

BAB II

KAJIAN LITERATUR

2.1 Sistem Informasi

Sistem informasi merupakan suatu sistem yang dirancang untuk mengelola data sehingga dapat menghasilkan informasi yang berguna bagi pengguna. Sistem informasi biasanya dikembangkan untuk menggantikan atau memperbaiki sistem lama yang masih memiliki berbagai permasalahan. Dengan adanya sistem informasi, diharapkan dapat meningkatkan efisiensi dan efektivitas dalam pengolahan data serta penyampaian informasi kepada pengguna [10].

Perancangan sistem informasi dilakukan melalui proses pengembangan sistem yang bertujuan untuk menghasilkan sistem yang mampu mendukung kebutuhan organisasi. Sistem yang dikembangkan diharapkan dapat mengatasi berbagai kendala yang terdapat pada sistem sebelumnya serta mampu memberikan solusi yang lebih baik dalam pengolahan dan penyampaian informasi. Oleh karena itu, pengembangan sistem informasi menjadi bagian penting dalam meningkatkan kinerja suatu organisasi [10].

Sistem informasi memiliki beberapa fungsi utama dalam mendukung aktivitas organisasi. Fungsi tersebut berkaitan dengan pengolahan data hingga menjadi informasi yang bermanfaat bagi pengguna. Berikut merupakan fungsi sistem informasi [11] :

1. Mengumpulkan data dari berbagai sumber yang berkaitan dengan aktivitas organisasi
2. Mengolah data menjadi informasi yang memiliki nilai guna
3. Menyimpan data dan informasi agar dapat digunakan kembali.
4. Menyebarkan informasi kepada pihak yang membutuhkan secara tepat waktu.
5. Mendukung proses pengambilan keputusan dalam organisasi.

Fungsi-fungsi tersebut menunjukkan bahwa sistem informasi memiliki peran penting dalam meningkatkan efisiensi kerja serta membantu organisasi dalam mencapai tujuan yang diinginkan.



Gambar 2. 1 Komponen Sistem Informasi [12]

Selain memiliki fungsi, sistem informasi juga terdiri dari beberapa komponen utama. Komponen-komponen sistem informasi dapat dilihat pada Gambar 2.1 yaitu [11] :

1. Perangkat keras (*hardware*), yaitu perangkat fisik yang digunakan untuk menjalankan sistem informasi.
2. Perangkat lunak (*software*), yaitu program yang digunakan untuk mengolah data menjadi informasi.
3. Data, yaitu bahan dasar yang akan diproses menjadi informasi.
4. Manusia (*brainware*), yaitu pengguna yang mengoperasikan sistem informasi.
5. Prosedur, yaitu langkah-langkah dalam pengolahan data agar menghasilkan informasi yang sesuai

Pada penelitian ini, konsep sistem informasi digunakan sebagai dasar dalam pengembangan sistem *monitoring* status perbaikan berbasis *web* pada Advance Service Center Medan. Sistem informasi digunakan pada tahap pengelolaan data perbaikan, pemantauan status perbaikan, serta penyampaian informasi kepada pelanggan melalui notifikasi *WhatsApp*. Dengan demikian, sistem yang dikembangkan diharapkan mampu meningkatkan efisiensi pelayanan serta mempermudah pengguna dalam memperoleh informasi terkait status perbaikan.

2.2 *Monitoring*

Monitoring merupakan suatu kegiatan yang dilakukan untuk menjamin tercapainya tujuan yang telah ditetapkan dalam suatu organisasi atau sistem. *Monitoring* dilakukan dengan cara mengamati, mengkaji, serta mengevaluasi pelaksanaan suatu kegiatan agar dapat diketahui apakah kegiatan tersebut berjalan sesuai dengan rencana yang telah ditentukan. *Monitoring* juga dapat diartikan sebagai proses untuk menilai dan mengukur pelaksanaan suatu kegiatan, serta mengidentifikasi permasalahan yang terjadi sehingga dapat segera dilakukan tindakan perbaikan. Dengan adanya *monitoring*, pihak yang berkepentingan dapat mengetahui tingkat keberhasilan suatu kegiatan serta melakukan evaluasi terhadap kinerja yang telah dilakukan [13].

Dalam penerapannya, *monitoring* sering digunakan dalam berbagai bidang, salah satunya dalam sistem informasi. Sistem *monitoring* memungkinkan pengguna untuk memperoleh informasi secara berkala maupun secara langsung terhadap suatu objek yang dipantau. Hal ini bertujuan agar pengguna dapat mengetahui kondisi atau status dari suatu objek secara cepat dan akurat [14]. Sistem *monitoring* juga dapat dimanfaatkan untuk memberikan informasi terkait lokasi atau kondisi suatu objek. Dengan memanfaatkan teknologi tertentu,

sistem *monitoring* mampu memberikan informasi berupa data lokasi maupun kondisi objek sehingga memudahkan pengguna dalam melakukan pengawasan [15].

Selain itu, sistem *monitoring* dapat membantu dalam meningkatkan efisiensi kerja, terutama dalam proses pengawasan dan pengambilan keputusan. Dengan adanya sistem *monitoring* berbasis teknologi, proses pengumpulan data dan penyampaian informasi dapat dilakukan secara lebih cepat, sehingga dapat mengurangi kesalahan serta meningkatkan keakuratan data yang diperoleh [14].

Berdasarkan penjelasan tersebut, konsep *monitoring* pada penelitian ini akan digunakan untuk memantau proses layanan atau perbaikan barang secara berkala. Sistem *monitoring* yang dikembangkan bertujuan untuk memberikan informasi terkait status barang, mulai dari barang masuk, proses pengecekan, hingga status akhir servis. Dengan adanya sistem ini, pengguna dapat mengetahui perkembangan kondisi barang secara lebih cepat dan akurat.

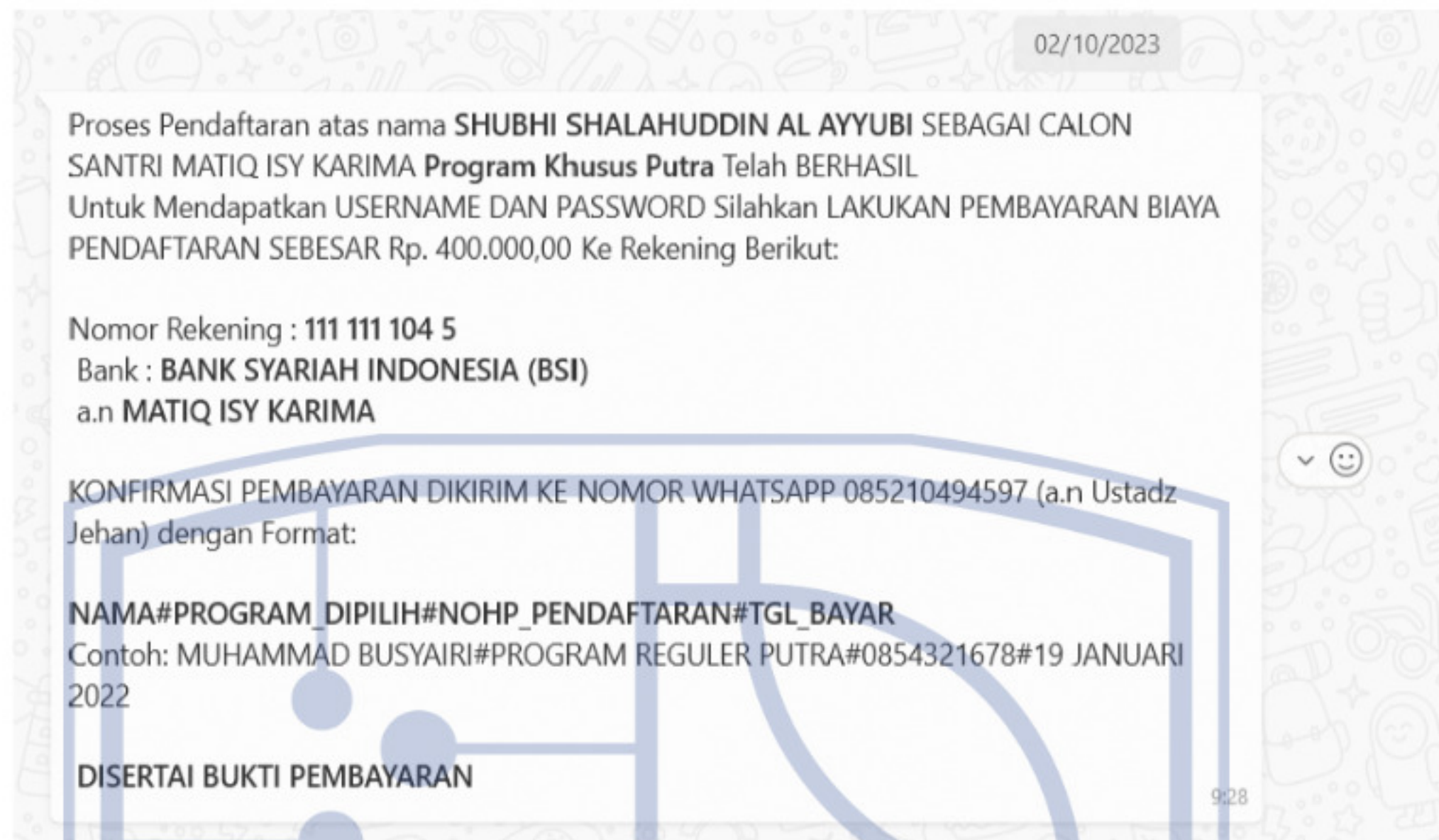
2.3 Notifikasi WhatsApp (*WhatsApp API*)

Notifikasi merupakan suatu bentuk penyampaian informasi kepada pengguna yang bertujuan untuk memberikan pemberitahuan terkait suatu aktivitas atau kejadian dalam sistem. Dalam perkembangan teknologi saat ini, penyampaian notifikasi dapat dilakukan secara otomatis melalui berbagai media komunikasi, salah satunya adalah aplikasi *WhatsApp*. *WhatsApp* merupakan aplikasi perpesanan instan yang memungkinkan pengguna untuk mengirim dan menerima pesan melalui jaringan internet tanpa menggunakan pulsa, serta dapat digunakan sebagai media penyebaran informasi secara cepat dan luas [16].

