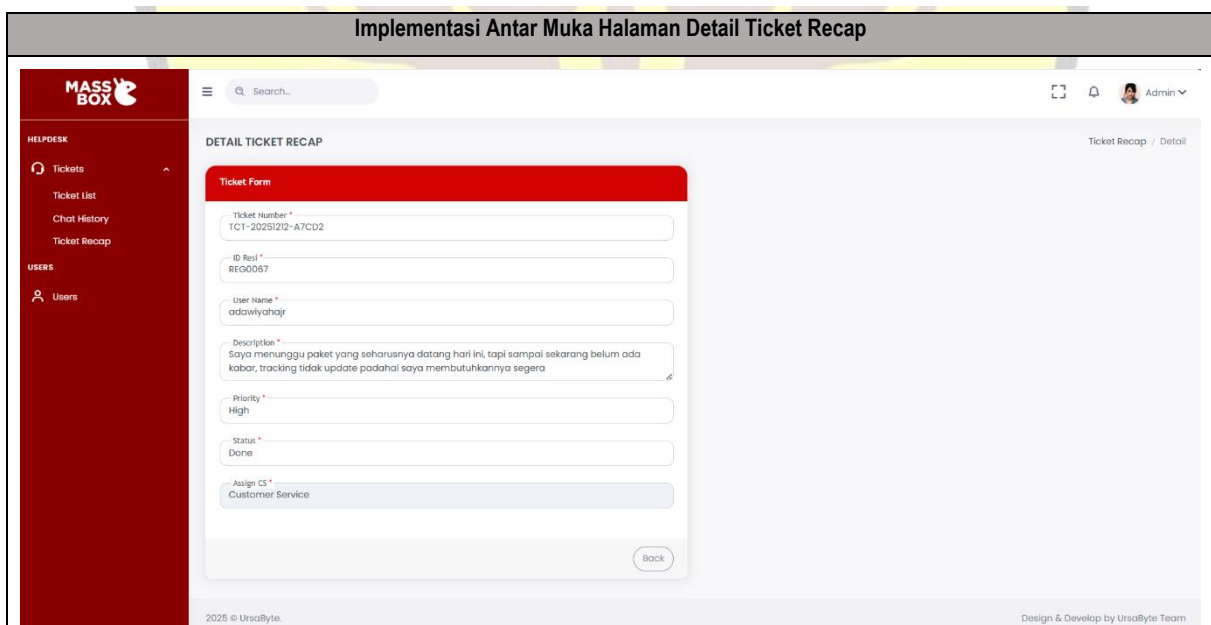


Gambar 5.21 Implementasi Antar Muka Halaman Ticket Recap

Deskripsi

Gambar di atas menampilkan halaman rekap tiket yang menampilkan daftar tiket pengguna dengan status selesai. Halaman ini menyajikan informasi berupa nomor tiket, ID resi, deskripsi, prioritas, dan status, serta menyediakan tombol "View" yang memungkinkan admin meninjau detail setiap tiket.

Tabel 5.27 Implementasi Antar Muka Halaman Detail Ticket Recap

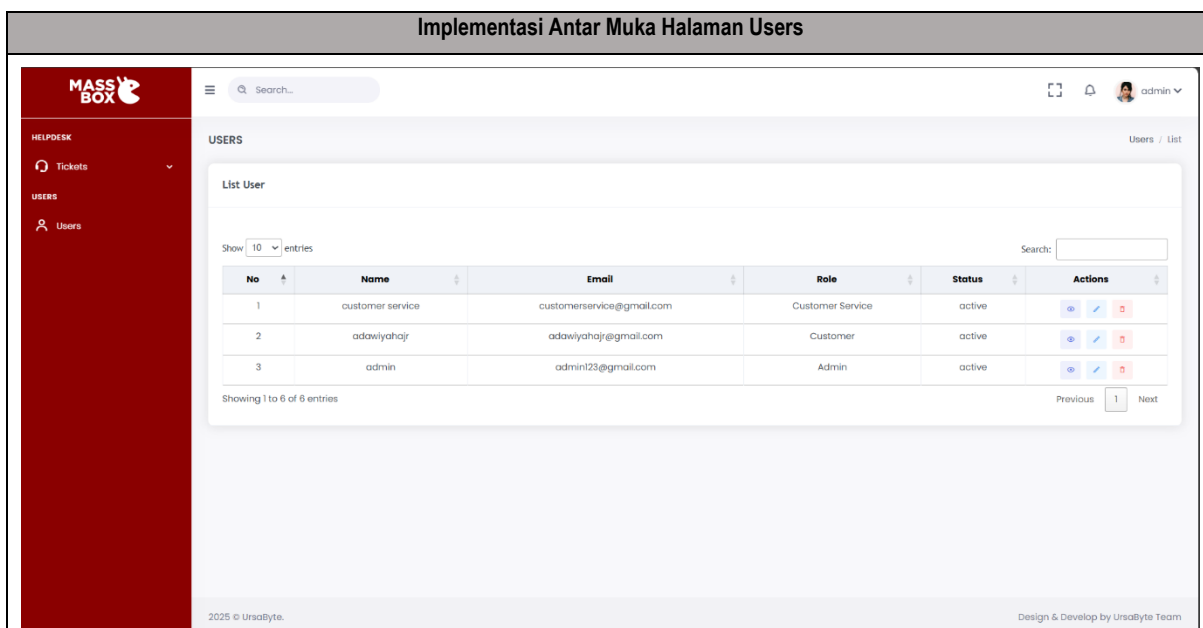


Gambar 5.22 Implementasi Antar Muka Halaman Detail Ticket Recap

Deskripsi

Gambar di atas menampilkan halaman detail tiket yang ditampilkan setelah admin menekan tombol "View" pada daftar tiket, di mana sistem menyajikan informasi lengkap terkait tiket tersebut untuk keperluan peninjauan lebih lanjut.

Tabel 5.28 Implementasi Antar Muka Halaman Users

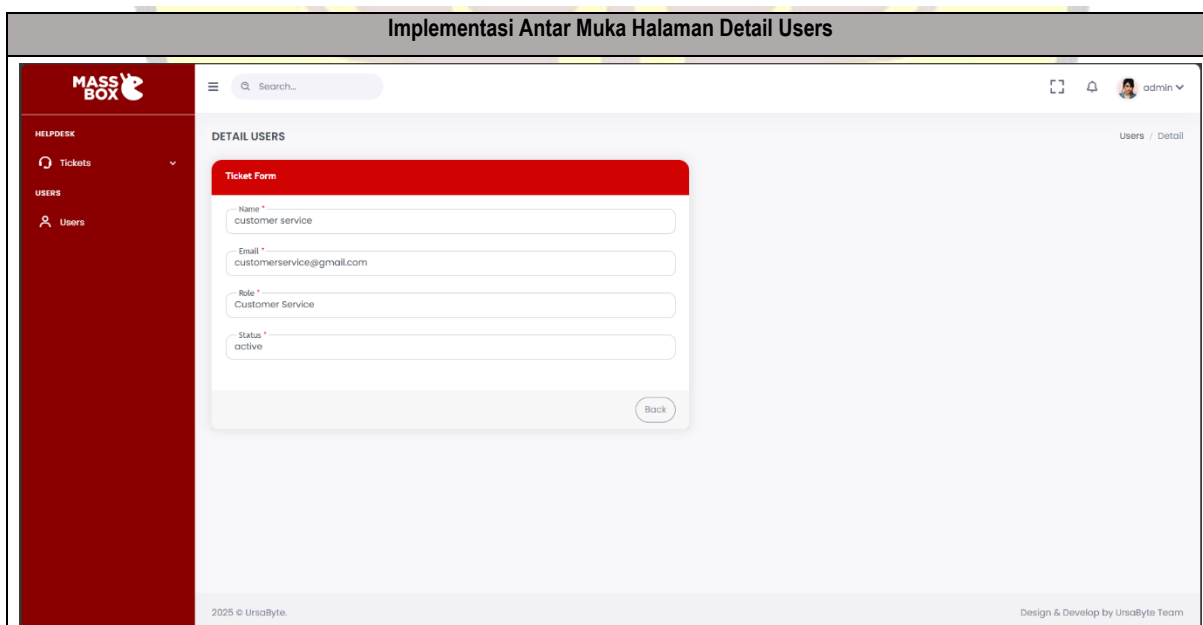


Gambar 5.23 Implementasi Antar Muka Halaman Users

Deskripsi

Gambar di atas menampilkan halaman daftar pengguna yang menyajikan tabel berisi informasi setiap pengguna, meliputi nama, email, role, dan status. Admin dapat mengelola data pengguna dengan melihat detail, melakukan pengeditan, atau menghapus data melalui tombol aksi yang tersedia pada setiap baris.

Tabel 5.29 Implementasi Antar Muka Halaman Detail Users



Gambar 5.24 Implementasi Antar Muka Halaman Detail Users

Deskripsi

Gambar di atas menampilkan halaman detail pengguna yang berisi kolom input untuk nama, alamat surel, peran, dan status pengguna. Setiap kolom dilengkapi dengan validasi wajib, serta terdapat tombol untuk menyimpan perubahan data atau kembali ke halaman daftar pengguna.

Tabel 5.30 Implementasi Antar Muka Halaman Edit Users

Implementasi Antar Muka Halaman Edit Users

Gambar 5.25 Implementasi Antar Muka Halaman Edit Users

Deskripsi

Gambar di atas menampilkan halaman pengeditan pengguna, di mana admin dapat memperbarui informasi pengguna dengan validasi untuk memastikan data lengkap dan benar sebelum disimpan.

