

BAB II

KAJIAN LITERATUR

2.1 Konsep Sistem Informasi

Konsep sistem informasi merupakan fondasi utama dalam memahami bagaimana teknologi, data, dan manusia berinteraksi dalam suatu organisasi untuk mencapai tujuan tertentu. Pemahaman yang komprehensif mengenai konsep ini memerlukan tinjauan mendalam terhadap elemen-elemen penyusunnya, yaitu sistem dan informasi, serta bagaimana keduanya berintegrasi menjadi satu kesatuan yang fungsional dalam mendukung operasional lembaga [10].

2.1.1 Sistem

Secara teoretis, sistem didefinisikan sebagai sekumpulan elemen, komponen, atau variabel yang terorganisasi, saling berinteraksi, dan terintegrasi untuk mencapai tujuan yang sama [11]. Hal menekankan bahwa sistem merupakan kelompok dari dua atau lebih komponen atau subsistem yang saling berhubungan [10]. Dalam konteks organisasi, sistem sering kali dipandang sebagai jaringan kerja dari prosedur-prosedur yang saling berkaitan, yang berkumpul bersama untuk melakukan suatu kegiatan atau menyelesaikan sasaran tertentu [12].

Setiap sistem memiliki karakteristik tertentu yang membedakannya dengan lingkungan luar, termasuk adanya batas sistem (*boundary*) yang menentukan ruang lingkup operasi sistem tersebut. Komponen-komponen di dalam sistem bekerja melalui mekanisme masukan (*input*), proses, dan keluaran (*output*), di mana energi atau data yang masuk diolah untuk menghasilkan hasil yang bermanfaat bagi pengguna. Selain itu, hubungan antar subsistem sangat menentukan keberhasilan suatu sistem karena koordinasi yang baik diperlukan agar sasaran akhir dapat tercapai secara efektif dan efisien [10].

Dalam klasifikasinya, sistem dapat bersifat fisik maupun abstrak, serta terbuka atau tertutup. Sistem informasi merupakan contoh dari sistem buatan manusia yang melibatkan interaksi antara manusia dengan mesin (*human-machine system*), di mana penggunaan komputer menjadi media utama dalam memproses data. Dengan demikian, pemahaman terhadap sistem tidak hanya terbatas pada komponen teknisnya, tetapi juga mencakup ketergantungan antar bagian dalam menjalankan fungsi organisasional [10].

2.1.2 Informasi

Informasi merupakan data yang telah diolah menjadi bentuk yang lebih berguna dan memiliki arti bagi penerimanya. Data sendiri merupakan bentuk mentah yang belum memiliki makna, seperti angka, simbol, atau gambar, yang kemudian melalui proses pengolahan tertentu untuk menghasilkan pengetahuan yang dapat mendukung pengambilan keputusan [10], [11]. Menurut Raymond McLeod, kegunaan utama dari informasi adalah untuk membantu pemimpin atau manajer dalam membuat keputusan yang tepat berdasarkan fakta yang ada [10].

Kualitas sebuah informasi sangat bergantung pada tiga pilar utama, yaitu relevansi, akurasi, dan ketepatan waktu. Informasi dikatakan relevan apabila memiliki manfaat langsung bagi penerimanya sesuai dengan konteks permasalahan yang dihadapi. Akurasi merujuk pada ketepatan nilai informasi tersebut, sementara ketepatan waktu memastikan informasi tersedia sebelum keputusan diambil, karena informasi yang terlambat tidak lagi memiliki nilai guna dalam proses manajerial [10].

Selain kualitas, informasi juga memiliki siklus hidup yang disebut sebagai siklus informasi (*information cycle*), di mana tindakan yang dilakukan berdasarkan informasi akan menghasilkan data baru yang kemudian ditangkap sebagai masukan untuk diproses kembali. Dalam perspektif manajemen, informasi juga diklasifikasikan berdasarkan dimensi waktu, sasaran, dan persyaratan fungsionalnya. Hal ini menunjukkan bahwa informasi bukan sekadar produk sampingan, melainkan aset krusial yang menentukan kredibilitas dan kualitas layanan suatu lembaga [10], [13].

2.1.3 Sistem Informasi

Sistem informasi adalah sebuah komponen penting dalam organisasi modern yang memungkinkan pengumpulan, penyimpanan, pemrosesan, dan penyebaran informasi untuk tujuan strategis [10]. Secara lebih spesifik, Tata Sutabri mendefinisikan sistem informasi sebagai suatu sistem di dalam organisasi yang berbagai mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi yang bersifat manajerial dengan kegiatan strategi organisasi. Sistem ini mengintegrasikan teknologi informasi dengan prosedur kerja dan sumber daya manusia untuk menghasilkan laporan yang diperlukan oleh pihak internal maupun eksternal [11], [12].

Struktur sistem informasi terdiri dari beberapa blok bangunan (*building block*), yang meliputi blok masukan, blok model, blok keluaran, blok teknologi, blok basis data, dan blok kendali. Blok-blok ini bekerja secara sinergis di mana data yang dimasukkan akan dimanipulasi oleh model logika dan matematika tertentu untuk menghasilkan keluaran berkualitas yang disimpan dalam basis data. Keberadaan blok kendali sangat penting untuk melindungi sistem dari berbagai ancaman seperti kecurangan, kegagalan sistem, atau gangguan eksternal lainnya yang dapat merusak integritas data [12].

Implementasi sistem informasi, khususnya yang berbasis *web* pada kantor notaris, bertujuan untuk meningkatkan efisiensi, keamanan, dan aksesibilitas dokumen yang berkaitan dengan hak-hak legal klien. Dengan mengotomatisasi proses manual yang cenderung tidak efisien dan rawan kerusakan, sistem informasi mampu meminimalisir kesalahan manusia (*human error*) serta memungkinkan pelacakan status dokumen secara waktu nyata (*real-time*) [13]. Pada akhirnya, penggunaan sistem informasi yang terintegrasi membantu organisasi dalam meningkatkan kualitas pelayanan dan mendukung pengambilan keputusan yang lebih akurat di berbagai tingkatan manajemen [12].

2.2 Administrasi Notaris

Administrasi secara teoretis didefinisikan sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, yang berkumpul bersama untuk melakukan suatu kegiatan atau menyelesaikan sasaran tertentu [12]. Dalam konteks kenotariatan, administrasi merupakan pengelolaan data dan pemantauan proses manajemen yang mendukung pelaksanaan tugas Notaris dan Pejabat Pembuat Akta Tanah (PPAT) dalam menghasilkan akta autentik [11], [14]. Peran administrasi sangat vital sebagai pendukung fungsi operasional organisasi yang bersifat manajerial, bertujuan mempertemukan kebutuhan pengolahan transaksi harian dengan kegiatan strategis lembaga untuk memberikan pelayanan jasa hukum yang efektif kepada masyarakat [12]. Melalui administrasi yang baik, notaris dapat menjamin integritas data klien serta keabsahan dokumen yang berkaitan dengan hak-hak legal klien [13].