Dalam implementasinya pada sistem informasi, pengiriman pesan *WhatsApp* dapat dilakukan secara otomatis dengan memanfaatkan *WhatsApp API*. *WhatsApp API* merupakan suatu teknologi yang memungkinkan sistem untuk mengirim pesan secara otomatis kepada pengguna tanpa harus dilakukan secara manual melalui aplikasi *WhatsApp*. Penggunaan *WhatsApp API* memungkinkan sistem untuk terhubung langsung dengan *database* sehingga pesan dapat dikirim secara cepat, tepat, dan dalam jumlah banyak kepada pengguna [16].

Selain itu, penggunaan *WhatsApp Gateway* sebagai bagian dari implementasi *API* memungkinkan integrasi sistem dengan layanan pesan *WhatsApp* sehingga dapat mendukung pengiriman pesan otomatis, balasan otomatis, serta pelacakan pesan yang telah dikirim [17]. Penerapan notifikasi *WhatsApp* dalam sistem informasi memberikan berbagai manfaat, di antaranya meningkatkan efektivitas komunikasi, mempercepat penyampaian informasi kepada pengguna, serta membantu pengguna dalam memperoleh informasi secara langsung tanpa harus membuka sistem secara manual [18]. Contoh notifikasi WhatsApp

yang dikirimkan kepada pengguna dapat dilihat pada Gambar 2.2. Berdasarkan Gambar 2.2, notifikasi *WhatsApp* berisi informasi yang dikirim secara otomatis kepada pengguna terkait suatu aktivitas dalam sistem, seperti pemberitahuan pendaftaran atau status layanan.



Gambar 2. 2 Contoh Notifikasi WhatsApp [17]

Pemilihan *WhatsApp* sebagai media notifikasi didasarkan pada efektivitasnya sebagai sarana komunikasi digital yang telah luas digunakan masyarakat Indonesia. *WhatsApp Gateway* merupakan pengembangan dari *SMS Gateway* yang lebih sesuai dengan kebiasaan masyarakat *modern* karena pengguna cenderung menggunakan paket data internet dibanding pulsa reguler, sehingga layanan notifikasi melalui *WhatsApp* lebih mudah diakses, praktis, dan efisien dalam penyampaian informasi otomatis kepada pelanggan. Penggunaan *WhatsApp* juga memungkinkan informasi diterima lebih cepat tanpa mengharuskan pelanggan datang langsung ke lokasi layanan [19]. Dalam penelitian ini, implementasi notifikasi *WhatsApp* dilakukan menggunakan layanan *WhatsApp Gateway* pihak ketiga, yaitu *Fonnte*. *Fonnte* merupakan platform *API* yang memungkinkan sistem mengirim pesan *WhatsApp* secara otomatis melalui integrasi dengan aplikasi *web* menggunakan HTTP request. Pemilihan *Fonnte* didasarkan pada kemudahan integrasi, dokumentasi yang sederhana, efisiensi biaya, serta kesesuaiannya dengan kebutuhan sistem *monitoring* status perbaikan. Dengan menggunakan *Fonnte*, sistem dapat mengirimkan notifikasi perubahan status perbaikan kepada pelanggan tanpa memerlukan pengelolaan infrastruktur *WhatsApp Business API* secara mandiri.

2.4 Aplikasi Berbasis Web

Aplikasi berbasis *web* merupakan suatu program atau perangkat lunak yang dapat diakses melalui jaringan internet dengan menggunakan *web browser*. Aplikasi ini memungkinkan pengguna untuk berinteraksi dengan sistem melalui antarmuka berbasis *web* sehingga dapat memproses data dan menghasilkan informasi sesuai kebutuhan pengguna [20]. Menurut Simarmata, aplikasi *web* merupakan suatu sistem informasi yang mampu mendukung interaksi pengguna melalui *interface* berbasis *web*. Aplikasi *web* dirancang untuk membantu pengguna dalam mengolah data serta mempermudah dalam memperoleh informasi yang dibutuhkan.[21]. Aplikasi berbasis *web* bekerja dengan konsep *client-server*, dimana pengguna mengakses sistem melalui *browser* yang kemudian berkomunikasi dengan *server* untuk memproses data. Alur kerja aplikasi berbasis *web* dapat dilihat pada Gambar 2.3.



Gambar 2. 3 Arsitektur Aplikasi Berbasis Web [22].

Aplikasi berbasis *web* memiliki keunggulan dibandingkan dengan aplikasi desktop, yaitu dapat diakses kapan saja dan dimana saja selama terhubung dengan jaringan internet. Selain itu, aplikasi berbasis *web* tidak memerlukan proses instalasi pada perangkat pengguna karena sistem telah terpasang pada *server* dan dapat digunakan melalui berbagai sistem operasi seperti *Windows*, *Linux*, maupun *MacOS* [20]. Penggunaan aplikasi berbasis *web* dalam sistem informasi memberikan kemudahan dalam pengelolaan data serta meningkatkan efisiensi kerja. Hal ini dikarenakan proses pengolahan data yang sebelumnya dilakukan secara manual dapat dilakukan secara terkomputerisasi sehingga lebih cepat, akurat, dan terorganisir dengan baik [23].

Selain itu, sistem informasi berbasis *web* juga mampu membantu dalam penyampaian informasi secara lebih efektif dan efisien. Informasi yang dihasilkan dapat diakses secara langsung oleh pengguna sehingga mendukung proses pengambilan keputusan dalam suatu organisasi atau instansi [24]. Dengan demikian, aplikasi berbasis *web* memiliki peran penting dalam pengembangan sistem informasi, terutama dalam mendukung kemudahan

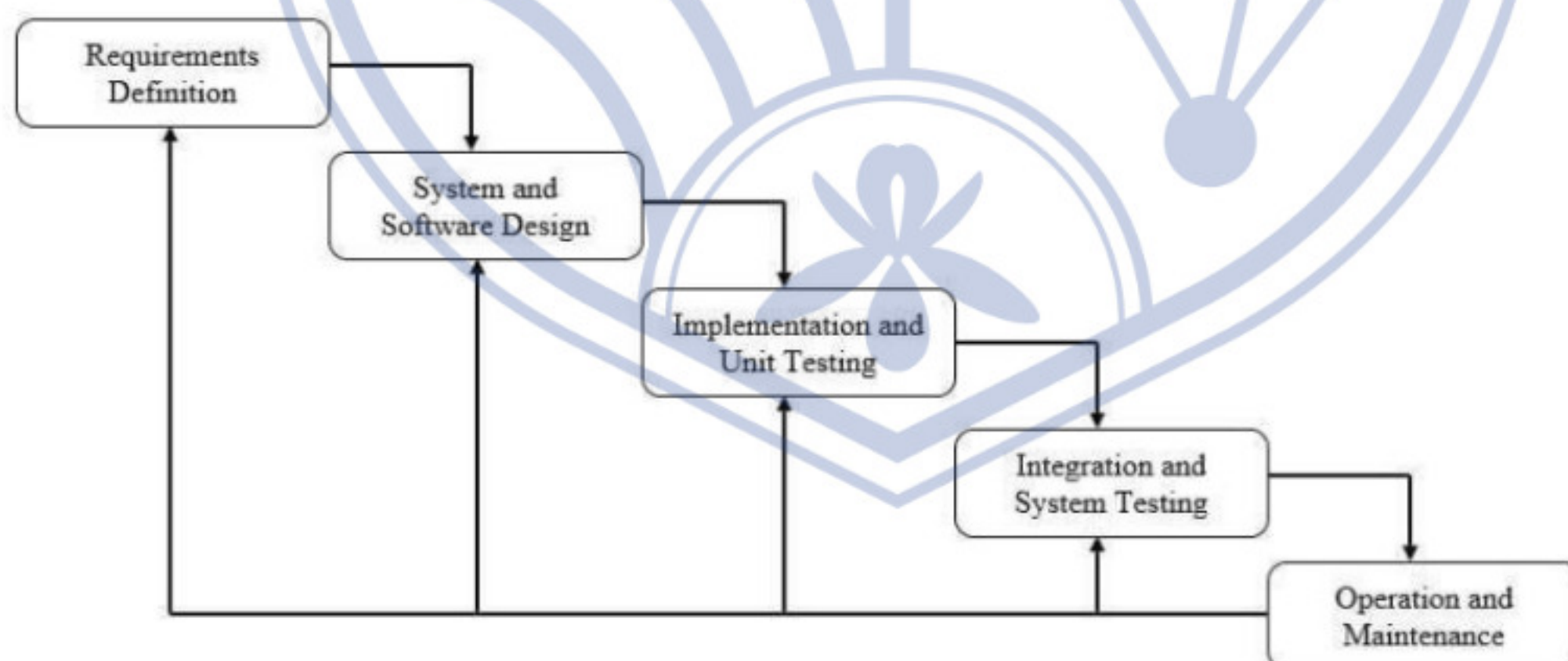
akses, efisiensi pengolahan data, serta penyampaian informasi yang cepat dan akurat kepada pengguna.