Tabel 5.31 Implementasi Antar Muka Halaman Konfirmasi Penghapusan User

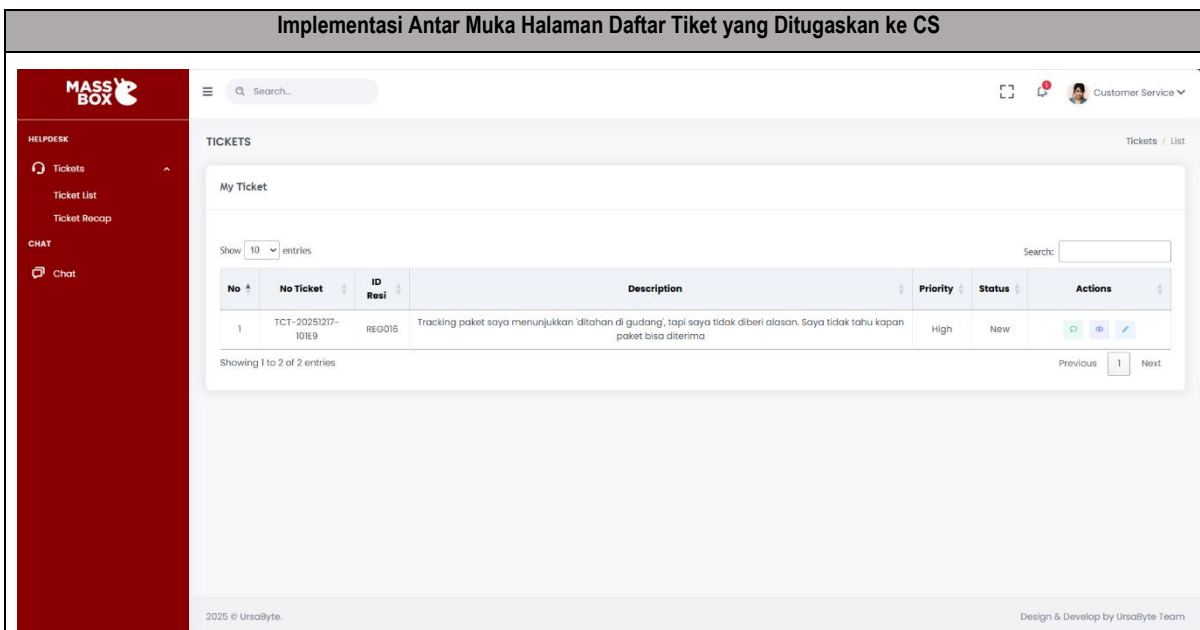
Implementasi Antar Muka Halaman Konfirmasi Penghapusan User

Gambar 5.26 Implementasi Antar Muka Halaman Konfirmasi Penghapusan User

Deskripsi

Gambar di atas menampilkan notifikasi konfirmasi penghapusan pengguna, yang berfungsi untuk meminta persetujuan admin sebelum sistem melakukan penghapusan data pengguna secara permanen.

Tabel 5.32 Implementasi Antar Muka Halaman Daftar Tiket yang Ditugaskan ke CS

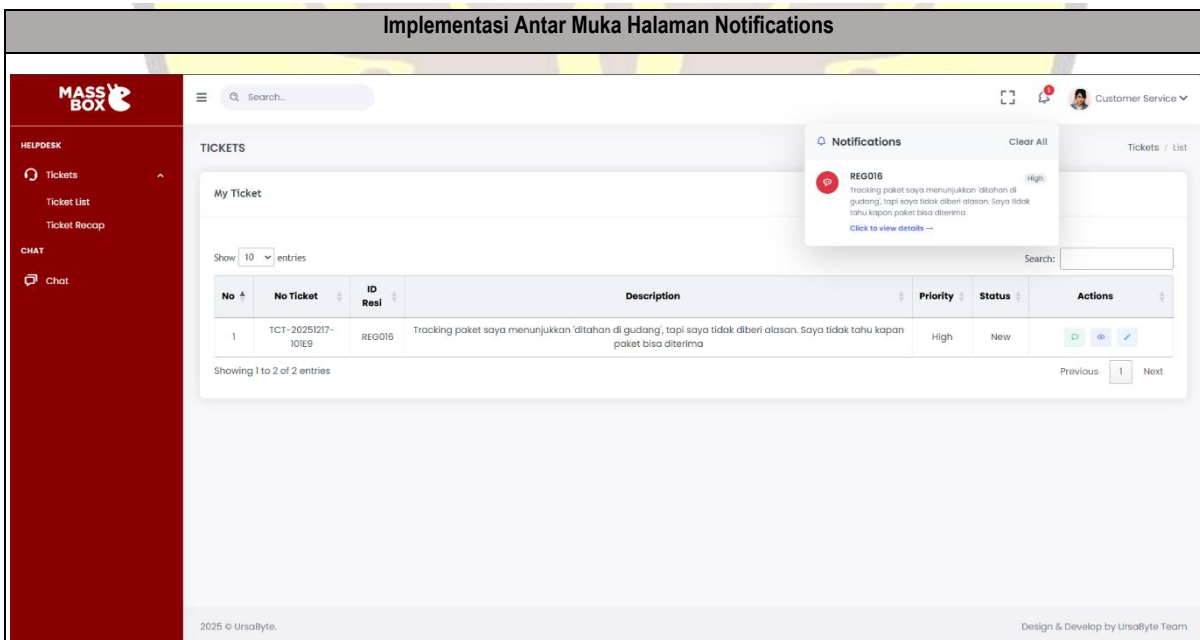


Gambar 5.27 Implementasi Antar Muka Halaman Daftar Tiket yang Ditugaskan ke CS

Deskripsi

Gambar di atas menampilkan halaman daftar ticket yang telah diassign ke *Customer service*, yang menampilkan informasi ticket seperti nomor ticket, ID resi, deskripsi, prioritas, dan status. *Customer service* dapat memulai percakapan dengan pelanggan melalui tombol Chat, melihat detail ticket melalui tombol View, atau memperbarui ticket menggunakan tombol Edit.

Tabel 5.33 Implementasi Antar Muka Halaman Notifications



Gambar 5.28 Implementasi Antar Muka Halaman Notifications

Deskripsi

Gambar di atas menampilkan halaman notifikasi untuk *Customer service*, yang berfungsi memberitahukan adanya ticket baru atau pembaruan ticket yang memerlukan perhatian, sehingga *Customer service* dapat segera menindaklanjuti dengan cepat dan tepat.

Tabel 5.34 Implementasi Antar Muka Halaman Edit Ticket

Implementasi Antar Muka Halaman Edit Ticket

The screenshot shows the 'EDIT' page for a ticket. The form contains the following fields:

- Ticket Number: TCT-20251217-101E9
- ID Real: REG016
- User Name: adawiyahajir
- Description: Tracking paket saya menunjukkan 'ditahan di gudang', tapi saya tidak diberi alasan. Saya tidak tahu kapan paket bisa diterima
- Priority: High
- Status: New
- Assign CS: Customer Service

Buttons: Back, Save

Footer: 2025 © UrsaByte. Design & Develop by UrsaByte Team

Gambar 5.29 Implementasi Antar Muka Halaman Edit Ticket

Deskripsi

Gambar di atas menampilkan halaman Edit Ticket, yang muncul setelah pengguna menekan notifikasi ticket. Pada halaman ini, sistem menampilkan formulir untuk memperbarui informasi atau status ticket yang telah dibuat, guna memastikan data ticket tetap akurat dan terkini.

Tabel 5.35 Implementasi Antar Muka Halaman Konfirmasi Update Ticket

Implementasi Antar Muka Halaman Konfirmasi Update Ticket

The screenshot shows the 'TICKETS' page with a confirmation modal. The modal text is: "Ticket has been successfully updated." The background table shows the following data:

No	No Ticket	ID Real	Priority	Status	Actions
1	TCT-20251217-101E9	REG016	High	In Progress	[Refresh] [Close] [Edit]

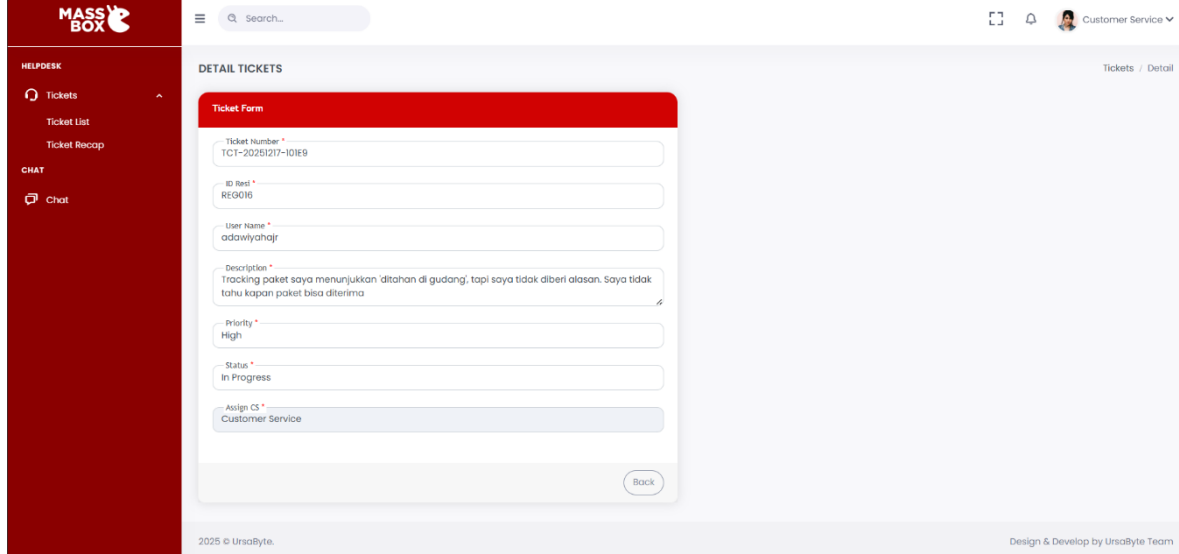
Footer: 2025 © UrsaByte. Design & Develop by UrsaByte Team

Gambar 5.30 Implementasi Antar Muka Halaman Konfirmasi Update Ticket

Deskripsi

Gambar di atas menampilkan halaman notifikasi konfirmasi pembaruan ticket, yang muncul setelah pengguna berhasil memperbarui status ticket. Notifikasi ini berfungsi untuk memberikan umpan balik bahwa perubahan data telah berhasil disimpan oleh sistem.