Dasar hukum pelaksanaan administrasi notaris di Indonesia berlandaskan pada Undang-Undang Nomor 2 Tahun 2014 tentang Perubahan atas Undang-Undang Nomor 30 Tahun 2004 tentang Jabatan Notaris (UUJN) [15], [16]. Selain UUJN, administrasi bagi PPAT secara spesifik mengacu pada Peraturan Pemerintah Nomor 37 Tahun 1998 dan perubahannya melalui PP Nomor 24 Tahun 2016 [17]. Secara hukum, notaris memiliki kewajiban administratif untuk membuat akta dalam bentuk minuta akta, yaitu naskah asli

akta yang memuat tanda tangan para pihak, saksi, dan notaris sebagai bukti autentik yang memiliki kekuatan hukum tetap. Minuta akta ini disimpan oleh notaris sebagai bagian dari Protokol Notaris yang merupakan arsip negara, serta menjadi dasar dalam penerbitan salinan atau *grosse* akta kepada para pihak [16]. Selain itu, notaris juga berkewajiban menjamin kepastian tanggal serta menjaga keutuhan dan keamanan dokumen tersebut. Di era digital, kewajiban administratif ini juga didukung oleh UU Nomor 11 Tahun 2008 tentang Informasi dan Transaksi Elektronik (UU ITE) yang mengakui dokumen dan tanda tangan elektronik sebagai alat bukti hukum yang sah, sehingga memungkinkan notaris untuk mengintegrasikan teknologi digital dalam proses pelayanan hukum [15], [16].

Tugas dan fungsi administrasi notaris mencakup pengelolaan akta, data klien, hingga penyimpanan dokumen protokol secara sistematis [16]. Jenis-jenis administrasi yang dikelola meliputi administrasi akta (*register* penomoran), administrasi klien (identitas para pihak), administrasi layanan (status proses permohonan), dan administrasi arsip (minuta dan warkah) [14], [16]. Proses administrasi ini umumnya dimulai dari tahap penerimaan berkas identitas klien, verifikasi dokumen, penyusunan draf akta, hingga tahap pelaporan bulanan dan pengarsipan protokol [14], [17]. Efektivitas dalam setiap tahapan ini sangat menentukan kualitas pelayanan, mengingat kesalahan kecil dalam penomoran atau pengetikan dapat berisiko hukum di kemudian hari [14], [18].

Sistem administrasi manual yang masih banyak digunakan pada kantor notaris saat ini memiliki berbagai keterbatasan, seperti ketergantungan pada buku register fisik dan penyimpanan kertas di lemari arsip yang rentan terhadap kerusakan, kehilangan, maupun pemalsuan [13], [16], [18]. Kelemahan utama sistem manual meliputi lambatnya proses pencarian data, risiko redundansi data, serta ketidakteraturan nomor urut akta akibat *human error* [14], [16], [18]. Oleh karena itu, terdapat kebutuhan mendesak akan sistem administrasi terkomputerisasi yang mampu meningkatkan efisiensi waktu, keamanan data, dan kemudahan akses [14], [18][19]. Sistem yang baik harus memiliki karakteristik terstruktur, akurat, dan menjamin kerahasiaan (*confidentiality*) melalui pengaturan hak akses terbatas agar integritas dokumen negara tetap terjaga [13], [18].

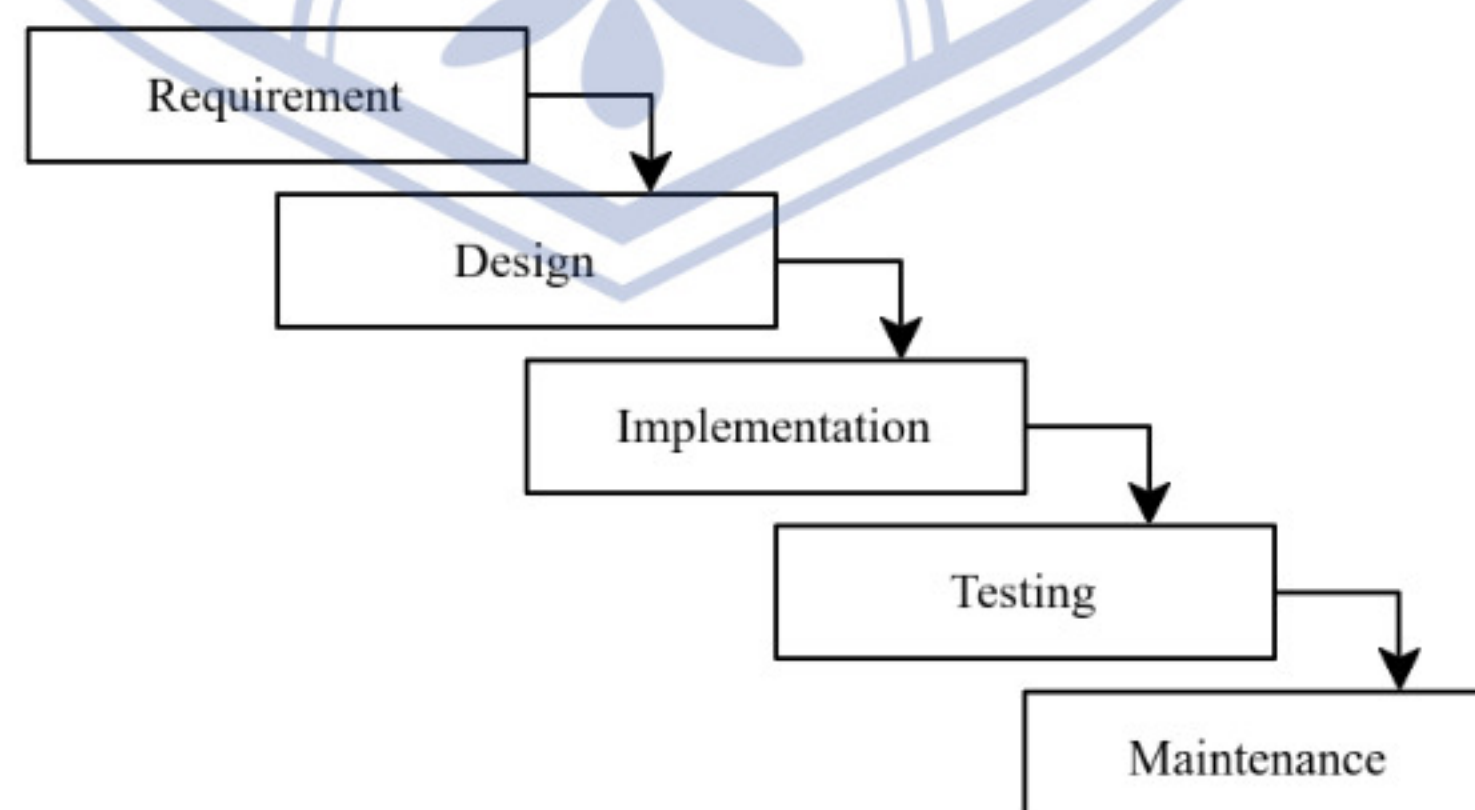
Teknologi informasi, khususnya sistem berbasis *web*, memainkan peran transformatif dalam modernisasi administrasi notaris melalui digitalisasi arsip dan penyediaan akses multi-pengguna (*multi-user access*) [13],[14],[18]. Implementasi sistem ini memungkinkan otomatisasi penomoran akta, pelacakan riwayat dokumen secara waktu nyata (*real-time*), dan penyediaan *audit trail* atau *log* aktivitas pengguna guna meminimalisir praktik kecurangan [13]. Inovasi seperti *Cyber Notary* dan integrasi dengan sistem