Pada penelitian ini, aplikasi berbasis *web* digunakan sebagai media dalam pengembangan sistem *monitoring* status perbaikan pada Advance Service Center Medan. Sistem yang dibangun memungkinkan pengguna untuk mengakses informasi status perbaikan secara online tanpa harus datang langsung ke lokasi. Dengan adanya aplikasi berbasis *web* ini, proses penyampaian informasi menjadi lebih cepat, efisien, serta dapat diakses kapan saja dan dimana saja oleh pelanggan.

2.5 Metode *Waterfall*

Metode *Waterfall* merupakan salah satu metode dalam *Software Development Life Cycle (SDLC)* yang digunakan dalam pengembangan sistem informasi secara sistematis dan berurutan. Metode ini dikenal sebagai model sekuensial linier, dimana setiap tahapan harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya [25]. Model *Waterfall* merupakan model klasik yang banyak digunakan dalam pengembangan perangkat lunak karena memiliki alur yang terstruktur dan terdokumentasi dengan baik.

Setiap tahapan dalam metode ini saling berkaitan, dimana hasil dari satu tahap akan menjadi masukan bagi tahap selanjutnya [26]. Dengan pendekatan yang terstruktur, metode ini memudahkan pengembang dalam melakukan perencanaan dan pengendalian proses pengembangan sistem. Oleh karena itu, metode *Waterfall* cocok digunakan pada pengembangan sistem yang memiliki kebutuhan yang jelas dan tidak sering mengalami perubahan.



Gambar 2. 4 Metode *Waterfall* [23]

Berdasarkan Gambar 2.4, terlihat bahwa setiap tahapan dalam metode *Waterfall* dilakukan secara bertahap dan sistematis. Adapun tahapan-tahapan dalam metode *Waterfall* adalah sebagai berikut [23]:

1. Analisis Kebutuhan (*Requirement Analysis*)

Tahap analisis kebutuhan merupakan tahap awal dalam pengembangan sistem yang bertujuan untuk mengidentifikasi kebutuhan pengguna secara menyeluruh. Proses ini dilakukan melalui observasi, wawancara, serta studi pustaka untuk memperoleh informasi yang akurat. Hasil dari tahap ini akan menjadi dasar dalam perancangan sistem pada tahap selanjutnya.

2. Perancangan Sistem (*System Design*)

Tahap perancangan sistem dilakukan berdasarkan hasil analisis kebutuhan yang telah diperoleh. Pada tahap ini dilakukan perancangan struktur sistem, *database*, serta antarmuka sistem yang akan digunakan. Perancangan ini bertujuan untuk memberikan gambaran menyeluruh mengenai sistem yang akan dikembangkan.

3. Implementasi (*Implementation*)

Tahap implementasi merupakan proses penerjemahan desain sistem ke dalam bentuk kode program sehingga sistem dapat dijalankan sesuai dengan kebutuhan yang telah dirancang sebelumnya.

4. Pengujian (*Testing*)

Tahap pengujian dilakukan untuk memastikan bahwa sistem yang telah dibangun dapat berjalan dengan baik serta sesuai dengan kebutuhan pengguna. Proses ini dilakukan untuk menemukan kesalahan atau *bug* pada sistem sebelum digunakan.

5. Pemeliharaan (*Maintenance*)

Tahap pemeliharaan dilakukan untuk memperbaiki kesalahan yang tidak ditemukan pada tahap sebelumnya serta melakukan pengembangan lebih lanjut terhadap sistem sesuai kebutuhan pengguna.

Pada penelitian ini, metode *Waterfall* digunakan sebagai acuan dalam proses pengembangan sistem *monitoring* status perbaikan secara terstruktur dan sistematis. Namun, tahapan yang dilakukan hanya sampai pada tahap pengujian (*testing*) dan tidak mencakup tahap pemeliharaan (*maintenance*). Hal ini disebabkan karena penelitian berfokus pada pengembangan sistem dalam skala terbatas serta adanya keterbatasan waktu penelitian, sementara tahap pemeliharaan umumnya dilakukan setelah sistem digunakan dalam jangka waktu tertentu dan memerlukan evaluasi lanjutan terhadap performa sistem. Penerapan metode *Waterfall* dalam penelitian ini meliputi tahap analisis kebutuhan untuk mengidentifikasi permasalahan serta kebutuhan sistem terkait *monitoring* perbaikan dan pengiriman notifikasi kepada pelanggan, tahap perancangan sistem untuk menyusun *use case diagram*, *database*, dan tampilan antarmuka, tahap implementasi untuk membangun sistem sesuai dengan desain yang telah dirancang, serta tahap pengujian untuk memastikan bahwa seluruh fitur, seperti

pembaruan status perbaikan dan pengiriman notifikasi *WhatsApp*, dapat berjalan dengan baik. Dengan menggunakan metode *Waterfall*, proses pengembangan sistem menjadi lebih terarah dan terdokumentasi secara sistematis.

2.6 *Unified Modelling Language (UML)*

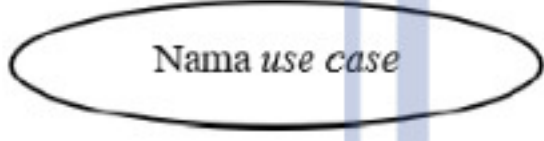
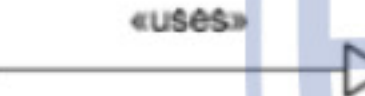
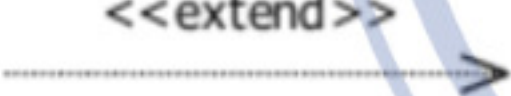
Unified Modeling Language (UML) merupakan suatu bahasa pemodelan yang digunakan untuk merancang, memvisualisasikan, serta mendokumentasikan sistem perangkat lunak. *UML* digunakan sebagai alat bantu dalam proses perancangan sistem agar struktur, alur, serta interaksi dalam sistem dapat digambarkan secara jelas dan mudah dipahami oleh pengembang maupun pihak terkait [27]. *UML* memiliki berbagai jenis diagram yang dapat digunakan untuk menggambarkan sistem dari berbagai sudut pandang, seperti struktur sistem, perilaku sistem, serta interaksi antar komponen dalam sistem. Dalam *UML* versi terbaru terdapat berbagai macam diagram, namun dalam pengembangan sistem informasi umumnya digunakan beberapa diagram utama seperti *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram* [28]. Penggunaan *UML* dalam perancangan sistem sangat penting karena dapat membantu dalam memahami kebutuhan sistem secara lebih terstruktur, mengurangi kesalahan dalam pengembangan, serta mempermudah komunikasi antara pengembang dengan pengguna sistem. Dengan adanya *UML*, proses pengembangan sistem menjadi lebih sistematis dan terarah. Pada penelitian ini, *UML* digunakan sebagai alat bantu dalam merancang sistem *monitoring* status perbaikan berbasis *web* pada Advance Service Center Medan. Diagram yang digunakan dalam penelitian ini meliputi *Use Case Diagram* dan *Activity Diagram* untuk menggambarkan interaksi pengguna serta alur proses dalam sistem.

2.6.1 *Use Case Diagram*

Use Case Diagram merupakan salah satu diagram dalam *Unified Modeling Language (UML)* yang digunakan untuk menggambarkan hubungan antara aktor (pengguna) dengan sistem serta fungsi-fungsi yang tersedia dalam sistem. Diagram ini menunjukkan bagaimana pengguna berinteraksi dengan sistem serta layanan apa saja yang dapat diakses oleh pengguna [27]. *Use Case Diagram* berfungsi untuk memberikan gambaran umum mengenai sistem dari sudut pandang pengguna. Dalam diagram ini, aktor digambarkan sebagai pihak yang berinteraksi dengan sistem, sedangkan *use case* menggambarkan aktivitas atau fungsi yang dapat dilakukan oleh aktor tersebut. Dengan adanya diagram ini, pengembang dapat memahami kebutuhan sistem secara lebih jelas serta mengetahui interaksi antara pengguna dan sistem sebelum sistem diimplementasikan [28]. Dalam *Use Case Diagram*