Tabel 5.36 Implementasi Antar Muka Halaman Detail Tickets

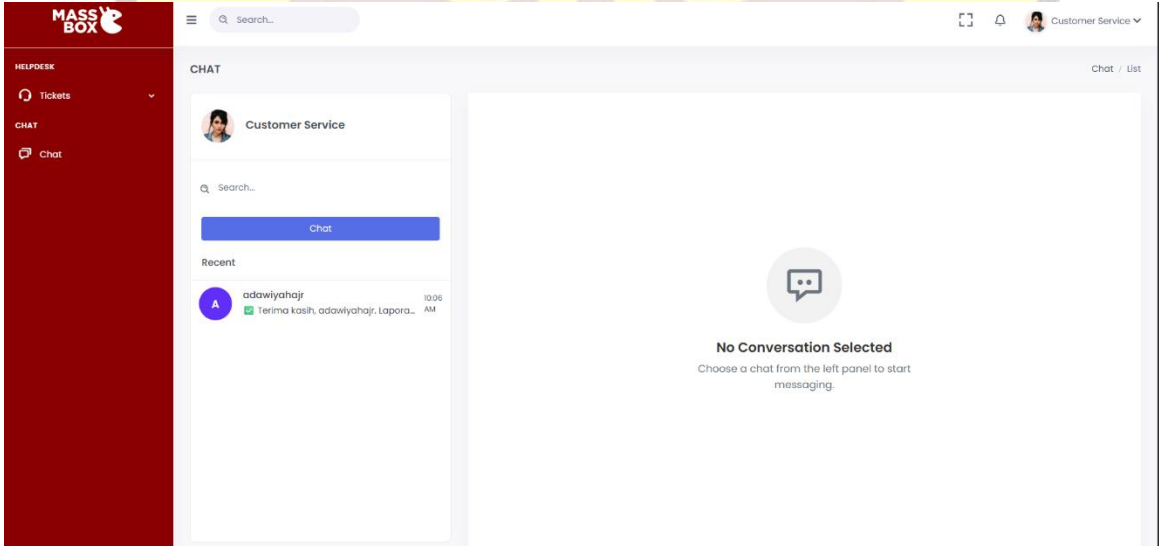
Implementasi Antar Muka Halaman Detail Tickets	
	

Gambar 5.31 Implementasi Antar Muka Halaman Detail Tickets

Deskripsi

Gambar di atas menampilkan halaman detail ticket, yang menampilkan informasi lengkap mengenai ticket tertentu untuk memudahkan *Customer service* dalam meninjau dan menindaklanjuti laporan pengguna.

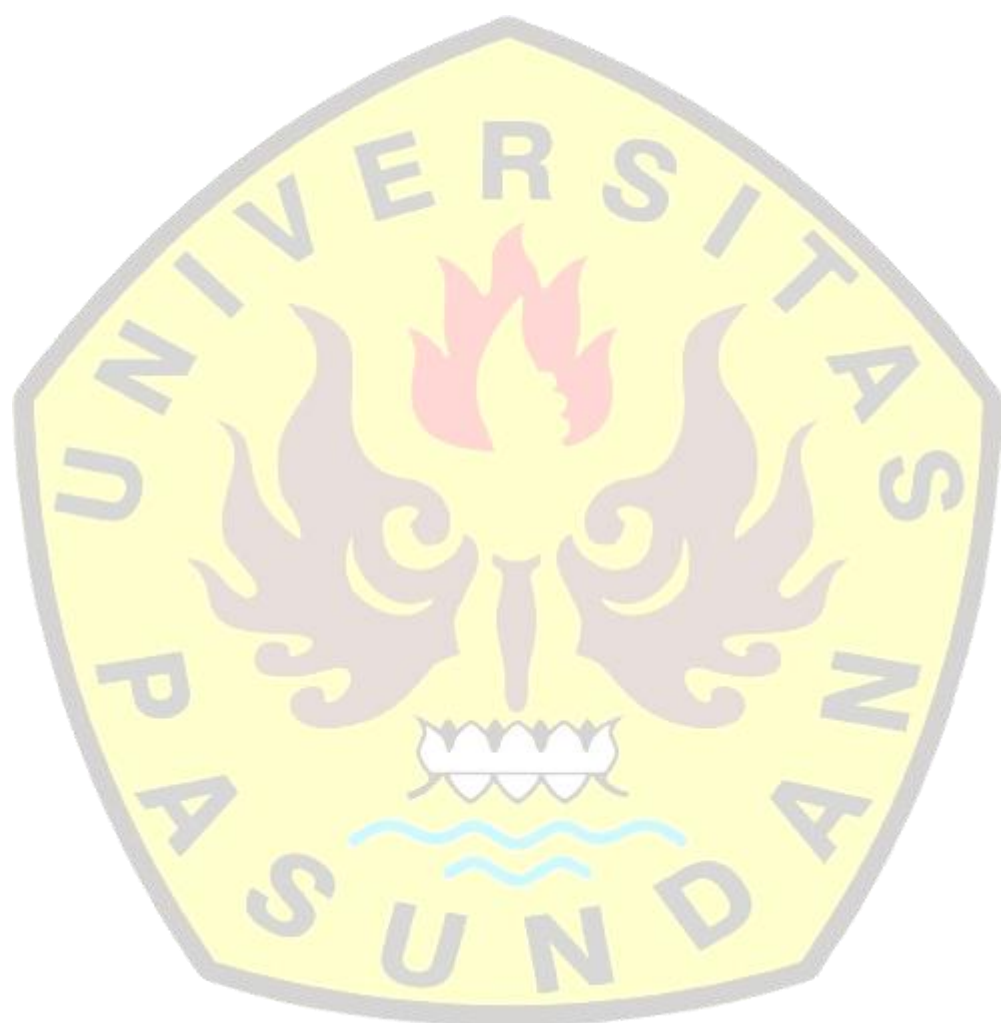
Tabel 5.37 Implementasi Antar Muka Halaman Pemberitahuan Ketika Tidak Ada Percakapan yang Dipilih

Implementasi Antar Muka Halaman Detail Tickets	
	

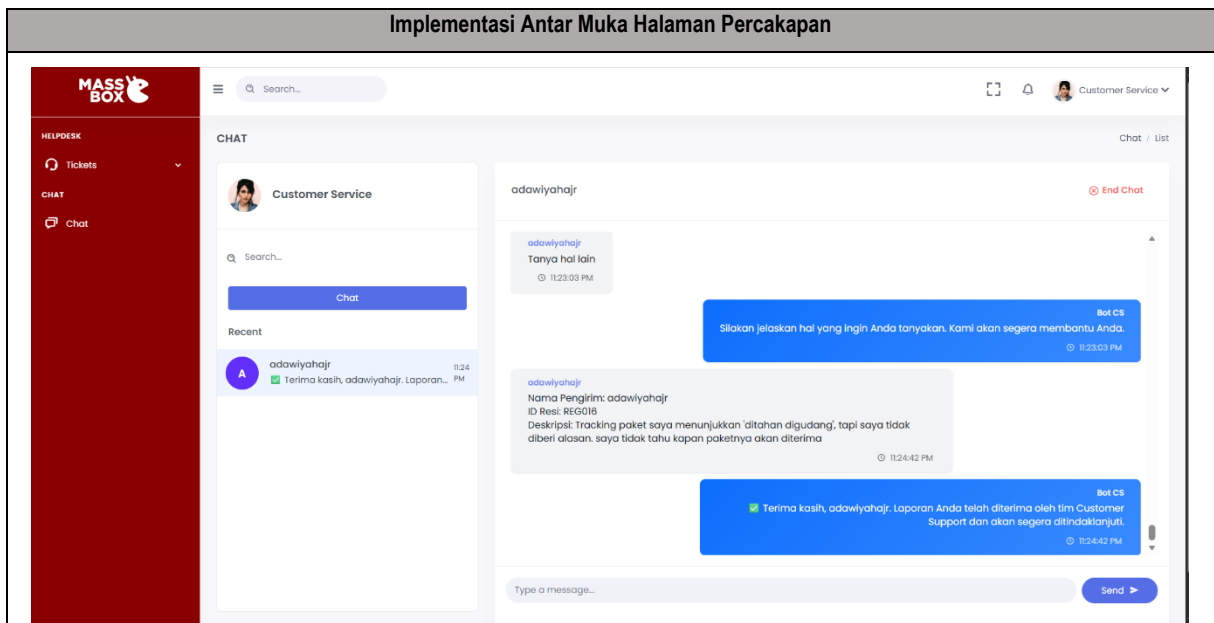
Gambar 5.32 Implementasi Antar Muka Halaman Pemberitahuan Ketika Tidak Ada Percakapan yang Dipilih

Deskripsi

Gambar diatas menampilkan pemberitahuan yang ditampilkan ketika belum ada percakapan yang dipilih oleh pengguna, dengan tujuan memberikan arahan agar pengguna memilih percakapan sebelum memulai pengiriman pesan.