Administrasi Hukum Umum (AHU) *Online* maupun sistem pendaftaran hak atas tanah secara elektronik membuktikan bahwa teknologi mampu meminimalkan hambatan birokrasi dan memperkuat kepastian hukum [15],[16]. Dengan dukungan teknologi, administrasi notaris beralih dari sekadar pencatatan rutin menjadi sistem manajemen pengetahuan yang mendukung pengambilan keputusan manajerial yang lebih akurat dan profesional [13].

2.3 Waterfall

Metode pengembangan perangkat lunak *Waterfall* merupakan salah satu model pengembangan sistem yang paling fundamental dan sering disebut sebagai *linear sequential model* atau *classical life cycle* [20], [21]. Model ini mengusulkan pendekatan pengembangan perangkat lunak yang bersifat sistematis dan sekuensial, di mana setiap tahapan dalam proses pengembangan harus diselesaikan secara berurutan [21], [22]. Pendekatan *Waterfall* dipandang sebagai model *System Development Life Cycle* (SDLC) pertama yang digunakan secara luas dalam rekayasa perangkat lunak untuk memastikan keberhasilan suatu proyek melalui struktur yang terorganisasi dengan baik [20], [22].

Analogi "air terjun" dalam model ini menggambarkan bahwa setiap fase pengembangan mengalir secara linear dari tingkat atas menuju ke bawah, sehingga setiap tahap tidak boleh dilakukan secara bersamaan atau tumpang tindih. Sifatnya yang terstruktur memastikan bahwa informasi sistem yang dikembangkan menjadi lebih terfokus dan terdokumentasi di setiap tahapannya [22], [23]. Meskipun model ini dianggap kurang iteratif dan fleksibel dibandingkan metode modern lainnya, *Waterfall* tetap menjadi pilihan utama dalam studi pengembangan sistem karena kesederhanaan dan kedisiplinannya dalam proses pengerjaan [20], [23].



Gambar 2.1 Tahapan Waterfall

Secara teknis, model *Waterfall* merupakan metode pengembangan perangkat lunak yang terdiri dari beberapa tahapan berurutan yang harus diselesaikan secara sistematis. Setiap tahap memiliki peran spesifik dalam memastikan bahwa perangkat lunak yang dikembangkan sesuai dengan kebutuhan pengguna dan memiliki kualitas yang terjamin [20], [23].

Adapun tahapan dalam model *Waterfall* adalah sebagai berikut:

1. *Requirements* (Analisis Kebutuhan)

Tahap ini merupakan proses awal yang bertujuan untuk mengidentifikasi, menganalisis dan mendefinisikan kebutuhan perangkat lunak yang ingin dibuat secara menyeluruh. Kebutuhan yang dikumpulkan harus jelas dan terperinci agar dapat dipahami dengan baik oleh pengguna maupun pengembang [20], [23].

2. *Design* (Perancangan)

Setelah kebutuhan teridentifikasi, tahap ini berfokus pada penerjemahan kebutuhan ke dalam bentuk rancangan sistem. Perancangan mencakup struktur data, arsitektur sistem, serta desain antarmuka yang akan digunakan oleh pengguna [20], [21], [23].

3. *Coding* (Implementasi/Pengkodean)

Pada tahap ini, hasil perancangan diimplementasikan ke dalam bahasa pemrograman tertentu. Tujuannya adalah menghasilkan program komputer yang dapat berjalan sesuai dengan spesifikasi yang telah ditentukan [20].

4. *Testing* (Pengujian)

Tahap pengujian dilakukan untuk memverifikasi bahwa seluruh komponen sistem telah berfungsi dengan baik. Pengujian mencakup aspek logika dan fungsional guna memastikan sistem bebas dari kesalahan sebelum digunakan [20], [21], [23].

5. *Maintenance* (Pemeliharaan)

Pemeliharaan mencakup perbaikan kesalahan yang ditemukan serta penyesuaian sistem agar tetap berjalan optimal sesuai kebutuhan pengguna [20], [23].

2.5 Analisis *Pieces*

Analisis *PIECES* merupakan sebuah kerangka kerja (*framework*) yang digunakan untuk mengklasifikasikan masalah (*problems*), peluang (*opportunities*), dan arahan (*directives*) yang ditemukan pada tahap definisi ruang lingkup analisis serta perancangan sistem [24]. Secara akademis, kerangka kerja ini berfungsi sebagai instrumen evaluasi untuk mengukur kinerja suatu sistem informasi atau aplikasi berdasarkan indikator-indikator tertentu [25]. Dengan menggunakan pendekatan ini, pengembang dapat mendeteksi berbagai

permasalahan dari perspektif yang komprehensif, sehingga dihasilkan pertimbangan-pertimbangan baru yang krusial dalam proses pengembangan sistem [24], [26]. Peran utamanya dalam siklus pengembangan adalah untuk memastikan bahwa sistem yang dibangun mampu menunjang kinerja organisasi menjadi lebih baik dan mencapai tingkat kepuasan pengguna yang optimal [27].

Struktur utama analisis *PIECES* terdiri dari enam variabel penilaian, yaitu kinerja (*performance*), informasi (*information*), ekonomi (*economic*), pengendalian (*control*), efisiensi (*efficiency*), dan pelayanan (*service*). Keenam variabel ini digunakan untuk mengevaluasi sistem secara menyeluruh guna mengidentifikasi permasalahan serta peluang peningkatan dalam proses operasional [24], [25]. Adapun penjelasan dari masing-masing komponen dalam analisis *PIECES* adalah sebagai berikut:

1. *Performance* (Kinerja)

Variabel ini berfokus pada evaluasi prosedur yang sedang berjalan, khususnya dalam meningkatkan kecepatan penyelesaian proses dan kualitas output yang dihasilkan. Kinerja sistem diukur berdasarkan kemampuannya dalam menangani beban kerja serta kecepatan dalam memproses data [25], [27].

2. *Information* (Informasi)

Variabel informasi digunakan untuk menilai kualitas informasi yang dihasilkan oleh sistem. Hal ini mencakup akurasi, relevansi, dan kemudahan akses data bagi pengguna dalam mendukung proses pengambilan keputusan [27].