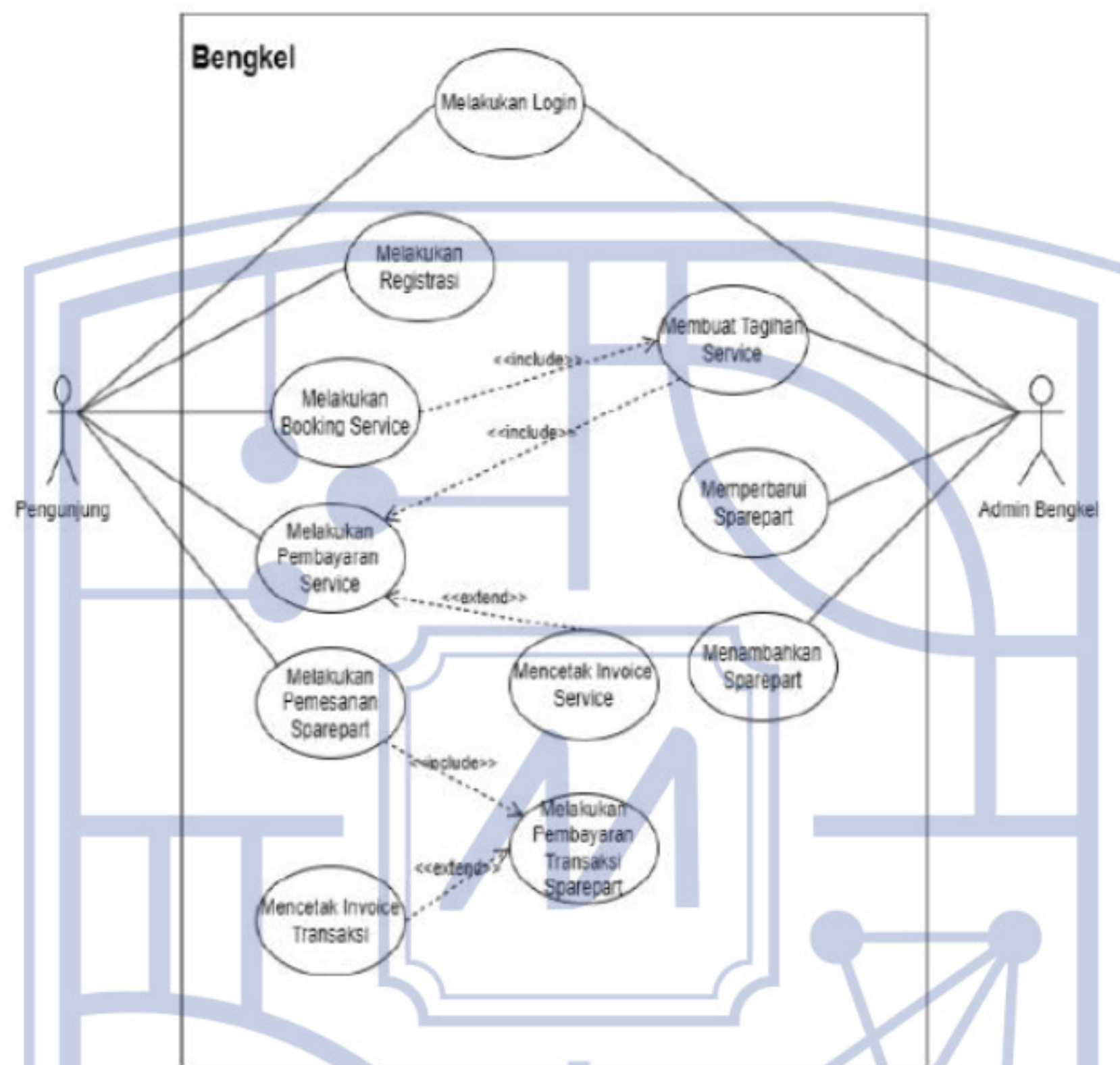
terdapat beberapa komponen utama yang digunakan, berikut penjelasan mengenai komponen *Use Case* dapat dilihat pada tabel 2.1.

Tabel 2. 1 Komponen *Use Case Diagram* [29]

Simbol	Nama	Penjelasan
	Aktor (<i>Actor</i>)	Aktor adalah entitas di luar sistem yang berinteraksi langsung dengan sistem, bisa berupa manusia atau sistem lain. Aktor merepresentasikan peran, bukan individu.
	<i>Use Case</i>	<i>Use case</i> adalah fungsionalitas atau layanan yang disediakan oleh sistem untuk aktor. Biasanya ditulis dalam bentuk kata kerja yang menggambarkan aktivitas.
	Asosiasi (<i>Association</i>)	Hubungan antara aktor dengan <i>use case</i> yang menunjukkan adanya interaksi atau komunikasi antara keduanya.
 	<i>Include (Uses)</i>	Relasi yang menunjukkan bahwa suatu <i>use case</i> selalu membutuhkan <i>use case</i> lain sebagai bagian dari prosesnya (wajib dijalankan).
	<i>Extend</i>	Relasi yang menunjukkan bahwa suatu <i>use case</i> dapat diperluas oleh <i>use case</i> lain dalam kondisi tertentu (opsional).
	<i>Generalization</i>	Hubungan pewarisan antara aktor atau <i>use case</i> , dimana satu elemen merupakan turunan dari elemen yang lebih umum.

Sebagai contoh, dapat dilihat dari gambar 2.5, *Use Case Diagram* menggambarkan interaksi antara dua aktor utama yaitu admin bengkel dan pengunjung. Admin bengkel merupakan pihak yang memiliki hak akses terhadap pengelolaan sistem, sedangkan pengunjung merupakan pengguna yang dapat mengakses layanan yang tersedia dalam sistem. *Use Case Diagram* tersebut digunakan sebagai dasar dalam pengembangan sistem, khususnya dalam menentukan kebutuhan fungsional sistem. Setiap *use case* yang terdapat pada diagram akan dijadikan acuan dalam perancangan fitur-fitur yang akan diimplementasikan pada sistem.

Pada diagram tersebut, admin bengkel dapat melakukan beberapa aktivitas seperti *login*, membuat tagihan perbaikan, memperbarui data *sparepart*, serta menambahkan data *sparepart*. Sementara itu, pengunjung dapat melakukan registrasi, *login*, melakukan pemesanan jasa perbaikan, melakukan pembayaran perbaikan, melakukan pemesanan *sparepart*, serta melakukan pembayaran dan mencetak tagihan transaksi



Gambar 2. 5 Use Case Diagram [30]

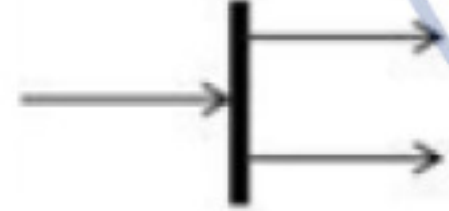
Pada penelitian ini, *Use Case Diagram* digunakan sebagai acuan dalam mengidentifikasi kebutuhan fungsional sistem *monitoring* status perbaikan serta menggambarkan interaksi antara pengguna dengan sistem yang akan dikembangkan. Selain itu, diagram ini juga membantu dalam memahami alur penggunaan sistem secara lebih jelas. Dengan demikian, proses perancangan sistem dapat dilakukan secara lebih terstruktur dan sistematis.

2.6.2 Activity Diagram

Activity Diagram merupakan salah satu diagram *UML* yang digunakan untuk menggambarkan alur aktivitas atau proses dalam suatu sistem. Diagram ini menunjukkan urutan kegiatan yang terjadi, mulai dari awal hingga akhir proses secara terstruktur sehingga memudahkan dalam memahami alur kerja sistem [27]. *Activity Diagram* berfungsi untuk memvisualisasikan proses bisnis atau alur kerja sistem secara lebih detail. Dalam diagram ini, setiap aktivitas digambarkan secara berurutan sehingga memudahkan dalam memahami bagaimana