Tabel 5.38 Implementasi Antar Muka Halaman Percakapan

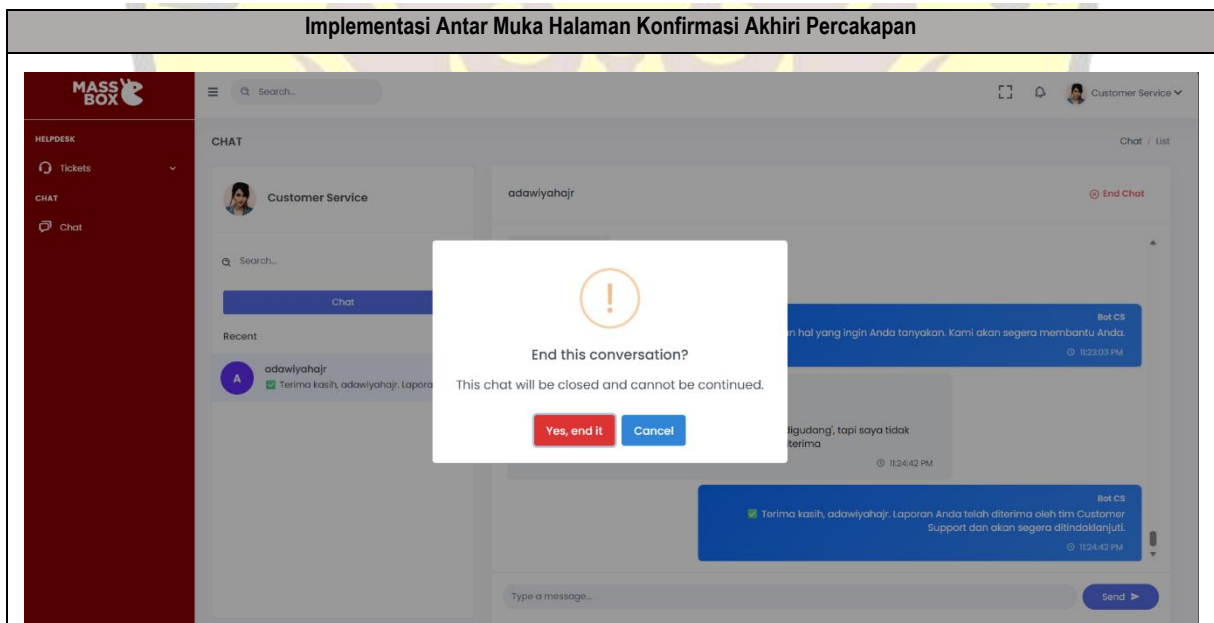


Gambar 5.33 Implementasi Antar Muka Halaman Percakapan

Deskripsi

Gambar di atas menampilkan halaman percakapan ticket, yang menampilkan riwayat komunikasi antara *Customer service* dan pelanggan setelah *Customer service* menekan ikon Chat pada daftar ticket. Tampilan ini memudahkan *Customer service* dalam meninjau konteks percakapan sebelumnya serta melanjutkan komunikasi secara langsung dan responsif.

Tabel 5.39 Implementasi Antar Muka Halaman Konfirmasi Akhir Percakapan

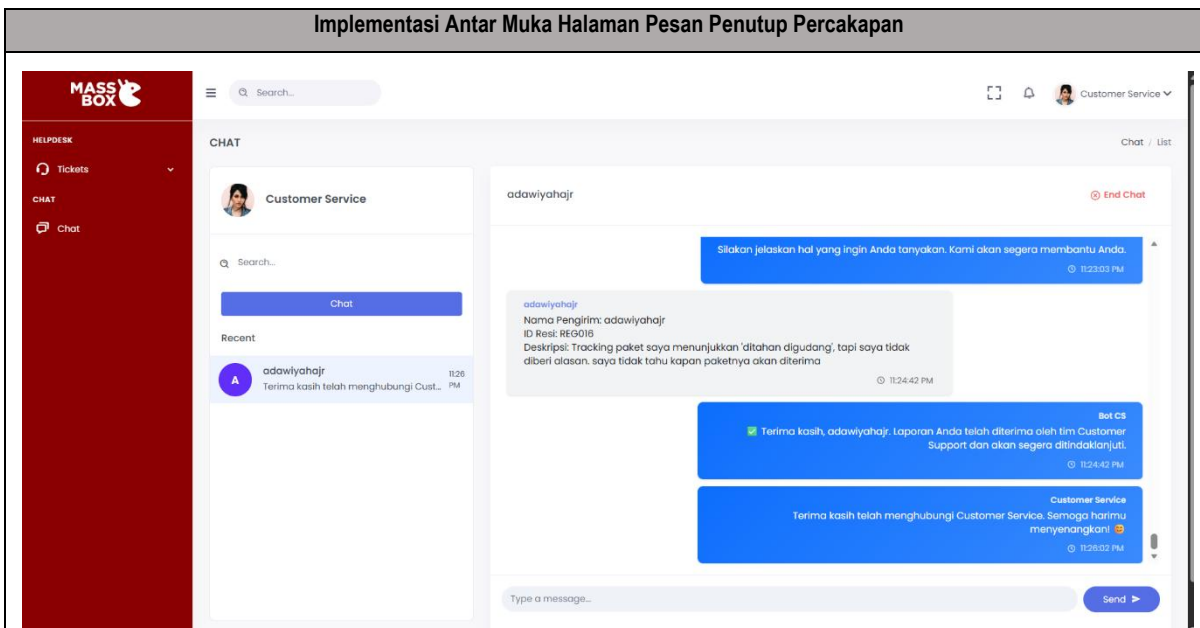


Gambar 5.34 Implementasi Antar Muka Halaman Konfirmasi Akhir Percakapan

Deskripsi

Gambar di atas menampilkan halaman notifikasi konfirmasi akhir percakapan, yang muncul ketika *Customer service* menekan tombol 'End Chat' pada halaman percakapan. Notifikasi ini berfungsi untuk memastikan bahwa *Customer service* benar-benar ingin mengakhiri percakapan secara permanen sebelum tindakan tersebut dikonfirmasi.

Tabel 5.40 Implementasi Antar Muka Halaman Pesan Penutup Percakapan

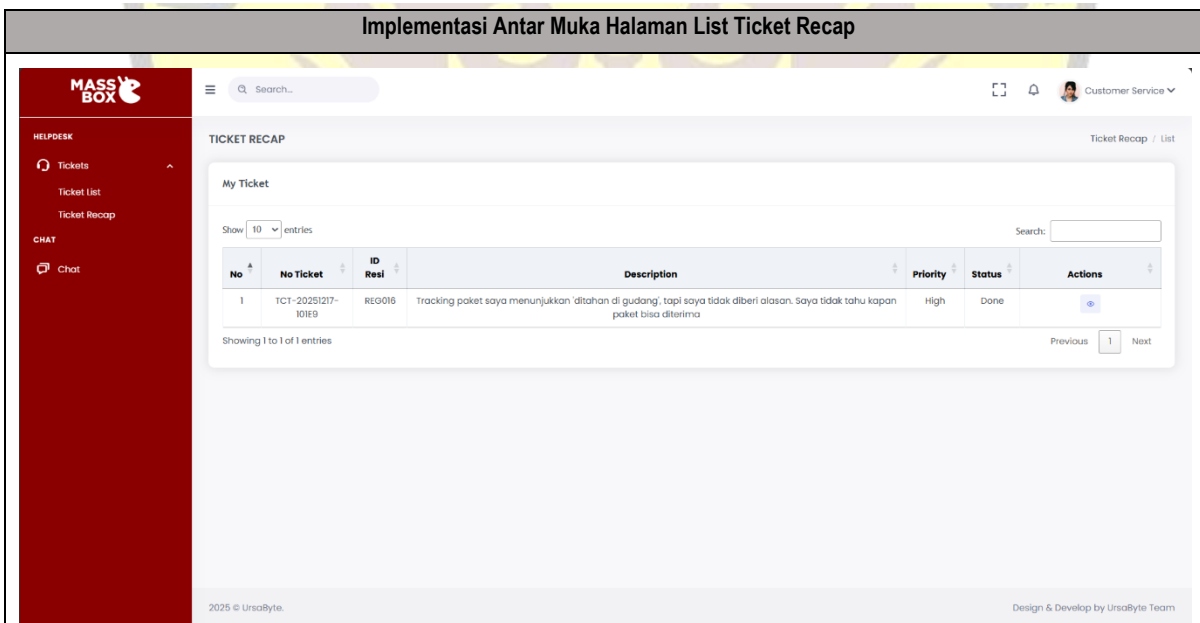


Gambar 5.35 Implementasi Antar Muka Halaman Pesan Penutup Percakapan

Deskripsi

Gambar di atas menampilkan halaman pesan penutup percakapan, yang muncul setelah *Customer service* berhasil mengakhiri sesi chat. Pesan ini berfungsi sebagai pemberitahuan akhir bahwa percakapan telah selesai dan sebagai bentuk ucapan terima kasih atas interaksi yang telah dilakukan.

Tabel 5.41 Implementasi Antar Muka Halaman List Ticket Recap



Gambar 5.36 Implementasi Antar Muka Halaman List Ticket Recap

Deskripsi

Gambar di atas menampilkan halaman detail user, yang menampilkan informasi lengkap terkait data user, seperti nama, email, peran (role), dan status. Setiap kolom memiliki validasi wajib untuk memastikan kelengkapan data, serta menyediakan tombol untuk menyimpan perubahan atau kembali ke daftar pengguna.

Tabel 5.42 Implementasi Antar Muka Halaman Detail Ticket Recap

Implementasi Antar Muka Halaman List Ticket Recap

DETAIL TICKET RECAP

Ticket Form

Ticket Number *
TC1-2025/217-101E9

ID Issue *
REG010

User Name *
ackawlychajr

Description *
Tracking paket saya menunjukkan 'ditahan di gudang', tapi saya tidak diberi alasan. Saya tidak tahu kapan paket bisa diterima

Priority *
High

Status *
Done

Assign CS *
Customer Service

Back

2025 © UrsaByte. Design & Develop by UrsaByte Team

Gambar 5.37 Implementasi Antar Muka Halaman Detail Ticket Recap

Deskripsi

Gambar di atas menampilkan halaman detail ticket recap, yang muncul setelah *Customer service* menekan tombol 'View' pada daftar ticket. Pada halaman ini, sistem menampilkan informasi lengkap mengenai ticket, termasuk ID ticket, deskripsi kendala, prioritas, status, serta data pelanggan terkait.

5.2. Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan suatu proses untuk menguji atau memastikan program yang telah dibuat sudah sesuai dengan yang diharapkan, dengan pengujian perangkat lunak ini dapat mengetahui kualitas perangkat lunak tersebut. Salah satu cara untuk melakukan pengujian perangkat lunak yaitu dengan menggunakan black box testing.