3. *Economic* (Ekonomi)

Variabel ini menilai sistem dari aspek biaya dan manfaat yang dihasilkan. Evaluasi dilakukan untuk memastikan bahwa sistem memberikan nilai guna yang sebanding atau lebih besar dibandingkan dengan biaya operasional yang dikeluarkan [25], [27].

4. *Control* (Pengendalian)

Komponen pengendalian berkaitan dengan tingkat keamanan dan pengawasan dalam sistem. Fokus utama berada pada perlindungan data serta kemampuan sistem dalam mencegah dan mengatasi potensi ancaman atau gangguan [27].

5. *Efficiency* (Efisiensi)

Variabel efisiensi menekankan pada optimalisasi penggunaan sumber daya. Sistem diharapkan mampu meminimalkan input manual serta memaksimalkan hasil dengan proses yang lebih cepat dan efektif [27].

6. *Service* (Pelayanan)

Variabel pelayanan menilai kualitas interaksi antara sistem dengan pengguna. Aspek yang diperhatikan meliputi kemudahan penggunaan, keakuratan fitur, serta kesesuaian sistem dengan kebutuhan operasional sehari-hari [27].

Dalam tahap analisis sistem, *PIECES* berperan sebagai panduan untuk melakukan identifikasi masalah secara sistematis melalui observasi dan wawancara [26]. Langkah-langkah analisis dimulai dengan memetakan kondisi sistem yang sedang berjalan (biasanya sistem manual) dan membandingkannya dengan kebutuhan fungsional pada sistem baru yang diusulkan [24], [26]. Jika dibandingkan dengan metode evaluasi lain seperti *Technology Acceptance Model* (TAM) yang hanya menitikberatkan pada persepsi kemudahan dan kemanfaatan, kerangka *PIECES* memiliki keunggulan pada kedalaman aspek evaluasinya yang mencakup enam poin kritis [27]. Meskipun demikian, *PIECES* memiliki keterbatasan dalam hal detail alur waktu proses, namun sifatnya yang ringan dan tidak memerlukan data masif menjadikannya sangat efektif untuk penelitian pada organisasi atau perusahaan berskala menengah [24], [27].

Melalui analisis *PIECES*, sistem berbasis *web* yang dikembangkan dirancang untuk melakukan digitalisasi proses tersebut, sehingga memungkinkan akses multi-pengguna yang lebih cepat, integrasi data yang akurat, serta peningkatan kualitas pelayanan hukum kepada masyarakat melalui transparansi status layanan [24], [26].

2.6 Unified Modeling Language

Unified Modeling Language (UML) merupakan bahasa standar yang digunakan untuk mendefinisikan, memvisualisasikan, membangun, dan mendokumentasikan artefak dari sistem perangkat lunak [28]. UML muncul sebagai respons atas kebutuhan akan standarisasi bahasa pemodelan visual yang mampu menggambarkan arsitektur sistem dalam pemrograman berorientasi objek [29], [30]. Sebagai alat komunikasi yang bersifat ekspresif, UML memfasilitasi pertukaran ide antara analis sistem, pengembang, dan pemangku kepentingan bisnis dengan menyajikan model yang terstruktur dan sistematis [29].

Pemodelan dengan UML tidak terbatas pada metodologi tertentu, melainkan berfungsi sebagai kerangka kerja yang fleksibel dalam setiap fase siklus hidup pengembangan sistem (*System Development Life Cycle*) [30], [31]. Melalui abstraksi konsep dasar yang mencakup klasifikasi struktural dan perilaku dinamis, UML memungkinkan pengembang untuk memahami alur kerja serta interaksi antar pengguna dengan lebih mudah [32], [33]. Penggunaan diagram UML yang tepat sangat krusial untuk meminimalkan

kesalahan pada tahap pengembangan serta memastikan bahwa spesifikasi sistem yang dirancang telah sesuai dengan kebutuhan nyata di lapangan [33].

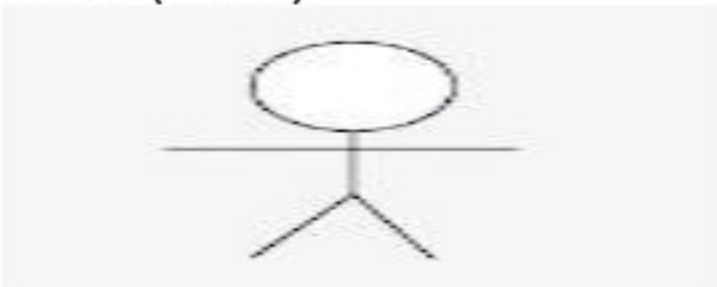
Integrasi UML dalam proyek rekayasa perangkat lunak bertujuan untuk mentransformasi metode administrasi konvensional menjadi ekosistem digital yang lebih sederhana dan efisien [34]. Dengan menyediakan perangkat diagram standar, UML membantu memodelkan struktur statis maupun perilaku dinamis sistem guna menjaga konsistensi data dan integritas operasional lembaga [29], [34]. Keandalan UML sebagai standar industri menjadikannya instrumen utama dalam merancang sistem informasi yang kompleks, di mana visualisasi hubungan komponen yang detail menjadi jembatan komunikasi yang efektif antara pengembang dan pengguna akhir [30], [34]. Berikut beberapa jenis diagram yang tercakup dalam *unified modeling language*:

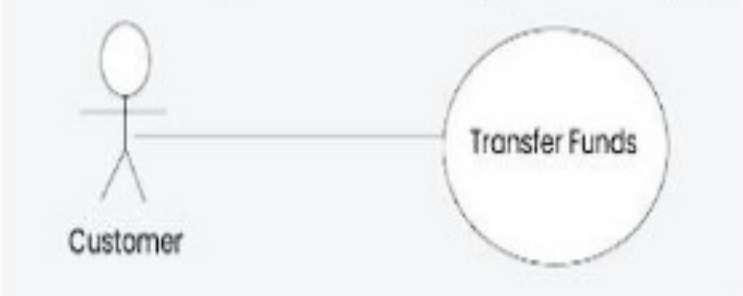
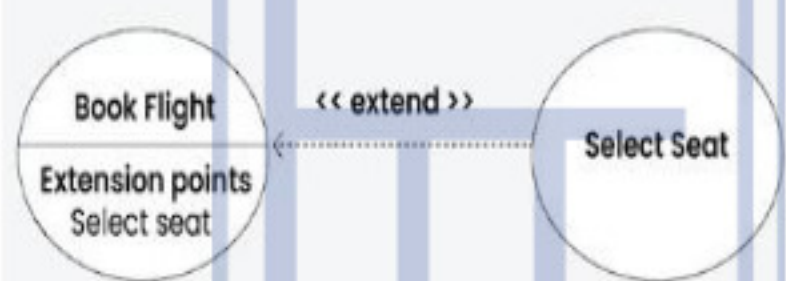
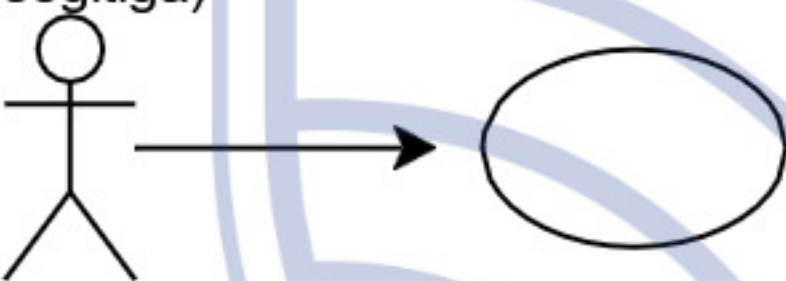
1. Use Case Diagram