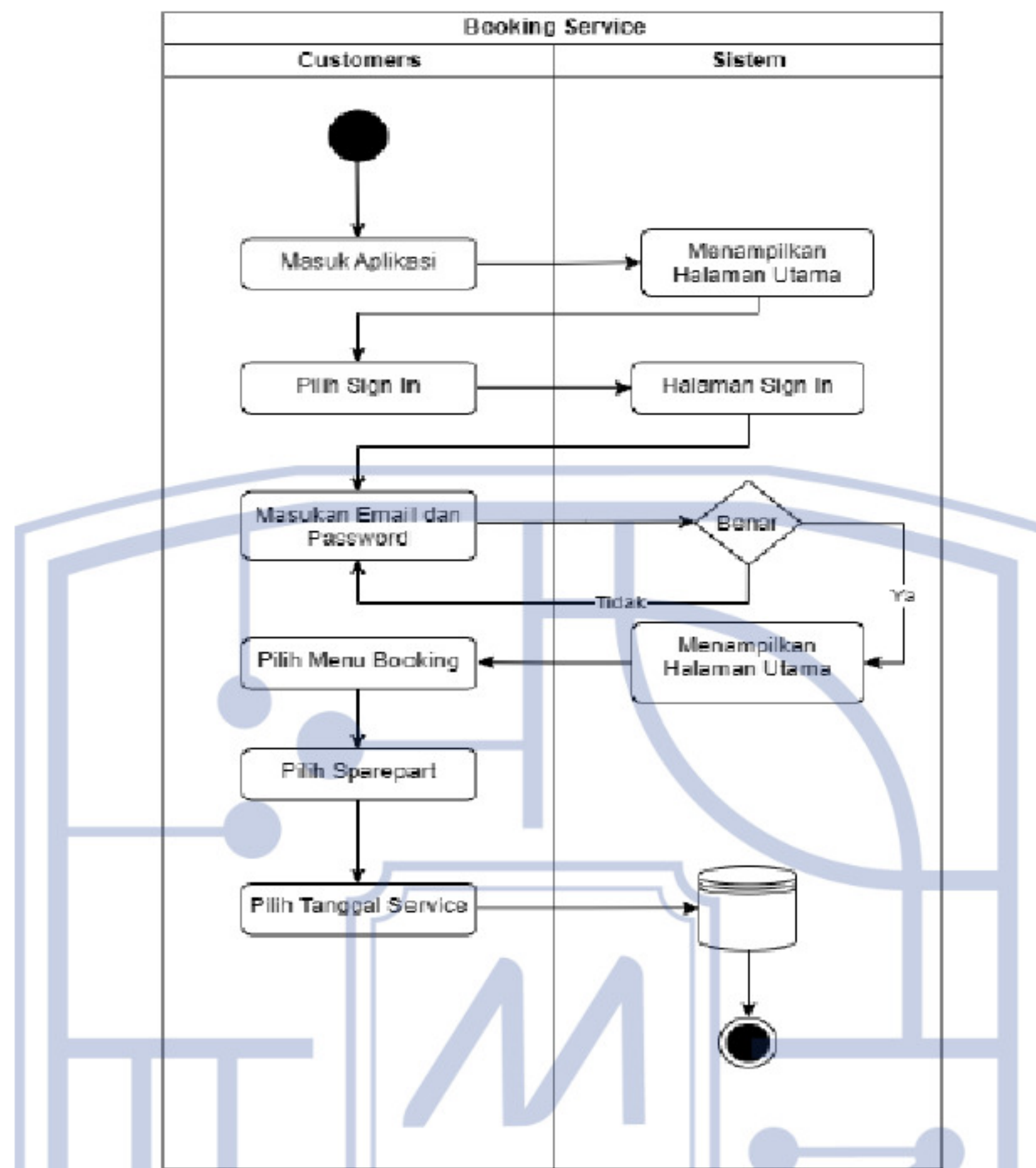
suatu proses berjalan dalam sistem [28]. Dengan menggunakan *Activity Diagram*, pengembang dapat mengetahui langkah-langkah yang terjadi dalam sistem serta hubungan antar aktivitas. Hal ini sangat membantu dalam menganalisis dan merancang sistem agar berjalan sesuai dengan kebutuhan pengguna. Dalam *Activity Diagram* terdapat beberapa komponen utama yang digunakan, berikut penjelasan mengenai komponen *Activity Diagram* dapat dilihat pada tabel 2.2

Tabel 2. 2 Komponen *Activity Diagram* [29]

Simbol	Nama	Penjelasan
	<i>Initial Node (Start)</i>	Menunjukkan titik awal dimulainya suatu proses atau aktivitas dalam sistem.
	<i>Final Node (End)</i>	Menunjukkan titik akhir dari suatu proses atau aktivitas.
	<i>Activity / Action</i>	Menunjukkan suatu aktivitas atau proses yang dilakukan dalam sistem.
	<i>Decision</i>	Digunakan untuk menunjukkan percabangan berdasarkan kondisi tertentu (ya/tidak).
	<i>Control Flow</i>	Menunjukkan aliran atau urutan proses dari satu aktivitas ke aktivitas lainnya.
	<i>Fork</i>	Digunakan untuk memecah satu alur aktivitas menjadi beberapa alur yang berjalan secara paralel (bersamaan).
	<i>Join (Penggabungan)</i>	Digunakan untuk menggabungkan beberapa alur aktivitas paralel menjadi satu alur kembali.

Sebagai contoh dapat dilihat pada gambar 2.6, pada sistem informasi layanan perbaikan bengkel berbasis *web* terdapat *Activity Diagram* yang digunakan, yaitu *Activity Diagram* untuk proses pemesanan jasa perbaikan. *Activity Diagram* pemesanan jasa perbaikan menggambarkan alur pengguna dalam melakukan pemesanan layanan mulai dari proses *login*

hingga memilih layanan yang diinginkan, diagram tersebut menunjukkan tahapan aktivitas yang dilakukan secara berurutan serta keputusan yang terjadi dalam setiap proses.



Gambar 2. 6 Activity Diagram Pemesanan Jasa Perbaikan [30]

Dengan demikian, *Activity Diagram* pada penelitian ini digunakan untuk menggambarkan alur proses dalam sistem *monitoring* status perbaikan serta sebagai acuan dalam perancangan dan pengembangan sistem yang akan dibangun. Diagram ini membantu peneliti dalam memahami urutan proses yang harus diimplementasikan serta hubungan antar aktivitas dalam sistem, sehingga setiap tahapan proses dapat dirancang secara jelas dan terstruktur. Selain itu, *Activity Diagram* juga digunakan sebagai dasar dalam menentukan kebutuhan fungsional sistem, sehingga proses perancangan sistem dapat dilakukan secara lebih sistematis dan sesuai dengan kebutuhan yang telah diidentifikasi.

2.7 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan suatu model atau diagram yang digunakan untuk merancang basis data dengan menggambarkan hubungan antar entitas dalam suatu sistem. *ERD* digunakan untuk memperlihatkan relasi antar objek atau data yang saling berhubungan sehingga memudahkan dalam memahami struktur data yang akan

dibangun [31]. Dengan menggunakan *ERD*, perancang sistem dapat menentukan tabel, atribut, serta hubungan antar data secara jelas sebelum sistem diimplementasikan.

ERD juga berfungsi sebagai alat untuk memodelkan kebutuhan data secara visual sehingga mempermudah dalam proses pengembangan sistem berbasis database. Dalam perancangannya, *ERD* menggambarkan entitas sebagai objek utama, atribut sebagai karakteristik dari entitas, serta hubungan (*relationship*) yang menghubungkan antar entitas [32]. Dengan adanya *ERD*, struktur database dapat dirancang secara lebih terorganisir dan mengurangi kesalahan dalam pengolahan data. Hal ini sejalan dengan penelitian yang menyatakan bahwa *ERD* mampu memberikan gambaran yang jelas mengenai struktur data serta hubungan antar entitas sehingga mempermudah dalam proses pengembangan sistem [33].

Dalam perancangan *Entity Relationship Diagram (ERD)*, terdapat beberapa komponen utama yang digunakan untuk menggambarkan struktur dan hubungan data dalam sistem. Komponen-komponen tersebut saling berhubungan satu sama lain sehingga membentuk suatu model database yang utuh dan mudah dipahami. Adapun komponen utama dalam *ERD* adalah sebagai berikut [32] :

1. Entitas, merupakan objek utama dalam sistem yang memiliki data dan akan disimpan dalam *database*. Entitas dapat berupa manusia, tempat, atau objek lain yang memiliki peran penting dalam sistem, seperti pelanggan, data perbaikan, atau teknisi.
2. Atribut, merupakan karakteristik atau informasi yang dimiliki oleh suatu entitas. Atribut digunakan untuk menjelaskan detail dari entitas, seperti nama, ID, alamat, atau tanggal.
3. Relasi, merupakan hubungan antar entitas yang menunjukkan keterkaitan antara satu entitas dengan entitas lainnya dalam sistem.
4. Kardinalitas, merupakan konsep yang menunjukkan jumlah hubungan antar entitas, seperti one-to-one, one-to-many, dan many-to-many.