Tabel 5.41 Pengujian Perangkat Lunak

Test Scenario	Test Case	Test Step	Test Data	Expected Result	Actual Result	Pass / Fail
TS-01 Melakukan Login pada website <i>Customer</i>	TC-1.1 Masuk dengan akun yang benar	<ol style="list-style-type: none"> Buka aplikasi "website <i>Customer</i>" Pada halaman login, masukkan email dan password yang valid Ketuk tombol "Login" 	Email : adawiyahajr@gmail.com Password : adawiyah123	Setelah menekan tombol "Login", pengguna berhasil masuk kedalam halaman beranda aplikasi.	Setelah menekan tombol "Login", pengguna berhasil masuk kedalam halaman beranda aplikasi.	Pass
	TC-1.2 Masuk dengan akun yang belum terdaftar	<ol style="list-style-type: none"> Buka aplikasi "website <i>Customer</i>" Pada halaman login, masukkan email dan password yang tidak valid Ketuk tombol "Login" 	Email : adwvhajriah@gmail.com Password : 123456	Pengguna menerima pesan bahwa "Login failed. Please check your email and password." dan tetap di halaman "Login"	Pengguna menerima pesan bahwa "Login failed. Please check your email and password." dan tetap di halaman "Login"	Pass
	TC-1.3 Masuk dengan email yang benar tetapi	<ol style="list-style-type: none"> Buka aplikasi "website <i>Customer</i>" Pada halaman login, 	Email : adawiyahajr@gmail.com Password : 123456	Pengguna menerima pesan bahwa "Login failed. Please check your email and password."	Pengguna menerima pesan bahwa "Login failed. Please check your email and password."	Pass

Test Scenario	Test Case	Test Step	Test Data	Expected Result	Actual Result	Pass / Fail
	password salah	<p>masukkan email benar tetapi password yang tidak valid</p> <p>3. Ketuk tombol "Login"</p>		dan tetap di halaman "Login"	dan tetap di halaman "Login"	
	TC-1.4 Masuk dengan email yang salah tetapi password benar	<p>1. Buka aplikasi "website Customer"</p> <p>2. Pada halaman login, masukkan email tidak valid dan password valid</p> <p>3. Ketuk tombol "Login"</p>	<p>Email : adwyhajriah@gmail.com</p> <p>Password : adawiyah123</p>	Pengguna menerima pesan bahwa "Login failed. Please check your email and password." dan tetap di halaman "Login"	Pengguna menerima pesan bahwa "Login failed. Please check your email and password." dan tetap di halaman "Login"	Pass
	TC-1.5 Masuk tanpa menggunakan email dan password	<p>1. Buka aplikasi "website Customer"</p> <p>2. Pada halaman login, biarkan email dan password kosong</p>	<p>Email : -</p> <p>Password : -</p>	Pengguna menerima pesan bahwa kolom email dan password wajib diisi	Pengguna menerima pesan bahwa kolom email dan password wajib diisi	Pass
	TC-1.6 Masuk menggunakan email dan password dengan role pengguna yang berbeda	<p>1. Buka aplikasi "website Customer"</p> <p>2. Pada halaman login, masukkan email dan password yang tidak</p>	<p>Email : admin123@gmail.com</p> <p>Password : 123456</p>	Pengguna menerima pesan "Access Denied. Only users with the Customer role are allowed to log in here." dan tetap berada di halaman Login.	Pengguna menerima pesan "Access Denied. Only users with the Customer role are allowed to log in here." dan tetap berada di halaman Login.	Pass

Test Scenario	Test Case	Test Step	Test Data	Expected Result	Actual Result	Pass / Fail
		sesuai dengan role 3. Ketuk tombol "Login"				
TS-02 Melakukan Login pada website admin	TC-2.1 Masuk dengan akun yang benar	1. Buka aplikasi "website admin" 2. Pada halaman login, masukkan email dan password yang valid 3. Ketuk tombol "Login"	Email : admin123@gmail.com Password : 123456	Setelah menekan tombol "Login", pengguna berhasil masuk kedalam halaman beranda aplikasi.	Setelah menekan tombol "Login", pengguna berhasil masuk kedalam halaman beranda aplikasi.	Pass
	TC-2.2 Masuk dengan akun yang belum terdaftar	1. Buka aplikasi "website admin" 2. Pada halaman login, masukkan email dan password yang tidak valid 3. Ketuk tombol "Login"	Email : admin@gmail.com Password : 123456	Pengguna menerima pesan bahwa "Login failed. Please check your email and password." dan tetap di halaman "Login"	Pengguna menerima pesan bahwa "Login failed. Please check your email and password." dan tetap di halaman "Login"	Pass

Test Scenario	Test Case	Test Step	Test Data	Expected Result	Actual Result	Pass / Fail
	TC-2.3 Masuk dengan email yang benar tetapi password salah	<ol style="list-style-type: none"> Buka aplikasi "website admin" Pada halaman login, masukkan email benar tetapi password yang tidak valid Ketuk tombol "Login" 	Email : admin123@gmail.com Password : admin123	Pengguna menerima pesan bahwa "Login failed. Please check your email and password." dan tetap di halaman "Login"	Pengguna menerima pesan bahwa "Login failed. Please check your email and password." dan tetap di halaman "Login"	Pass
	TC-2.4 Masuk dengan email yang salah tetapi password benar	<ol style="list-style-type: none"> Buka aplikasi "website admin" Pada halaman login, masukkan email tidak valid dan password valid Ketuk tombol "Login" 	Email : admin@gmail.com Password : 123456	Pengguna menerima pesan bahwa "Login failed. Please check your email and password." dan tetap di halaman "Login"	Pengguna menerima pesan bahwa "Login failed. Please check your email and password." dan tetap di halaman "Login"	Pass
	TC-2.5 Masuk tanpa menggunakan email dan password	<ol style="list-style-type: none"> Buka aplikasi "website admin" Pada halaman login, biarkan email dan password kosong Ketik tombol "Login" 	Email : - Password : -	Pengguna menerima pesan bahwa kolom email dan password wajib diisi	Pengguna menerima pesan bahwa kolom email dan password wajib diisi	Pass

Test Scenario	Test Case	Test Step	Test Data	Expected Result	Actual Result	Pass / Fail
	TC-2.6 Masuk menggunakan email dan password dengan role pengguna yang berbeda	<ol style="list-style-type: none"> 1. Buka aplikasi "website admin" 2. Pada halaman login, masukkan email dan password yang tidak sesuai dengan role 3. Ketuk tombol "Login" 	Email : adawiyahair@gmail.com Password : adawiyah123	Pengguna menerima pesan "Access Denied. Only users with the Admin role are allowed to log in here." dan tetap berada di halaman Login.	Pengguna menerima pesan bahwa "Login failed. Please check your email and password." dan tetap di halaman "Login"	Pass
TS-03 Melakukan Login pada website <i>Customer service</i>	TC-3.1 Masuk dengan akun yang benar	<ol style="list-style-type: none"> 1. Buka aplikasi "website <i>Customer service</i>" 2. Pada halaman login, masukkan email dan password yang valid 3. Ketuk tombol "Login" 	Email : Customerservice@gmail.com Password : 123456	Setelah menekan tombol "Login", pengguna berhasil masuk kedalam halaman beranda aplikasi.	Setelah menekan tombol "Login", pengguna berhasil masuk kedalam halaman beranda aplikasi.	Pass
	TC-3.2 Masuk dengan akun yang belum terdaftar	<ol style="list-style-type: none"> 1. Buka aplikasi "website <i>Customer service</i>" 2. Pada halaman login, 	Email : cs@gmail.com Password : 123456	Pengguna menerima pesan bahwa "Login failed. Please check your email and password." dan tetap di	Pengguna menerima pesan bahwa "Login failed. Please check your email and password." dan tetap di	Pass

Test Scenario	Test Case	Test Step	Test Data	Expected Result	Actual Result	Pass / Fail
		<p>masukkan email dan password yang tidak valid</p> <p>3. Ketuk tombol "Login"</p>		halaman "Login"	halaman "Login"	
	TC-3.3 Masuk dengan email yang benar tetapi password salah	<p>1. Buka aplikasi "website Customer"</p> <p>2. Pada halaman login, masukkan email benar tetapi password yang tidak valid</p> <p>3. Ketuk tombol "Login"</p>	<p>Email : Customerservice@gmail.com</p> <p>Password : Customerservice123</p>	<p>Pengguna menerima pesan bahwa "Login failed. Please check your email and password." dan tetap di halaman "Login"</p>	<p>Pengguna menerima pesan bahwa "Login failed. Please check your email and password." dan tetap di halaman "Login"</p>	Pass
	TC-3.4 Masuk dengan email yang salah tetapi password benar	<p>1. Buka aplikasi "website Customer service"</p> <p>2. Pada halaman login, masukkan email tidak valid dan password valid</p> <p>3. Ketuk tombol "Login"</p>	<p>Email : adwyhajriah@gmail.com</p> <p>Password : 123456</p>	<p>Pengguna menerima pesan bahwa "Login failed. Please check your email and password." dan tetap di halaman "Login"</p>	<p>Pengguna menerima pesan bahwa "Login failed. Please check your email and password." dan tetap di halaman "Login"</p>	Pass

Test Scenario	Test Case	Test Step	Test Data	Expected Result	Actual Result	Pass / Fail
	TC-3.5 Masuk tanpa menggunakan email dan password	<ol style="list-style-type: none"> 1. Buka aplikasi "website <i>Customer service</i>" 2. Pada halaman login, biarkan email dan password kosong 3. Ketuk tombol "Login" 	Email : - Password : -	Pengguna menerima pesan bahwa kolom email dan password wajib diisi	Pengguna menerima pesan bahwa kolom email dan password wajib diisi	Pass
	TC-3.6 Masuk menggunakan email dan password dengan role pengguna yang berbeda	<ol style="list-style-type: none"> 1. Buka aplikasi "website <i>Customer service</i>" 2. Pada halaman login, masukkan email dan password yang tidak sesuai dengan role 3. Ketuk tombol "Login" 	Email : adawiyahajr@gmail.com Password : 123456	Pengguna menerima pesan "Access Denied. Only users with the <i>Customer service</i> role are allowed to log in here." dan tetap berada di halaman Login.	Pengguna menerima pesan "Access Denied. Only users with the <i>Customer service</i> role are allowed to log in here." dan tetap berada di halaman Login.	Pass
TS-04 Fitur live chat	TC-4.1 <i>Customer</i> mengirim pesan ke <i>Customer service</i>	<ol style="list-style-type: none"> 1. Buka aplikasi "website <i>Customer</i>" 2. Klik icon <i>Customer service care</i> 3. Klik pertanyaan bantuan 	Pesan : Bagaimana cara klaim jika barang rusak?	<i>Customer</i> langsung menerima pesan balasan bot yaitu "Klaim bisa diajukan maksimal 1x24 jam setelah barang diterima." dan tetap berada di halaman chat.	<i>Customer</i> langsung menerima pesan balasan bot yaitu "Klaim bisa diajukan maksimal 1x24 jam setelah barang diterima." dan	Pass