Use Case Diagram merupakan salah satu diagram perilaku dalam UML yang berfungsi untuk memodelkan fungsionalitas atau layanan yang disediakan oleh sistem informasi [30], [35]. Diagram ini digunakan untuk memberikan gambaran secara umum mengenai interaksi yang terjadi antara sistem dengan lingkungannya melalui aktor-aktor eksternal [28], [29]. Fokus utama dari *use case* adalah mengidentifikasi kebutuhan fungsional dari sudut pandang pengguna, sehingga pengembang dapat memahami apa saja layanan yang harus disediakan sistem tanpa harus menjelaskan mekanisme teknis internalnya [33], [35].

Struktur utama dalam *use case diagram* terdiri dari aktor, *use case*, dan hubungan antar elemen tersebut yang digambarkan melalui garis asosiasi [29], [35]. Aktor mempresentasikan entitas eksternal seperti manusia, kelompok, atau sistem lain yang berinteraksi dengan sistem untuk memulai suatu fungsi tertentu [28], [29]. Sementara itu, *use case* disimbolkan dengan bentuk elips yang mewakili unit fungsionalitas spesifik atau tugas yang dapat dilakukan oleh sistem berdasarkan permintaan aktor [29], [35]. Berikut tabel simbol-simbol yang membangun struktur dalam *use case diagram* beserta deskripsinya [36].

Tabel 2.1 Simbol Komponen Use Case Diagram

Simbol	Deskripsi	Fungsionalitas
Aktor (<i>Actor</i>) 	Simbol visual berupa gambar orang (<i>stick figure</i>). Simbol ini merepresentasikan pengguna atau sistem eksternal yang berinteraksi dengan sistem yang sedang dimodelkan.	Merepresentasikan pengguna atau sistem eksternal yang berinteraksi dengan sistem yang sedang dimodelkan.

Use Case 	Simbol visual berupa bentuk elips dengan nama di dalamnya. Simbol ini merepresentasikan unit fungsionalitas sistem atau interaksi spesifik antara aktor dan sistem.	Merepresentasikan unit fungsionalitas sistem atau interaksi spesifik antara aktor dan sistem.
Asosiasi (Garis Penghubung) 	Menghubungkan antara aktor dengan <i>use case</i> . Simbol ini digunakan untuk menunjukkan interaksi atau komunikasi langsung antara aktor dan <i>use case</i> tertentu.	Digunakan untuk menunjukkan interaksi atau komunikasi langsung antara aktor dan <i>use case</i> tertentu.
Include (Garis Putus-putus berpanah dengan label <<include>>) 	Menunjukkan bahwa suatu <i>use case</i> secara fungsional membutuhkan <i>use case</i> lain. Simbol ini digunakan untuk memodelkan fungsionalitas yang wajib ada (<i>required</i>) atau bagian dari <i>use case</i> dasar agar <i>use case</i> tersebut dapat selesai dijalankan.	Digunakan untuk memodelkan fungsionalitas yang wajib ada (<i>required</i>) atau bagian dari <i>use case</i> dasar agar <i>use case</i> tersebut dapat selesai dijalankan.
Extend (Garis Putus-putus berpanah dengan label <<extend>>) 	Menunjukkan perluasan opsional dari suatu <i>use case</i> . Simbol ini digunakan untuk menambahkan langkah atau perilaku tambahan ke <i>use case</i> dasar di bawah kondisi tertentu (opsional).	Digunakan untuk menambahkan langkah atau perilaku tambahan ke <i>use case</i> dasar di bawah kondisi tertentu (opsional).
[20]Generalisasi (Garis berpanah segitiga) 	Menunjukkan hubungan pewarisan (<i>inheritance</i>). Simbol ini digunakan untuk menggambarkan hubungan antara <i>use case</i> atau aktor yang bersifat umum dengan yang bersifat lebih spesifik.	Digunakan untuk menggambarkan hubungan antara <i>use case</i> atau aktor yang bersifat umum dengan yang bersifat lebih spesifik.

Identifikasi aktor dan fungsionalitas secara akurat pada tahap awal perancangan merupakan langkah krusial untuk menentukan ruang lingkup operasional sistem [29], [33]. Dalam penerapannya, *use case diagram* juga dapat menggunakan relasi tambahan seperti *include* untuk menunjukkan ketergantungan fungsionalitas wajib, serta *extend* untuk fungsi yang bersifat opsional atau kondisional [28], [32]. Hal ini memastikan bahwa setiap proses bisnis, mulai dari pengelolaan data hingga pembuatan laporan, dapat teridentifikasi dengan jelas siapa yang berhak mengakses dan mengoperasikannya [30], [33], [35].

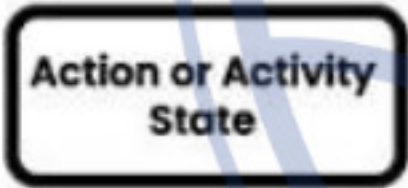
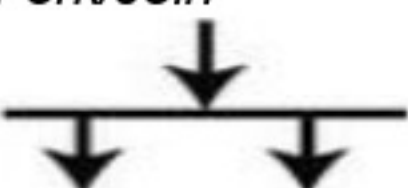
2. Activity Diagram

Activity Diagram merupakan diagram perilaku yang memvisualisasikan alur aktivitas atau proses bisnis yang terjadi di dalam sebuah sistem [28], [33]. Diagram ini memberikan gambaran mengenai pemodelan sekuensial dan algoritma melalui aliran

logika yang disesuaikan dengan berbagai kondisi operasional [29]. Fungsi utama dari diagram aktivitas adalah untuk menunjukkan urutan tindakan yang harus dilakukan oleh sistem, mulai dari proses masukan data, pengolahan, hingga penyajian informasi akhir [29], [33].

Secara visual, *activity diagram* memiliki kemiripan dengan *flowchart* konvensional, namun memiliki keunggulan dalam memodelkan perilaku dinamis yang melibatkan proses paralel, percabangan, maupun aktivitas yang berlangsung secara bersamaan [28] [29]. Diagram ini menggunakan sekumpulan notasi standar, seperti *initial state* (lingkaran hitam kecil) untuk menandai awal proses dan *final state* (*bull's eye*) untuk menandakan berakhirnya seluruh aliran aktivitas [29]. Setiap aktivitas di dalamnya biasanya diawali dengan kata kerja untuk merepresentasikan tindakan konkret yang dilakukan dalam aliran kerja sistem [29]. Berikut tabel simbol-simbol yang membangun struktur dalam *activity diagram* beserta deskripsinya [36].