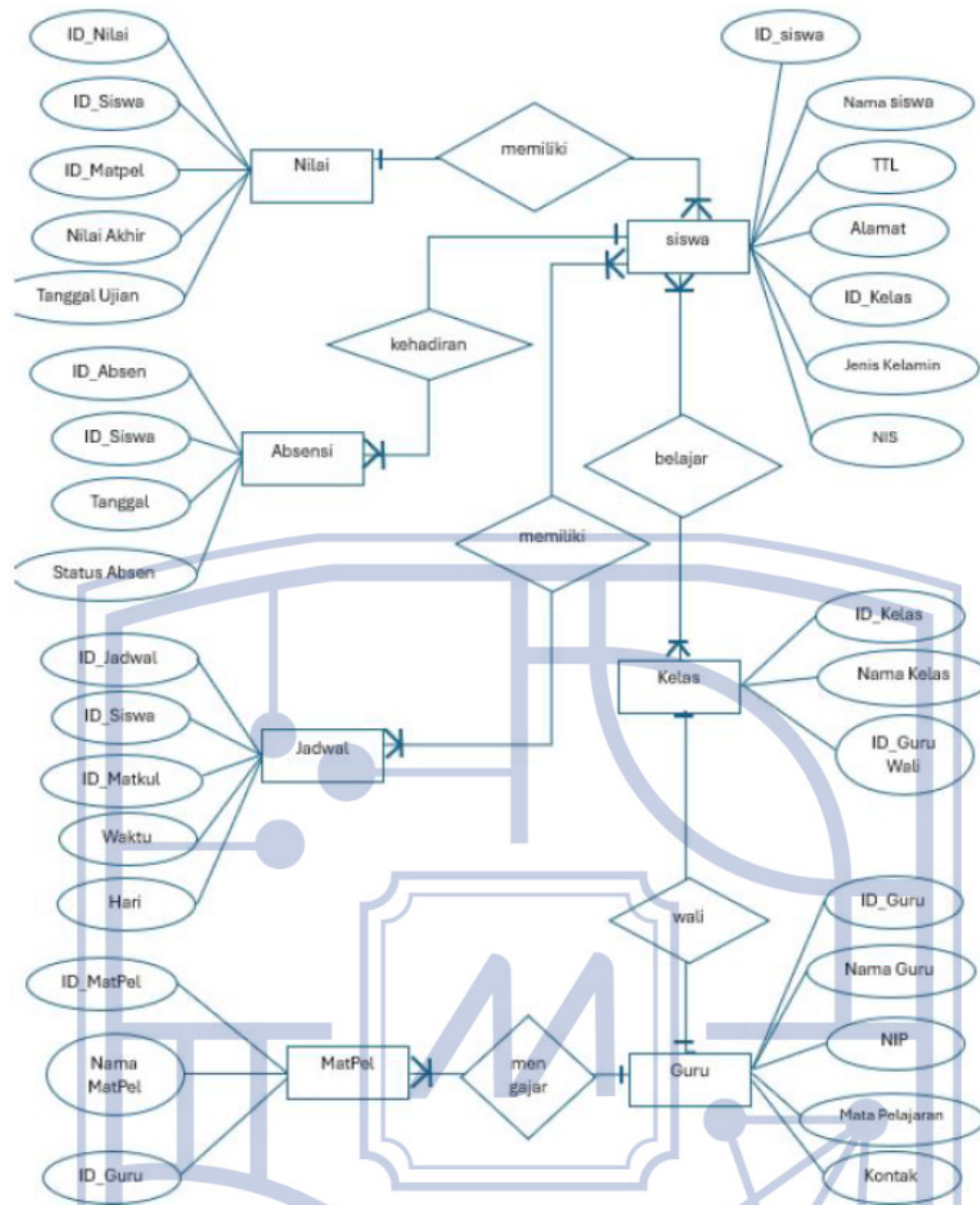
Penggunaan *ERD* dalam perancangan sistem informasi sangat penting karena dapat membantu dalam menggambarkan struktur *database* secara sistematis dan mudah dipahami. *ERD* digunakan sebagai dasar dalam pembuatan *database* sehingga hubungan antar tabel dapat dirancang dengan baik. Selain itu, *ERD* juga mempermudah pengembang dalam melakukan implementasi sistem karena struktur data sudah tergambar dengan jelas sejak awal perancangan [34].

Untuk memperjelas bentuk *Entity Relationship Diagram (ERD)*, simbol dasar yang digunakan dapat dilihat pada tabel 2.3.

Tabel 2. 3 Komponen *ERD* [35]

Simbol	Nama	Penjelasan
	Entitas (<i>Entity</i>)	Entitas adalah objek atau konsep yang dapat diidentifikasi dan memiliki data untuk disimpan dalam basis data. Biasanya direpresentasikan dalam bentuk persegi panjang dan diberi nama sesuai objeknya.
	Atribut (<i>Attribute</i>)	Atribut adalah karakteristik atau properti dari suatu entitas yang digunakan untuk mendeskripsikan entitas tersebut, seperti nama, alamat, atau ID.
	Relasi (<i>Relationship</i>)	Relasi adalah hubungan yang menghubungkan dua atau lebih entitas dalam suatu basis data. Relasi biasanya diberi nama sesuai hubungan yang terjadi.
	Garis (<i>Connector</i>)	Garis digunakan untuk menghubungkan entitas dengan atribut serta menghubungkan entitas dengan relasi dalam <i>ERD</i> .

Selain simbol dasar, contoh penerapan *ERD* dalam suatu sistem dapat dilihat pada Gambar 2.7 berikut. Berdasarkan Gambar 2.7, *Entity Relationship Diagram (ERD)* menggambarkan hubungan antar entitas yang terdapat dalam suatu sistem. Pada diagram tersebut, entitas ditunjukkan dengan bentuk persegi panjang, seperti entitas Siswa, Kelas, Guru, dan Mata Pelajaran. Setiap entitas memiliki atribut yang digambarkan dengan bentuk elips, seperti ID_siswa, Nama_siswa, ID_kelas, dan atribut lainnya yang menjelaskan karakteristik dari masing-masing entitas. Selain itu, hubungan antar entitas ditunjukkan dengan relasi yang digambarkan menggunakan bentuk belah ketupat, seperti relasi memiliki, belajar, wali, dan mengajar. Relasi tersebut menunjukkan keterkaitan antar entitas dalam sistem. Diagram ini juga menunjukkan adanya keterhubungan antar data yang disusun secara terstruktur sehingga memudahkan dalam memahami alur data dalam sistem.



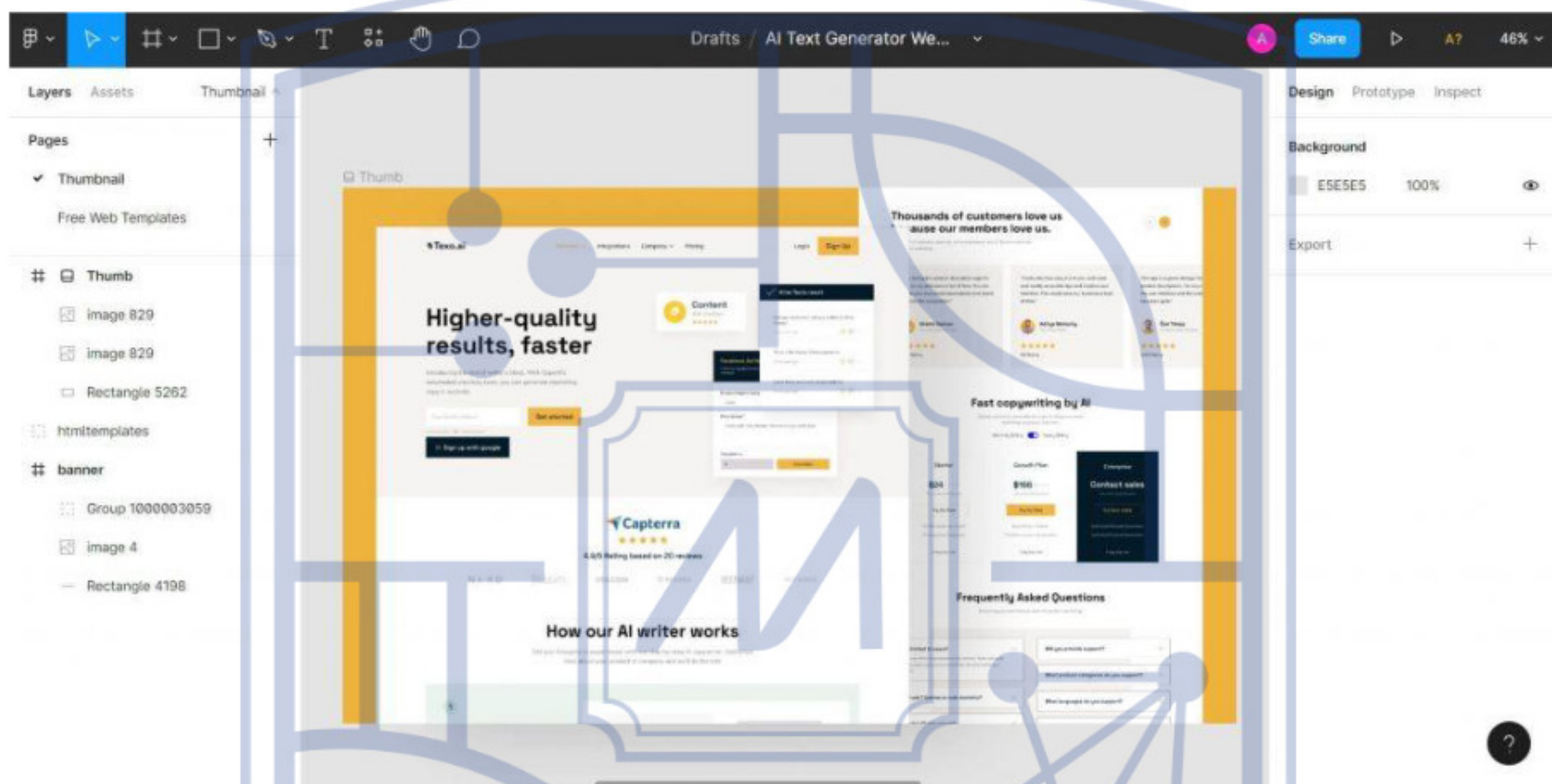
Gambar 2. 7 Contoh Penerapan ERD [33].

Pada penelitian ini, *ERD* digunakan untuk merancang struktur *database* pada sistem *monitoring* status perbaikan berbasis *web* di Advance Service Center Medan. *ERD* digunakan untuk menggambarkan hubungan antara entitas seperti data pelanggan, data perbaikan, serta status perbaikan yang saling berhubungan dalam sistem. Dengan adanya *ERD*, perancangan *database* menjadi lebih terstruktur sehingga sistem yang dikembangkan dapat berjalan dengan lebih efektif dan efisien.