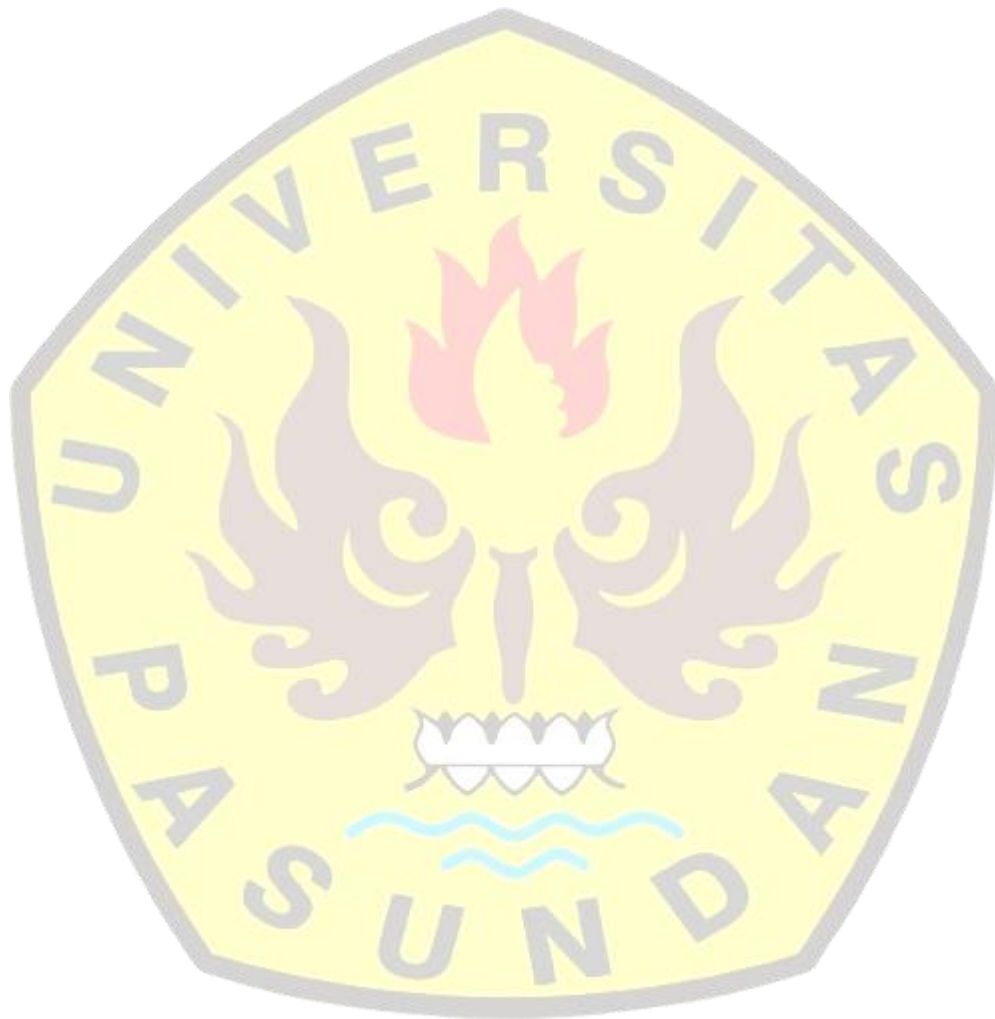
Test Scenario	Test Case	Test Step	Test Data	Expected Result	Actual Result	Pass / Fail
					tetap berada di halaman chat.	
	TC-4.2 <i>Customer service</i> menerima dan membalas pesan	<ol style="list-style-type: none"> Buka aplikasi "website <i>Customer service</i>" Klik menu chat untuk menampilkan daftar <i>Customer</i> yang melakukan chat. Klik <i>Customer</i> yang akan dibalas pesannya 	Klik chat dari adawiyahajr	<i>Customer service</i> menerima pesan dari adawiyahajr dan membalas pesan kepada adawiyahajr "halo ada yang bisa dibantu? Jika ada pertanyaan klik tanya hal lain pada pertanyaan bantuan"	<i>Customer service</i> menerima pesan dari adawiyahajr dan membalas pesan kepada adawiyahajr "halo ada yang bisa dibantu? Jika ada pertanyaan klik tanya hal lain pada pertanyaan bantuan"	Pass
	TC-4.3 <i>Customer</i> menerima pesan balasan dari <i>Customer service</i>	<ol style="list-style-type: none"> Buka aplikasi "website <i>Customer</i>" Klik icon <i>Customer service care</i> 	-	Pesan balasan muncul secara <i>Real-time</i> di chat <i>Customer</i>	Pesan balasan muncul secara <i>Real-time</i> di chat <i>Customer</i>	Pass
	TC-4.4 Tidak bisa mengirim pesan kosong	<ol style="list-style-type: none"> Buka aplikasi "website <i>Customer</i> atau <i>Customer service</i>" Klik icon <i>Customer service care</i> atau menu chat 	-	Pengguna tidak dapat mengirim pesan kosong karena tombol 'Send' dinonaktifkan ketika input pesan kosong.	Pengguna tidak dapat mengirim pesan kosong karena tombol 'Send' dinonaktifkan ketika input pesan kosong.	Pass

Test Scenario	Test Case	Test Step	Test Data	Expected Result	Actual Result	Pass / Fail
TS-05 Membuat Ticket	TC-5.1 <i>Customer</i> menanyakan kendala yang akan menjadi ticket	<ol style="list-style-type: none"> Buka aplikasi "website <i>Customer</i>" Klik icon <i>Customer service care</i> Klik pertanyaan bantuan "Tanya hal lain" Isi formulir ticket Submit a ticket 	ID Resi : REG089 Problem description : Saya menunggu paket yang seharusnya datang hari ini, tapi sampai sekarang belum ada kabar. Tracking tidak update, padahal saya membutuhkannya segera	Sistem menampilkan konfirmasi bahwa tiket berhasil dibuat dan tiket muncul di daftar tiket Admin beserta notifikasi terkait.	Sistem menampilkan konfirmasi bahwa tiket berhasil dibuat dan tiket muncul di daftar tiket Admin beserta notifikasi terkait.	Pass
	TC-5.2 Admin melihat ticket baru	<ol style="list-style-type: none"> Buka aplikasi "website admin" Buka menu ticket list Cek ticket yang masuk bisa menggunakan notification 	-	Sistem menampilkan tiket baru pada notifikasi dan menampilkan data tiket tersebut pada daftar tiket.	Sistem menampilkan tiket baru pada notifikasi dan menampilkan data tiket tersebut pada daftar tiket..	Pass
	TC-5.3 Admin menindaklanjuti ticket	<ol style="list-style-type: none"> Buka aplikasi "website admin" Buka menu list ticket dan pilih ticket Klik edit pada action Set priority dan assign cs Klik save 	Priority : High Assign CS : <i>Customer service</i>	Tiket berhasil diperbarui dan ditugaskan kepada <i>Customer service</i> yang bersangkutan.	Tiket berhasil diperbarui dan ditugaskan kepada <i>Customer service</i> yang bersangkutan.	Pass

Test Scenario	Test Case	Test Step	Test Data	Expected Result	Actual Result	Pass / Fail
	TC-5.4 <i>Customer service</i> menangani masalah sesuai dengan ticket yg di assign oleh admin	<ol style="list-style-type: none"> Buka aplikasi "website <i>Customer service</i>" Buka menu ticket list klik edit pada action set status menjadi in progress klik save 	Status : In Progress	Tiket berhasil diperbarui dan status tercatat sebagai "In Progress".	Tiket berhasil diperbarui dan status tercatat sebagai "In Progress".	Pass
	TC-5.5 <i>Customer service</i> menutup ticket	<ol style="list-style-type: none"> Buka aplikasi "website <i>Customer service</i>" Buka menu ticket list klik edit pada action set status menjadi done klik save 	Status : Done	Tiket berhasil diperbarui dan ticket yang sudah selesai akan tampil di menu ticket recap	Tiket berhasil diperbarui dan ticket yang sudah selesai akan tampil di menu ticket recap	Pass
	TC-5.6 <i>Customer service</i> mengakhiri sesi live chat setelah layanan selesai	<ol style="list-style-type: none"> Buka aplikasi "Website <i>Customer service</i>" Buka sesi live chat yang sedang berlangsung Pastikan proses layanan telah selesai Klik tombol End Chat 	Status Chat : Ended	Sistem berhasil mengakhiri sesi live chat dan menampilkan pesan otomatis sebagai penutup komunikasi kepada pengguna.	Sistem berhasil mengakhiri sesi live chat dan menampilkan pesan otomatis sebagai penutup komunikasi kepada pengguna.	Pass

Test Scenario	Test Case	Test Step	Test Data	Expected Result	Actual Result	Pass / Fail
	TC-5.7 Tidak bisa mengirim ticket tanpa id resi dan problem description	<ol style="list-style-type: none"> Buka aplikasi "website Customer" Klik icon Customer service care Klik pertanyaan bantuan "Tanya hal lain" Submit a ticket 	ID Resi : - Problem Description : -	Sistem menampilkan pesan validasi bahwa kolom ID Resi dan Problem Description wajib diisi sebelum proses dapat dilanjutkan.	Sistem menampilkan pesan validasi bahwa kolom ID Resi dan Problem Description wajib diisi sebelum proses dapat dilanjutkan.	Pass

Berdasarkan pengujian perangkat lunak pada tabel 5.44 menyajikan hasil pengujian perangkat lunak pada sistem "*Pengembangan Aplikasi Fitur Live Chat dan Sistem Ticketing untuk Pengiriman Barang Menggunakan Framework Angular (Studi Kasus: PT. UrsaByte)*". Pengujian ini dilakukan untuk memastikan setiap fitur utama sistem berjalan sesuai spesifikasi, termasuk: login untuk berbagai peran pengguna, pengiriman pesan melalui *live chat*, serta pembuatan dan penanganan tiket. Tabel ini menunjukkan skenario pengujian (*Test Scenario*), kasus pengujian (*Test Case*), langkah-langkah pengujian (*Test Step*), data yang digunakan (*Test Data*), hasil yang diharapkan (*Expected Result*), hasil aktual (*Actual Result*), dan status keberhasilan (*Pass / Fail*).