Tabel 2.2 Simbol Komponen *Activity Diagram*

Simbol	Deskripsi
<i>Initial Node</i>  Initial State	Simbol visual umum dalam diagram aktivitas UML yang menandakan titik awal dimulainya suatu aliran kerja atau aktivitas. <i>Initial node</i> menggambarkan tahap awal atau simbol untuk memulai serangkaian tindakan pada diagram aktivitas. Suatu proses hanya memiliki satu kondisi awal, kecuali untuk menggambarkan aktivitas bersarang. Digambarkan dengan lingkaran hitam kecil
<i>Action/Activity</i>  Action or Activity State	Simbol yang digunakan dalam diagram aktivitas untuk merepresentasikan urutan tindakan, unit fungsionalitas, atau proses tertentu yang terjadi dalam sistem. <i>Activity</i> menggambarkan aktivitas yang terjadi atau dilakukan dalam sebuah aliran kerja dan menyimpan seluruh aktivitas pada sebuah sistem. Biasanya diawali dengan kata kerja berupa tindakan konkret.
<i>Decision</i> 	Simbol (biasanya berbentuk belah ketupat) yang menunjukkan titik di mana aliran aktivitas dapat bercabang berdasarkan kondisi atau pilihan tertentu. <i>Decision</i> digunakan untuk menunjukkan aliran logika yang disesuaikan dengan berbagai kondisi (variasi aliran logika). Diwakili oleh simbol belah ketupat yang memiliki satu aliran masuk dan beberapa aliran keluar berdasarkan syarat tertentu
<i>Fork/Join</i> 	Simbol visual dalam UML yang secara eksplisit mendukung aktivitas paralel dan sinkronisasi antar aktivitas dalam suatu aliran kerja. <i>Fork</i> digunakan untuk menunjukkan aktivitas yang bekerja secara paralel atau bersamaan, yang membedakan diagram aktivitas dengan <i>flowchart</i> biasa.
<i>Final Node</i>  Final State	Simbol visual umum dalam diagram aktivitas UML yang menandakan akhir dari seluruh aliran kerja atau penyelesaian aktivitas. <i>Final node</i> menandakan keadaan akhir dari suatu aktivitas sebagai penyelesaian semua aktivitas atau aliran proses. Digambarkan dengan lingkaran hitam di sekelilingnya (biasa disebut <i>bull's eye</i>).

Penggunaan *activity diagram* sangat efektif untuk memberikan pemahaman yang mendalam mengenai alur kerja sistem yang kompleks, sehingga memudahkan analisis dalam mengidentifikasi potensi ketidakefisienan atau tumpang tindih proses [33]. Melalui pemodelan ini, pengembang dapat merancang tata letak antarmuka menu dan alur navigasi aplikasi yang logis sesuai dengan kebutuhan tim kerja di lapangan [29]. Integrasi diagram aktivitas dengan hasil analisis *use case* menjamin bahwa setiap proses fungsional telah terencana secara matang sebelum memasuki tahap pengkodean [29], [33].

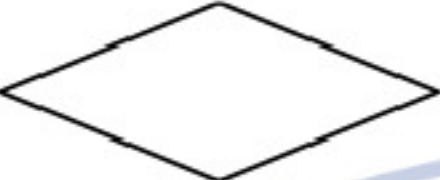
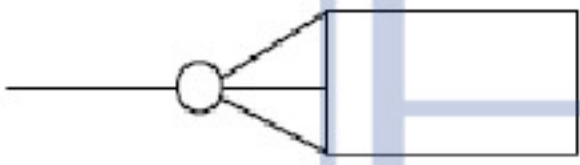
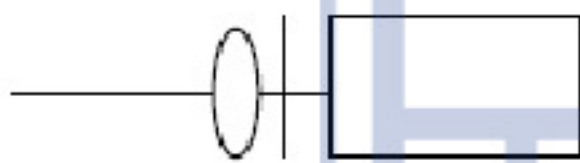
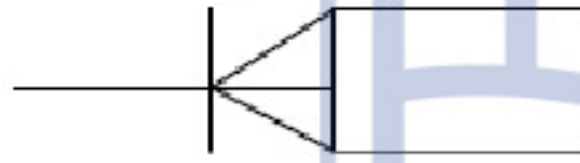
2.7 Entity Relationship Diagram

Entity Relationship Diagram (ERD) merupakan sebuah model data konseptual yang digunakan untuk menggambarkan struktur logis dari basis data secara grafis [37]. Secara historis, pendekatan ini diperkenalkan oleh Peter Chen pada tahun 1976 sebagai bentuk representasi abstrak dari data dan relasi antar data dalam sebuah sistem. ERD berfungsi sebagai alat komunikasi utama antara perancang sistem dengan pengguna akhir (*end-user*) untuk memverifikasi asumsi-asumsi yang muncul dalam proses pembuatan model, sekaligus memberikan pandangan tingkat tinggi mengenai elemen data dan interaksi yang terjadi di dalamnya [10].

Tujuan utama penggunaan ERD dalam perancangan basis data adalah untuk menghasilkan struktur data yang rapi, terukur, dan sistematis guna menghindari permasalahan klasik seperti inkonsistensi serta redundansi data [10]. Sebagai instrumen krusial dalam tahap desain, ERD memungkinkan pengembang untuk memetakan kebutuhan bisnis ke dalam bentuk entitas, atribut, dan relasi sebelum diimplementasikan ke dalam *Database Management System* (DBMS) [10], [38]. Selain itu, ERD berperan dalam proses *debugging* basis data apabila terjadi kesalahan pada keterhubungan data di kemudian hari [10].

Komponen utama penyusun ERD terdiri dari entitas, atribut, dan relasi yang masing-masing diwakili oleh simbol standar. Entitas merupakan gambaran objek dunia nyata yang dapat berupa individu, tempat, atau konsep, dan disimbolkan dengan bentuk persegi panjang. Atribut adalah identitas atau karakteristik yang melekat pada entitas untuk menginformasikan properti tertentu, yang dalam diagram dilambangkan dengan simbol elips. Relasi merupakan hubungan atau keterikatan antar entitas yang digambarkan dengan simbol belah ketupat [10]. Berikut tabel komponen dan atribut *entity relationship diagram* [39].