2.8 Figma

Figma merupakan salah satu tools desain berbasis *cloud* yang digunakan untuk merancang tampilan antarmuka (*UI*) dan pengalaman pengguna (*UX*). Figma dapat digunakan secara langsung melalui *browser* tanpa perlu instalasi serta mendukung kolaborasi secara *real-time* antar tim dalam proses desain [36]. Selain itu, Figma juga banyak digunakan dalam pembuatan desain aplikasi maupun website karena kemudahan

penggunaannya serta fleksibilitas yang dimiliki. Figma memiliki berbagai fitur yang mendukung proses perancangan *UI/UX* secara efektif. Fitur utama yang dimiliki Figma antara lain pembuatan desain berbasis vektor, pembuatan wireframe, serta prototyping yang digunakan untuk mensimulasikan interaksi antar halaman sebelum sistem dikembangkan [37]. Selain itu, Figma juga memiliki fitur kolaborasi secara *real-time* yang memungkinkan beberapa pengguna bekerja dalam satu proyek secara bersamaan, serta fitur *handoff* yang memudahkan pengembang dalam mengimplementasikan desain ke dalam sistem [36]. Dengan adanya fitur-fitur tersebut, Figma mampu meningkatkan efisiensi dan kualitas dalam proses perancangan sistem. Tampilan antarmuka Figma dapat dilihat pada gambar 2.8.



Gambar 2. 8 Tampilan Antarmuka Figma [38]

Pada penelitian ini, Figma digunakan sebagai *tools* dalam merancang tampilan antarmuka sistem *monitoring* status perbaikan berbasis *web* pada Advance Service Center Medan. Proses perancangan dilakukan mulai dari pembuatan *wireframe*, *prototype*, hingga desain akhir sistem yang akan digunakan. Dengan menggunakan Figma, diharapkan tampilan sistem yang dihasilkan lebih menarik, mudah digunakan, serta sesuai dengan kebutuhan pengguna sehingga dapat meningkatkan kualitas sistem yang dikembangkan.

2.9 *PIECES*

PIECES merupakan suatu model analisis yang digunakan untuk mengidentifikasi permasalahan dalam sistem secara spesifik sehingga dapat membantu dalam menentukan kebutuhan sistem yang akan dikembangkan. Metode ini berfungsi untuk menemukan pokok permasalahan yang terdapat pada sistem yang sedang berjalan serta memberikan solusi dalam pengembangan sistem yang lebih baik. Model *PIECES* digunakan untuk menganalisis

sistem berdasarkan beberapa aspek penting guna meningkatkan kualitas sistem informasi. Model *PIECES* terdiri dari enam aspek utama yaitu [39].

1. *Performance* (Kinerja)

Kinerja sistem berkaitan dengan kemampuan sistem dalam memproses data dan menghasilkan *output*. Kinerja juga berperan dalam menilai kemungkinan peningkatan suatu prosedur serta mengawasi keandalan sistem dalam mencapai tujuan yang diinginkan. Dalam sistem yang masih berjalan secara manual, proses pencarian data dan pembuatan laporan sering kali membutuhkan waktu yang lama sehingga kurang efisien.

2. *Information and Data* (Informasi dan Data)

Aspek informasi berkaitan dengan kualitas informasi yang dihasilkan oleh sistem. Informasi digunakan untuk memberikan penilaian terhadap prosedur yang sedang berjalan serta menentukan apakah perlu dilakukan perbaikan atau peningkatan kualitas informasi. Sistem yang kurang baik biasanya menyebabkan terjadinya miskomunikasi serta informasi yang tidak akurat dan tidak tepat waktu

3. *Economics* (Ekonomi)

Aspek ekonomi digunakan untuk menilai biaya yang dikeluarkan serta manfaat yang diperoleh dari sistem. Dalam hal ini, sistem yang baik adalah sistem yang mampu menekan biaya operasional serta meningkatkan nilai guna dari sistem tersebut. Biaya operasional yang tinggi menunjukkan bahwa sistem belum berjalan secara optimal.

4. *Control and Security* (Pengendalian dan Keamanan)

Pengendalian berkaitan dengan kemampuan sistem dalam mengontrol dan mengawasi jalannya proses. Sistem harus mampu meminimalkan kesalahan serta mencegah terjadinya kecurangan. Pengendalian yang baik akan membantu dalam mendeteksi permasalahan serta meningkatkan kualitas sistem secara keseluruhan.

5. *Efficiency* (Efisiensi)

Efisiensi berkaitan dengan penggunaan sumber daya dalam menjalankan sistem. Sistem yang efisien mampu meningkatkan kinerja operasional dengan penggunaan waktu dan tenaga yang lebih minimal. Perbaikan sistem diperlukan agar proses yang berjalan menjadi lebih cepat dan efektif dibandingkan sistem sebelumnya.

6. *Service* (Pelayanan)

Aspek pelayanan berkaitan dengan kemampuan sistem dalam memberikan layanan kepada pengguna. Sistem yang baik harus mampu meningkatkan kualitas pelayanan sehingga pengguna dapat memperoleh informasi dengan cepat dan tepat. Pelayanan yang kurang baik akan menyebabkan keterlambatan dalam memperoleh hasil yang dibutuhkan.

Metode *PIECES* pada penelitian ini digunakan untuk menganalisis permasalahan yang terdapat pada sistem service center yang masih berjalan secara manual. Analisis dilakukan pada setiap aspek *PIECES* untuk mengidentifikasi kelemahan sistem, seperti keterlambatan dalam pencarian data, ketidaktepatan informasi, serta kurang optimalnya pelayanan kepada pelanggan. Hasil dari analisis *PIECES* ini nantinya akan digunakan sebagai dasar dalam menentukan kebutuhan sistem serta merancang solusi berupa sistem informasi *monitoring* berbasis *website* yang dilengkapi dengan notifikasi *WhatsApp*.

2.10 *Black-box Testing*

Black Box Testing merupakan salah satu metode pengujian perangkat lunak yang berfokus pada pengujian fungsi sistem tanpa melihat struktur internal atau kode program. Pengujian ini dilakukan dengan cara memberikan *input* pada sistem dan kemudian mengamati *output* yang dihasilkan, apakah sudah sesuai dengan yang diharapkan [23]. Metode *Black Box Testing* digunakan untuk memastikan bahwa setiap fungsi dalam sistem berjalan dengan baik sesuai dengan kebutuhan pengguna. Pengujian ini dilakukan berdasarkan spesifikasi sistem yang telah dirancang sebelumnya, sehingga dapat diketahui apakah sistem telah memenuhi kebutuhan yang diinginkan [20].

Dalam pengujian *Black Box*, penguji tidak perlu mengetahui bagaimana sistem dibangun secara internal, melainkan hanya berfokus pada hasil yang diberikan oleh sistem. Hal ini membuat metode ini mudah digunakan serta efektif dalam menemukan kesalahan pada fungsi sistem [26]. Pengujian *Black Box* biasanya dilakukan dengan beberapa teknik, seperti pengujian *input* data, validasi *form*, serta pengujian *output* sistem. Tujuan utama dari pengujian ini adalah untuk memastikan bahwa sistem dapat berjalan sesuai dengan yang diharapkan dan bebas dari kesalahan yang dapat mengganggu pengguna [23]. Berikut merupakan contoh hasil pengujian menggunakan metode *Black Box Testing*. Berdasarkan tabel 2.4, pengujian *Black Box* dilakukan dengan menguji beberapa fungsi sistem seperti *login*, pengelolaan data pengguna, dan manajemen modul. Hasil pengujian menunjukkan bahwa seluruh fungsi sistem berjalan dengan baik sesuai dengan yang diharapkan [40].

Tabel 2. 4 Hasil Pengujian *Black Box Testing*

No	Form	Pengujian	Keterangan
1	<i>Login</i>	<i>Login</i> menggunakan id pengguna dan <i>password</i>	Sukses
2	Pengguna	Menyimpan data pengguna	Sukses
		Mengedit data pengguna	Sukses

		Menghapus data pengguna	Sukses
3	Management User	Menentukan hak akses berdasarkan level pengguna	Sukses
4	Modul	Upload modul	Sukses
		Edit modul	Sukses
		Download modul	Sukses

Pada penelitian ini, metode *Black Box Testing* digunakan untuk menguji fungsi-fungsi utama pada sistem informasi *monitoring* perbaikan berbasis *website* yang dikembangkan. Pengujian dilakukan pada fitur-fitur seperti pengelolaan data perbaikan, proses *monitoring* status perbaikan, serta pengiriman notifikasi *WhatsApp* kepada pelanggan. Hasil dari pengujian ini digunakan untuk memastikan bahwa seluruh fungsi sistem telah berjalan sesuai dengan kebutuhan yang telah ditentukan dan dapat digunakan dengan baik oleh pengguna.

2.11 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) merupakan metode pengujian perangkat lunak yang dilakukan oleh pengguna akhir (*end user*) untuk memastikan bahwa sistem yang dikembangkan telah sesuai dengan kebutuhan pengguna. *UAT* digunakan untuk mengukur kualitas sistem serta mengetahui tingkat kepuasan pengguna terhadap sistem yang telah dibangun. Pengujian ini berfokus pada pengalaman pengguna dalam menggunakan sistem secara langsung sehingga dapat diketahui apakah sistem telah memenuhi kebutuhan dan harapan pengguna [41].