BAB 6 PENUTUP

6.1. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, maka dapat ditarik beberapa kesimpulan sebagai berikut:

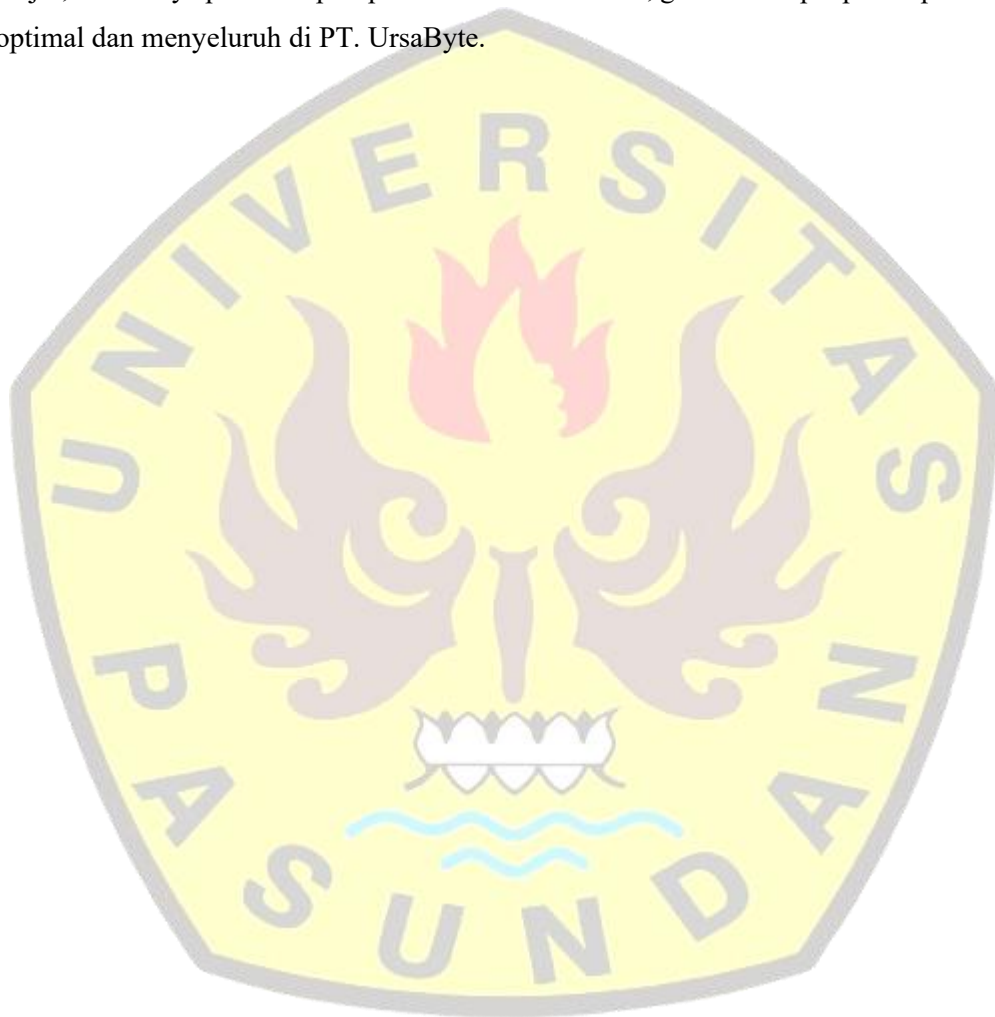
1. Penelitian ini berhasil mengembangkan aplikasi layanan live chat dan sistem ticketing pada PT. UrsaByte yang mampu menyediakan sarana komunikasi *Real-time* antara *Customer* dan *Customer service* dalam menangani kendala pengiriman barang secara lebih efektif.
2. Sistem yang dibangun mampu mencatat setiap keluhan pelanggan ke dalam bentuk tiket, menampilkan status serta tingkat prioritas penanganan, serta memastikan bahwa setiap tiket hanya ditangani oleh satu petugas guna menghindari terjadinya tumpang tindih proses penyelesaian.
3. Implementasi mekanisme multi-login dan autentikasi berbasis peran telah berjalan sesuai dengan perancangan, sehingga setiap pengguna dapat mengakses sistem berdasarkan hak dan tanggung jawabnya masing-masing.
4. Perancangan antarmuka pengguna menggunakan framework *Angular*, pengembangan backend dengan *Node.js*, serta integrasi *Firebase* sebagai basis data *Real-time* telah berhasil diimplementasikan sesuai dengan ruang lingkup penelitian yang ditetapkan.
5. Hasil pengujian sistem menunjukkan bahwa fitur utama, yaitu live chat, autentikasi pengguna, dan sistem ticketing, dapat berjalan sesuai dengan hasil pengujian dan memenuhi kebutuhan fungsional yang diharapkan.

6.2. Saran

Berdasarkan tahapan yang telah direalisasikan dalam penelitian tugas akhir ini, penulis memberikan beberapa saran yang dapat digunakan sebagai acuan dalam pengembangan sistem pada penelitian selanjutnya sebagai berikut :

1. Menambahkan fitur dashboard analitik yang mampu menyajikan data kinerja layanan, pola keluhan pengguna, serta tingkat penyelesaian tiket secara komprehensif.
2. Mengimplementasikan sistem notifikasi multi-channel, seperti email, WhatsApp, atau push notification, guna meningkatkan keterjangkauan informasi kepada pengguna.
3. Mengembangkan manajemen ID resi secara terintegrasi dengan sistem pengiriman barang, sehingga aplikasi tidak hanya menggunakan ID resi sebagai informasi pendukung, tetapi juga mampu melakukan pelacakan dan pengelolaan status pengiriman secara otomatis.

4. Menambahkan fitur penanda status ketersediaan online dan offline pada *Customer service* dan pengguna, sehingga kedua belah pihak dapat saling mengetahui kondisi kehadiran secara *real-time*. Penerapan fitur ini diharapkan dapat meningkatkan efektivitas komunikasi, mempercepat waktu respons layanan, serta mendukung interaksi live chat yang lebih efisien dan terkoordinasi.
5. Penelitian ini masih memiliki keterbatasan, karena pengembangan sistem hanya dilakukan hingga tahap construction, sehingga proses deployment dan evaluasi pemanfaatan sistem secara langsung di lingkungan operasional perusahaan belum dapat dilaksanakan.
6. Sistem yang dikembangkan masih memiliki peluang untuk dilakukan pengembangan lebih lanjut, khususnya pada tahap implementasi dan evaluasi, guna mencapai penerapan sistem yang optimal dan menyeluruh di PT. UrsaByte.



DAFTAR PUSTAKA

- [ALV23] Lius Alviando, A. B. “*Penerapan Websocket pada Sistem Live Chat berbasis Web (Studi Kasus Website Kwikku.com)*”, Vol. 7, No. 2, Februari 2023
- [DAY18] Brad Dayley, B. D. “*In Learning Angular: A Hands-On Guide to Angular 2 and Angular 4*”. Pearson Education, Inc., United States of America, 2018
- [DIA22] Diaconu, A. “*The WebSocket Handbook: Learn about the technology underpinning the realtime web and build your first web app powered by WebSockets*”, Februari 2022.
- [FAR22] Farhan, A. “*Rancang Bangun Aplikasi Pengelolaan Layanan Ticketing Helpdesk Berbasis Web (Studi Kasus : PT. Ussi Bandung)*”, Februari 2022
- [FAU21] Sri Tita Faulina, N. L. “*Penerapan Metode Waterfall pada Aplikasi Pemesanan Soundsystem Dan Organ Tunggal Jefri*”, JIK, VOL. 12, No. 2, Desember 2021
- [FEB23] Febryntara, I. G. “*Rancang Bangun Sistem Layanan Live Chat Berbasis Website pada Politeknik Negeri Bali Menggunakan Framework Laravel*”, 2023
- [HEN19] Hendrokusumo, R. “*Rancang Bangun Aplikasi Pengiriman Barang pada CV.Prima Semesta Internusa Berbasis Android*”, 2019.
- [ISK23] Ade Rahmat Iskandar, Y. S. “*Pemrograman Laravel. Informatika Bandung*”, Informatika Bandung, November 2023
- [KAS16] Rio A. Kasengkang, S. N. “*Analisis Logistik (Studi Kasus pada PT. Remenia Satori Tepas Kota Manado)*”, Volume 16 No. 01 Tahun 2016
- [KHA24] Ananda Salsabila Khairi, M. A. *Implementasi Role Based Access Control dalam Pengelolaan Hak Akses Koperasi Berbasis Mobile, Volume 09 Nomor 01, Juni 2024*
- [KHO25] Haykal Kholid Kz, M. S. “*Pembuatan Website BUMDes Berbasis Opensid di Desa Wano untuk Meningkatkan Pengetahuan Masyarakat*”, 2025
- [KIR16] Wahyu Kartika Candra Kirana, Z. Z. “*Pemanfaatan Directive pada Framework Angular untuk Pengembangan Website Penerimaan Mahasiswa Baru*”, 2016