Tabel 2.3 Simbol Komponen ERD

Simbol	Deskripsi
Entitas (Kotak/Persegi Panjang) 	Entitas merupakan elemen utama dalam ERD untuk merepresentasikan tabel data. Menggambarkan objek atau hal yang dapat diidentifikasi dan dibedakan dalam konteks tertentu dalam basis data atau pemodelan.
Relationship/Hubungan (Belah Ketupat/Garis) 	<i>Relationship</i> menggambarkan interaksi atau keterhubungan antar entitas dalam skema <i>database</i> . Keterkaitannya beraneka ragam mulai dari <i>many-to-many</i> hingga <i>many-to-one</i> .
Atribut (Elips/Oval) 	Atribut menggambarkan suatu keadaan, properti, atau karakteristik yang melekat pada suatu entitas. Atribut digunakan untuk memanipulasi informasi spesifik dalam struktur sistem.
Kardinalitas relasi optional many 	Hubungan antar entity dengan derajat kardinalitas relasi optional many
Kardinalitas relasi optional one 	Hubungan antar entity dengan derajat kardinalitas relasi optional one
Kardinalitas relasi mandatory many 	Hubungan antar entity dengan derajat kardinalitas relasi mandatory many
Kardinalitas relasi mandatory one 	Hubungan antar entity dengan derajat kardinalitas relasi mandatory one

Dalam pendefinisian atribut, terdapat elemen penting yang disebut sebagai kunci primer (*primary key*), yaitu atribut unik yang digunakan untuk mengidentifikasi setiap catatan dalam entitas secara spesifik, seperti ISBN pada buku atau Nomor Pokok pada identitas pegawai. Selain itu, terdapat kunci asing (*foreign key*) yang berfungsi sebagai penghubung antar tabel untuk mewakili hubungan data yang saling berkaitan [10]. Atribut-atribut ini kemudian diorganisasikan melalui proses normalisasi untuk memastikan integritas data tetap terjaga [10], [12].

Derajat hubungan antar entitas diukur melalui kardinalitas, yang menentukan jumlah maksimum entitas yang dapat terlibat dalam sebuah hubungan. Terdapat tiga jenis kardinalitas utama, yaitu *one-to-one* (1:1) di mana satu baris data pada tabel A hanya terhubung ke satu baris di tabel B; *one-to-many* (1:N) di mana satu entitas dapat berelasi

dengan banyak entitas lainnya; dan *many-to-many* (M:N) yang menunjukkan hubungan timbal balik antar entitas dalam jumlah banyak. Penentuan derajat kardinalitas ini sangat bergantung pada hasil analisis terhadap proses bisnis yang sebenarnya di lapangan [10].

ERD yang disusun dengan baik harus mengikuti langkah-langkah sistematis, dimulai dari identifikasi entitas yang relevan, menentukan atribut kunci, hingga memetakan relasi antar entitas tersebut. Kelebihan ERD terletak pada kemampuannya memvisualisasikan koneksi data yang kompleks secara sederhana sehingga meminimalisir kesalahan desain awal yang berbiaya mahal. Namun, tantangan utama dalam model ini adalah tingkat ketelitian yang tinggi dalam menangani relasi yang sangat rumit agar tidak terjadi duplikasi informasi [10].

2.8 Aplikasi Web

Aplikasi berbasis *web* merupakan salah satu bentuk nyata dari pemanfaatan teknologi informasi yang mampu memproses serta menyajikan data secara efektif, cepat, dan akurat melalui jaringan internet atau intranet [14]. Secara mendasar, aplikasi *web* didefinisikan sebagai perangkat lunak yang dijalankan pada *server* dan dapat diakses oleh pengguna melalui *web browser* [10]. Karakteristik utamanya meliputi aksesibilitas yang tinggi, kemudahan distribusi, serta ketersediaan layanan yang dapat dijangkau dari berbagai lokasi selama terdapat koneksi internet [40], [41]. Dalam perspektif sistem informasi, aplikasi *web* berfungsi sebagai antarmuka yang mempertemukan pengguna dengan logika bisnis dan basis data guna mendukung proses pengambilan keputusan di dalam suatu organisasi [11], [14].

Struktur aplikasi *web* dibangun di atas beberapa blok bangunan (*building block*) utama yang meliputi blok masukan, blok model, blok keluaran, blok teknologi, blok basis data, dan blok kendali. Blok masukan berfungsi menangkap data melalui media digital, sementara blok model memanipulasi data tersebut menggunakan kombinasi prosedur dan logika matematika tertentu guna menghasilkan keluaran berupa informasi yang berkualitas bagi semua tingkatan manajemen. Blok teknologi bertindak sebagai alat bantu (*tool box*) yang menjalankan model dan mengintegrasikan seluruh komponen, sedangkan blok basis data menjadi repositori pusat bagi data yang saling berhubungan. Keberadaan blok kendali sangat krusial dalam arsitektur ini untuk melindungi integritas data dari ancaman sabotase, kecurangan, maupun kegagalan teknis lainnya [12].

Arsitektur aplikasi *web* modern umumnya menggunakan model tiga tingkat (*three-tier architecture*) yang membagi sistem menjadi lapisan presentasi (klien), lapisan aplikasi (*server*), dan lapisan basis data. Pada model ini, pengguna berinteraksi melalui antarmuka di

sisi klien yang berkomunikasi dengan *server* aplikasi sebagai lapisan perantara (*middleware*). *Server* aplikasi mengolah logika bisnis dan kemudian berinteraksi dengan sistem manajemen basis data (DBMS) untuk pengambilan atau penyimpanan informasi. Keuntungan utama dari arsitektur ini adalah skalabilitas yang baik serta keamanan yang lebih terjamin karena lapisan basis data tidak dapat diakses secara langsung oleh klien, melainkan harus melalui kebijakan keamanan yang diterapkan pada lapisan *middleware* [10].

Pengembangan aplikasi *web* dewasa ini didukung oleh berbagai teknologi seperti bahasa pemrograman PHP dan kerangka kerja (*framework*) seperti *CodeIgniter* untuk mempercepat proses pembangunan sistem yang terstruktur [40]. Di sisi penyimpanan data, MySQL sering digunakan sebagai DBMS karena kemampuannya dalam mengelola data dalam jumlah besar secara efisien [10], [40]. Selain itu, aspek keamanan menjadi fokus utama melalui implementasi algoritma enkripsi seperti *Advanced Encryption Standard* (AES) 256 untuk melindungi kerahasiaan dokumen negara dan data sensitif klien dari akses yang tidak sah [40]. Integrasi antara perangkat lunak pengembang seperti *Visual Studio Code* dengan lingkungan *server* lokal seperti XAMPP memungkinkan pengembang untuk membangun prototipe sistem yang matang sebelum diimplementasikan secara luas [14].

2.9 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) merupakan tahapan akhir dalam proses pengujian perangkat lunak sebelum sistem diluncurkan atau diimplementasikan secara luas [42], [43]. Pengujian ini dilakukan secara langsung oleh pengguna akhir (*end-user*) melalui interaksi dengan sistem untuk memverifikasi apakah fungsionalitas yang tersedia telah berjalan sesuai dengan kebutuhan dan spesifikasi yang telah disepakati [43], [44]. Melalui UAT, sistem dievaluasi untuk memastikan bahwa seluruh item dalam dokumen persyaratan (*requirement*) telah terpenuhi sehingga perangkat lunak tersebut dapat dinyatakan layak untuk diterima oleh pengguna [45], [46].