Metode *UAT* digunakan untuk mengevaluasi efektivitas sistem dari perspektif pengguna, termasuk aspek kemudahan penggunaan, kelengkapan fitur, serta tingkat kepuasan pengguna terhadap sistem. Pengujian ini dilakukan dengan melibatkan beberapa responden sebagai pengguna sistem yang akan memberikan penilaian berdasarkan pengalaman mereka saat menggunakan sistem tersebut [42]. Berikut adalah tahapan-tahapan dalam melakukan *UAT* [41] :

1. Perencanaan *UAT (UAT Planning)*, yaitu menyusun rencana pengujian yang mencakup tujuan, ruang lingkup, serta pihak yang terlibat dalam proses pengujian.
2. Penyusunan Pertanyaan Pengujian (*Preparing Test Questions*), yaitu menyusun instrumen pengujian berupa kuesioner yang akan digunakan untuk mengukur tingkat penerimaan pengguna terhadap sistem.

3. Pelaksanaan Pengujian (*Conducting the Test*), yaitu pengguna melakukan pengujian sistem secara langsung dan mengisi kuesioner berdasarkan pengalaman penggunaan sistem.
4. Analisis Hasil Pengujian (*Analyzing Results*), yaitu mengolah hasil kuesioner untuk mengetahui tingkat penerimaan sistem serta menarik kesimpulan dari hasil pengujian yang telah dilakukan.

Dalam pelaksanaan *UAT*, metode pengumpulan data yang digunakan adalah kuesioner dengan menggunakan skala *Likert*. Skala *Likert* digunakan untuk mengukur tingkat kepuasan pelanggan berdasarkan kategori nilai tertentu. Hasil penilaian dari responden kemudian diolah menjadi nilai persentase untuk mengetahui tingkat penerimaan sistem [41]. Selanjutnya, hasil persentase tersebut diinterpretasikan ke dalam kriteria penilaian tertentu, seperti yang ditunjukkan pada tabel 2.5. Kriteria tersebut digunakan untuk mengelompokkan hasil penilaian pengguna ke dalam kategori tidak baik, kurang baik, cukup, baik, dan sangat baik berdasarkan rentang persentase yang diperoleh.

Tabel 2. 5 Skala Likert

% Jumlah Skor	Kriteria
20,00% - 36,00%	Tidak Baik
36,01% - 52,00%	Kurang Baik
52,01% - 68,00%	Cukup
68,01% - 84,00%	Baik
84,01% - 100%	Sangat Baik

Pada penelitian ini, metode *User Acceptance Testing (UAT)* digunakan untuk mengukur tingkat penerimaan pengguna terhadap sistem informasi *monitoring service* berbasis *website* yang telah dikembangkan. Pengujian dilakukan dengan melibatkan pengguna *service center* sebagai responden yang diminta untuk mencoba sistem serta mengisi kuesioner berdasarkan pengalaman penggunaan sistem. Hasil pengujian ini digunakan untuk mengetahui apakah sistem telah memenuhi kebutuhan pengguna serta layak untuk diimplementasikan.

2.12 Service Center

Service center merupakan bagian dari pelayanan yang berfungsi untuk membantu pelanggan dalam memenuhi kebutuhan layanan serta menangani keluhan yang disampaikan. Pada service center, pelanggan dapat menyampaikan keluhan baik secara langsung maupun tidak langsung. Keluhan tersebut kemudian akan diterima dan ditindaklanjuti oleh pihak

yang bertanggung jawab agar permasalahan yang dialami pelanggan dapat diselesaikan dengan baik [43]. Dalam pelaksanaannya, service center menjadi tempat yang terpusat dalam menerima berbagai bentuk keluhan pelanggan. Keluhan yang disampaikan dapat berupa keluhan terhadap pelayanan yang diberikan, fasilitas yang digunakan, maupun kualitas layanan yang dirasakan pelanggan. Keluhan tersebut dapat disampaikan secara langsung melalui komunikasi dengan petugas maupun secara tidak langsung melalui media tertentu yang disediakan oleh perusahaan [43].

Service center memiliki peran penting dalam proses penanganan keluhan pelanggan. Setiap keluhan yang diterima akan dicatat dan diteruskan kepada bagian yang terkait untuk segera ditindaklanjuti. Proses penanganan keluhan dilakukan agar permasalahan yang dialami pelanggan dapat diselesaikan dengan cepat dan tepat. Penanganan keluhan yang baik akan memberikan dampak positif terhadap pelanggan karena pelanggan merasa diperhatikan dan mendapatkan solusi atas permasalahan yang dihadapi [43]. Selain itu, service center juga berfungsi sebagai sarana komunikasi antara pelanggan dan perusahaan. Melalui service center, pelanggan dapat memperoleh informasi yang dibutuhkan terkait layanan yang diberikan oleh perusahaan. Service center juga menjadi penghubung dalam menyampaikan informasi dari perusahaan kepada pelanggan, sehingga komunikasi antara kedua pihak dapat berjalan dengan baik [43].

Dalam memberikan pelayanan, service center dituntut untuk memberikan pelayanan yang cepat, tepat, dan sesuai dengan kebutuhan pelanggan. Pelayanan yang tidak sesuai atau lambat dalam penanganannya dapat menimbulkan ketidakpuasan pelanggan. Oleh karena itu, service center harus mampu memberikan respon yang baik terhadap setiap keluhan maupun kebutuhan pelanggan agar pelayanan yang diberikan dapat berjalan secara optimal [44]. Kualitas pelayanan menjadi salah satu faktor penting dalam keberhasilan service center. Kualitas layanan yang baik akan mempengaruhi tingkat kepuasan pelanggan dalam menggunakan jasa yang diberikan oleh perusahaan. Semakin baik kualitas pelayanan yang diberikan, maka semakin tinggi pula tingkat kepercayaan dan kepuasan pelanggan terhadap perusahaan [45].

Dengan demikian, service center memiliki peran yang sangat penting dalam mendukung kegiatan pelayanan perusahaan, terutama dalam menangani keluhan, memberikan informasi, serta menjaga hubungan yang baik dengan pelanggan. Oleh karena itu, pengelolaan service center yang baik sangat diperlukan untuk meningkatkan kualitas pelayanan serta kepuasan pelanggan secara keseluruhan.

2.13 Advance Service Center

Advance Service Center merupakan layanan purna jual (*after sales service*) yang disediakan oleh Advance Digital sebagai brand elektronik lokal di Indonesia. *Service center* ini berfungsi sebagai tempat penanganan perbaikan, pengecekan, serta klaim garansi terhadap produk-produk Advance Digital. Advance Service Center ini berlokasi di Komplek The Enterprise, Jl. Kapten Muslim Blok B25, Dwi Kora, Kecamatan Medan Helvetia, Kota Medan, Sumatera Utara. Lokasi ini menjadi tempat berlangsungnya proses pelayanan perbaikan serta interaksi antara pelanggan dan pihak *service center*.

Pada Gambar 2.9 dapat dilihat kondisi Advance Service Center yang digunakan sebagai lokasi penelitian. Gambar tersebut menunjukkan area pelayanan yang menjadi tempat berlangsungnya aktivitas perbaikan, mulai dari penerimaan perangkat hingga proses monitoring perbaikan.



Gambar 2. 9 Advance Service Center

Dalam operasional sehari-hari, Advance Service Center telah menggunakan sistem berbasis komputer untuk membantu pengelolaan data perbaikan. Sistem tersebut digunakan oleh admin untuk mencatat data pelanggan, data perangkat, serta proses perbaikan yang dilakukan oleh teknisi. Selain itu, sistem juga menyediakan fitur untuk melihat status perbaikan perangkat. Adapun proses bisnis yang berjalan dimulai ketika pelanggan datang ke *service center* dengan membawa perangkat yang mengalami kerusakan. Admin kemudian melakukan input data pelanggan dan data perangkat ke dalam sistem. Setelah itu, perangkat

diserahkan kepada teknisi untuk dilakukan pengecekan guna mengetahui jenis kerusakan. Berdasarkan hasil pengecekan, teknisi akan menentukan tindakan perbaikan yang diperlukan dan memperbarui status perbaikan pada sistem.

Selanjutnya, perangkat akan melalui tahap perbaikan hingga selesai. Setelah proses perbaikan selesai, admin akan menginformasikan kepada pelanggan bahwa perangkat sudah dapat diambil. Namun, penyampaian informasi ini masih dilakukan secara manual melalui telepon atau pesan singkat. Pelanggan juga sering kali harus menghubungi pihak service center untuk menanyakan perkembangan status perbaikan perangkat mereka. Meskipun sistem yang ada telah membantu dalam pengelolaan data perbaikan, sistem tersebut belum memiliki fitur notifikasi otomatis kepada pelanggan ketika terjadi perubahan status perbaikan. Hal ini menyebabkan proses penyampaian informasi masih belum efektif dan bergantung pada komunikasi manual antara admin dan pelanggan.

Berdasarkan kondisi tersebut, Advance Service Center digunakan sebagai objek penelitian untuk mengembangkan sistem informasi monitoring status perbaikan berbasis web yang dilengkapi dengan notifikasi WhatsApp. Sistem yang dikembangkan diharapkan dapat meningkatkan efektivitas penyampaian informasi serta memberikan kemudahan bagi pelanggan dalam memperoleh informasi status perbaikan secara cepat dan tepat.