- [MAN20] Esther Sanda Manapa, E. A. “*Review of Real-time Operating System Research in Indonesia*”, 2020.
- [NOV22] Noviana, R. “*Pembuatan Aplikasi Penjualan Berbasis Web Monja Store Menggunakan PHP dan MYSQL*”, *Jurnal Teknik dan Science*, P-ISSN: 2828-7002 E-ISSN: 2828-6871, Juni 2022
- [NUR19] Muhamad Nurudin, W. J. “*Pengujian Black Box pada Aplikasi Penjualan Berbasis Web Menggunakan Teknik Boundary Value Analysis*”, Desember 2019
- [PER24] Edwar Permana, A. S. “*Pengembangan Aplikasi Live Chat Menggunakan Extreme Programing*”, Mei 2024
- [PRE20] Roger S. Pressman, B. R. “*Software Engineering: A Practitioner’s Approach ninth edition*”, 2020
- [PUR22] Ratih Purbasari, N. J. (n.d.). “*Digitalisasi Logistik Dalam Mendukung Kinerja E-Logistic Di Era Digital: A Literature Review*”, Vol 01, No.02 177-196, 2022
- [QOR25] Lamis F. Al-Qora’n, A. A.-S. A. *Modular Monolith Architecture in Cloud Environments: A Systematic Literature Review*, 29 Oktober 2025
- [RAH22] Marwanto Rahmatuloh, M. R. “*Rancang Bangun Sistem Informasi Jasa Pengiriman Barang Pada PT. Haluan Indah Transporindo Berbasis Web*”, *Jurnal Teknik Informatika*, Vol. 14, No. 1, Januari 2022
- [RIZ23] Muhamad Rizki, A. F. “*Aplikasi Chatbot Sebagai Layanan Live Chat untuk Penerimaan Mahasiswa Baru Menggunakan Metode Word Stemming dengan Regular Expression Pattern Matching*”, Volume 03 Nomor 02, Juli 2023
- [ROS20] Rosyana Fitria Purnomo, O. W. “*Firestore Membangun Aplikasi Berbasis Android*”, 2020
- [SEN24] R. Mochammad Armand Giri Seno, A. J. “*Pengembangan Fitur Chat Pada Website PT. Danatrans Service Logistics*”, Juli 2024
- [SUS16] Susanto, Y. “*Transportasi dalam Rantai Pasok dan Logistik*”, November 2016
- [WAH20] Wahid, A. A. “*Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi*”, *Jurnal Ilmu-ilmu Informatika dan Manajemen STMIK*, Oktober 2020
- [YUR23] Yuricha, I. K. P. *Penerapan Role Based Access Control dalam Sistem Supply Chain Management Berbasis Cloud*, Vol. 3 Iss. 2 October 2023, pp: 339-348

[ZEI23] Drs. Afrizal zein M.Kom, I. D. “*Konsep Dasar Rekayasa Perangkat Lunak*”,
Yayasan Cendikia Mulia Mandiri, April 2023



LAMPIRAN



Form Wawancara

Identifikasi Masalah

Penelitian Aplikasi Fitur Live Chat Pada Sistem Pengiriman Barang

Menggunakan Framework *Angular* (Studi Kasus : PT. UrsaByte)

Telah dilaksanakan wawancara berkaitan dengan penelitian yg akan dilakukan untuk memenuhi tugas akhir strata satu (S1). Dengan ini menerangkan bahwa mahasiswa yang bertanda tangan dibawah ini :

1. Pewawancara

Nama Lengkap : Adawiyah Ajriah Lubis
Perguruan Tinggi : Universitas Pasundan Bandung
Fakultas : Fakultas Teknik
Jurusan : Teknik Informatika
Nrp : 213040054

2. Narasumber

No	Nama Lengkap	Jabatan
1.	Abiyan Bagus Baskoro, S.Kom	Software Developer

3. Tempat Wawancara

Tempat: Online Via Chat
Waktu: 23 April 2025

Pewawancara melakukan sesi tanya jawab dengan narasumber terkait penelitian yang akan dilakukan. Narasumber memberikan jawaban sesuai dengan pertanyaan yang diajukan. Adapun pertanyaan dan hasil wawancara tercantum dibawah ini :

No	Pertanyaan	Jawaban
1.	Untuk proses pengiriman barang melalui website tersebut apakah barang akan langsung dikirimkan ke <i>Customer service</i> atau ke Perusahaan?	Website ini berfungsi untuk melakukan tracking terhadap barang-barang yang dikirimkan oleh <i>Customer</i>
2.	Bagaimana alur kerja dari tiket tersebut, apakah fitur ini digunakan oleh <i>Customer</i> untuk melaporkan permasalahan secara langsung?	Proses pembuatan tiket dilakukan oleh admin. <i>Customer</i> yang mengalami kendala akan melaporkan terlebih dahulu kepada <i>Customer service</i> , kemudian informasi tersebut diteruskan kepada admin untuk dibuatkan tiket. Selanjutnya, tiket tersebut akan ditindaklanjuti oleh bagian yang sesuai dengan tugas masing-masing
3.	Jika tiket berfungsi untuk mencatat permasalahan mengapa tiket tersebut dibuat oleh admin bukan oleh <i>Customer</i> langsung?	Tiket dibuat oleh admin untuk menginformasikan adanya permasalahan, baik yang berasal dari system maupun dari lapangan. Pembuatan oleh admin bertujuan agar terdapat satu peran saja, sebelum akhirnya didistribusikan ke masing-masing role tergantung laporannya
4.	Dalam proses penyampaian keluhan, bagaimana cara pelanggan menyampaikan permasalahan yang dialaminya?	<i>Customer</i> menyampaikan permasalahan yang dialaminya melalui via telepon atau dnegan bertemu secara langsung.
5.	Bagaimana sistem pencatatan keluhan saat ini dan apakah sudah terintegrasi dengan komunikasi <i>Customer</i> ?	Saat ini keluhan dicatat oleh admin setelah <i>Customer</i> melapor ke CS, sehingga prosesnya masih manual dan belum terintegrasi secara real-time.
6.	Apakah fitur live chat tersedia saat ini untuk memfasilitasi komunikasi <i>Customer</i> dengan CS?	Fitur live chat belum tersedia saat ini pelanggan masih melaporkan permasalahan melalui <i>Customer service</i> secara langsung, baik melalui telepon maupun pertemuan tatap muka

Mengetahui,

Peneliti



Adawiyah Ajriah Lubis

Software Developer



Abiyah Bagus Baskoro, S.Kom



Berita Acara

Pengujian Aplikasi

Penelitian Aplikasi Fitur Live Chat Pada Sistem Pengiriman Barang

Menggunakan Framework *Angular* (Studi Kasus : PT. UrsaByte)

No	Deskripsi Pengujian	Pass	fail
1.	Customer melakukan login menggunakan akun yang valid	✓	
2.	Customer melakukan percobaan login menggunakan akun yang belum terdaftar	✓	
3.	Customer login dengan email yang benar namun menggunakan password yang tidak sesuai	✓	
4.	Customer login dengan email yang tidak valid namun menggunakan password yang benar	✓	
5.	Customer melakukan login tanpa mengisi kolom email dan password	✓	
6.	Customer login menggunakan akun dengan role yang tidak sesuai	✓	
7.	admin melakukan login menggunakan akun yang valid	✓	
8.	Admin melakukan percobaan login menggunakan akun yang belum terdaftar	✓	
9.	Admin login dengan email yang benar namun menggunakan password yang tidak sesuai	✓	
10.	Admin login dengan email yang tidak valid namun menggunakan password yang benar	✓	
11.	Admin melakukan login tanpa mengisi kolom email dan password	✓	
12.	Admin login menggunakan akun dengan role yang tidak sesuai	✓	
13.	Customer service melakukan login menggunakan akun yang valid	✓	
14.	Customer Service melakukan percobaan login menggunakan akun yang belum terdaftar	✓	
15.	Customer Service login dengan email yang benar namun menggunakan password yang tidak sesuai	✓	
16.	Customer Service login dengan email yang tidak valid namun menggunakan password yang benar	✓	
17.	Customer Service melakukan login tanpa mengisi kolom email dan password	✓	
18.	Customer Service login menggunakan akun dengan role yang tidak sesuai	✓	
19.	Customer mengirimkan pesan melalui fitur Live Chat	✓	

20.	Customer Service menerima dan memberikan balasan terhadap pesan yang dikirim oleh Customer	✓	
21.	Customer menerima balasan pesan dari Customer Service	✓	
22.	Sistem tidak bisa mengirim pesan apabila kolom pesan kosong pada fitur Live Chat	✓	
23.	Sistem menolak pembuatan tiket apabila ID resi atau deskripsi masalah tidak diisi	✓	
24.	Customer membuat tiket dengan pengisian ID resi dan deskripsi masalah secara lengkap	✓	
25.	Admin menerima notifikasi tiket baru dan tiket tercatat secara otomatis pada daftar tiket	✓	
26.	Admin menindaklanjuti tiket dengan menentukan prioritas dan menetapkan Customer Service yang bertanggung jawab	✓	
27.	Customer Service menangani tiket sesuai tugas yang diberikan dan berkomunikasi dengan Customer, serta memperbarui status tiket menjadi "In Progress".	✓	
28.	Customer Service menutup tiket setelah penanganan selesai, dengan status diperbarui menjadi "Done"	✓	
29.	Customer Service mengakhiri sesi komunikasi live chat dengan mengklik <i>End Chat</i> setelah proses layanan selesai.	✓	

Tanggal pengujian aplikasi

Bandung, 8 januari 2026

Menyetujui,



(Juli Murwanto, A.Md.Kom)

Dokumentasi Pengujian Sistem

The screenshot displays a web application interface for a ticketing system. The browser address bar shows 'localhost:4201/admin/master/ticket/list'. The application has a red sidebar with navigation options: 'HELPDESK', 'Tickets', 'Ticket List', 'Chat History', 'Ticket Receipt', and 'Users'. The main content area is titled 'TICKETS' and contains a table with the following data:

No	No Ticket	ID Real	Description	Priority	Status	Actions
1	TCT-20200105-8472E	OK3302	Saya membutuhkan informasi terkait prosedur pengambilan barang setelah pengiriman selesai.	Low	New	[Icons]
2	TCT-20200105-C2001	YES560	Nomor resi yang saya terima untuk melakukan pelacakan pengiriman, Mohon dibantu pengecekan.	High	New	[Icons]
3	TCT-20200105-43932	REG785	Berdasarkan informasi pada sistem, paket dinyatakan telah diterima, namun pada kenyataannya saya belum menerima paket tersebut.	High	New	[Icons]

Below the table, it says 'Showing 1 to 4 of 4 entries'. The interface also includes a search bar and pagination controls. On the right side, there is a video conference window with two participants: 'adawiyah ajriah 213040054' (represented by a purple square with the letter 'a') and 'Juli Murwanto' (represented by a green square with a profile picture). The bottom of the screen shows a system tray with the time '17:59' and the user 'sih-vhpa-hyn', along with various system icons and a meeting control bar.