Tujuan utama dari pelaksanaan UAT adalah untuk menjamin bahwa sistem memenuhi kebutuhan dunia nyata pengguna dan relevan untuk mendukung operasional sehari-hari. Berbeda dengan metode pengujian teknis lainnya yang berfokus pada struktur kode atau performansi internal, UAT lebih menekankan pada aspek kegunaan (*usability*) dan tingkat kepuasan pengguna secara keseluruhan [47]. Proses ini berfungsi sebagai mekanisme validasi akhir untuk meyakinkan pemilik sistem maupun pelanggan bahwa aplikasi telah bekerja seperti yang diharapkan dan siap untuk masuk ke tahap produksi [43].

Peran UAT sangat krusial dalam pengujian sistem informasi karena melibatkan simulasi lingkungan kerja nyata yang memberikan kepercayaan kepada pemangku kepentingan terkait kesiapan sistem [43], [47]. UAT digunakan sebagai alat evaluasi untuk mengukur apakah sistem sudah memenuhi ekspektasi pengguna melalui umpan balik yang dikumpulkan dari responden [46], [44]. Hasil dari pengujian ini memberikan masukan berharga bagi tim pengembang untuk melakukan perbaikan fungsionalitas maupun desain sebelum sistem dirilis secara resmi, sehingga risiko terjadinya kegagalan sistem atau ketidaksesuaian fitur di masa depan dapat diminimalisir [47].

2.10 Skala Likert

Skala Likert adalah instrumen pengukuran yang dirancang untuk menghitung pendapat, sikap, dan persepsi seseorang atau sekelompok orang mengenai suatu fenomena sosial [42], [45]. Dalam kegiatan penelitian, skala ini berfungsi untuk menjabarkan variabel yang akan diukur menjadi indikator-indikator variabel yang lebih spesifik [45]. Indikator tersebut kemudian digunakan sebagai titik tolak untuk menyusun butir-butir instrumen berupa pernyataan atau pertanyaan dalam kuesioner yang akan disebarakan kepada responden [43], [46].

Penggunaan skala Likert umumnya menggunakan sistem penilaian lima tingkat yang memberikan fleksibilitas bagi responden untuk memilih tingkatan jawaban yang paling sesuai dengan persepsi mereka [43]. Tingkatan ini dimulai dari penilaian yang paling rendah hingga yang paling tinggi, yaitu Sangat Tidak Setuju (STS), Tidak Setuju (TS), Netral (N) atau Cukup, Setuju (S), dan Sangat Setuju (SS) [43], [44], [47]. Setiap pilihan jawaban memiliki bobot skor numerik tertentu, misalnya skor 1 untuk jawaban yang paling negatif hingga skor 5 untuk jawaban yang paling positif, sehingga data kualitatif dari persepsi pengguna dapat dikonversi menjadi data kuantitatif yang dapat dianalisis secara statistik [45], [47].

Dalam evaluasi sistem informasi, skala Likert banyak digunakan karena efektivitasnya dalam mengukur variabel-variabel subjektif seperti kepuasan pengguna dan penerimaan sistem [43]. Metode ini memungkinkan peneliti untuk menghitung persentase keberhasilan sistem berdasarkan rata-rata skor yang diperoleh, yang kemudian dikategorikan ke dalam rentang kualitas tertentu mulai dari sangat buruk hingga sangat baik [44], [46], [47]. Dengan hasil pengukuran yang terstruktur, pengembang sistem dapat mengidentifikasi parameter mana yang menjadi prioritas perbaikan guna meningkatkan kualitas interaksi dan pengalaman pengguna terhadap sistem yang dikembangkan [43].

2.11 Profil Mitra

Kantor Notaris/Pejabat Pembuat Akta Tanah (PPAT) Herli Tampubolon, S.H., M.Kn. merupakan kantor layanan jasa hukum yang bergerak di bidang kenotariatan dan pertanahan. Kantor ini pada awalnya didirikan di wilayah Tapanuli Utara tepatnya di Siborong Borong dengan fokus layanan sebagai Pejabat Pembuat Akta Tanah (PPAT), yang secara khusus menangani pembuatan akta-akta yang berkaitan dengan hak atas tanah.

Dalam perjalanan pengembangannya, setelah lebih kurang tiga tahun beroperasi, kantor ini berpindah lokasi ke Kabupaten Serdang Bedagai Seiring dengan perpindahan tersebut, kantor ini mengalami pengembangan fungsi dan kewenangan menjadi Kantor Notaris/PPAT. Transformasi ini memungkinkan perluasan ruang lingkup layanan, tidak hanya terbatas pada bidang pertanahan, tetapi juga mencakup berbagai aspek hukum perdata lainnya.

Keberadaan dan operasional kantor ini didukung oleh dasar hukum yang sah, sebagaimana tertuang dalam Keputusan Menteri Hukum Republik Indonesia tentang Pengangkatan Notaris serta Keputusan Menteri Agraria dan Tata Ruang/Kepala Badan Pertanahan Nasional tentang Pengangkatan Kembali sebagai Pejabat Pembuat Akta Tanah (PPAT). Dengan demikian, seluruh layanan yang diberikan memiliki kekuatan hukum yang jelas dan dapat dipertanggungjawabkan.

Di bawah kepemimpinan Herli Tampubolon, S.H., M.Kn., kantor ini senantiasa menjunjung tinggi prinsip profesionalitas, integritas, kehati-hatian, serta kepastian hukum dalam setiap pelaksanaan tugas jabatan. Pengalaman yang terbangun sejak awal berdiri hingga saat ini menjadi modal utama dalam memberikan pelayanan hukum yang berkualitas kepada masyarakat.



Gambar 2.2 Kantor Notaris/PPAT Herli Tampubolon, S.H., M.Kn.

Adapun ruang lingkup layanan yang disediakan meliputi:

1. Pembuatan akta autentik di bidang pertanahan, seperti jual beli, hibah, tukar-menukar, dan pembagian hak bersama;
2. Pendirian dan perubahan badan hukum, termasuk Perseroan Terbatas (PT), Yayasan, dan Commanditaire Vennootschap (CV);
3. Penyusunan dan pembuatan berbagai perjanjian hukum;
4. Pembuatan akta perjanjian pra-nikah;
5. Legalisasi, waarmeding, dan pengesahan dokumen;
6. Layanan kenotariatan lainnya sesuai dengan kewenangan yang diatur dalam peraturan perundang-undangan.

Dengan komitmen untuk memberikan pelayanan yang profesional, akuntabel, serta berorientasi pada kepuasan klien, Kantor Notaris/PPAT Herli Tampubolon, S.H., M.Kn. hadir sebagai mitra terpercaya dalam memberikan solusi hukum yang komprehensif dan berkelanjutan bagi masyarakat.

