

BAB II

KAJIAN LITERATUR

2.1 Konsep Dasar Sistem Informasi

Sistem merupakan sekumpulan elemen yang saling berhubungan dan saling berinteraksi dalam suatu kesatuan untuk mencapai tujuan tertentu. Dalam suatu organisasi, sistem dibutuhkan agar seluruh aktivitas dapat berjalan secara terarah, terkoordinasi, dan menghasilkan keluaran yang sesuai dengan kebutuhan. Sistem tidak hanya dipahami sebagai kumpulan bagian-bagian yang berdiri sendiri, tetapi sebagai susunan komponen yang memiliki hubungan kerja dan fungsi tertentu sehingga mampu membentuk satu kesatuan kerja yang utuh [1].

Secara umum, sistem dapat diartikan sebagai jaringan kerja dari prosedur-prosedur yang saling berkaitan, yang disusun untuk melaksanakan suatu kegiatan atau untuk mencapai sasaran tertentu. Dalam konteks pengembangan aplikasi *payroll*, sistem dibutuhkan untuk mengatur alur pengelolaan data karyawan, data absensi, komponen penggajian, perhitungan gaji, hingga pelaporan secara terintegrasi [2].

Pada prinsipnya, sistem dapat dibedakan menjadi sistem terbuka dan sistem tertutup. Sistem terbuka adalah sistem yang berhubungan dengan lingkungan luarnya dan menerima pengaruh dari luar, sedangkan sistem tertutup adalah sistem yang bekerja lebih terbatas dan tidak banyak dipengaruhi oleh lingkungan luar. Dalam penerapannya di perusahaan, sistem informasi *payroll* termasuk ke dalam sistem terbuka karena menerima *input* berupa data kehadiran, data jabatan, dan data penggajian yang diproses untuk menghasilkan informasi yang dibutuhkan manajemen [6].

Suatu sistem memiliki karakteristik yang dapat dikenali, antara lain adanya komponen, lingkungan luar sistem, batas sistem, penghubung sistem, masukan, keluaran, pengolah, dan sasaran sistem. Komponen sistem menunjukkan adanya bagian-bagian yang membentuk satu kesatuan. Lingkungan luar sistem adalah segala sesuatu di luar sistem yang dapat memengaruhi operasi sistem. Batas sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau dengan lingkungan luar. Penghubung sistem berfungsi menghubungkan subsistem satu dengan subsistem lainnya. Masukan adalah data atau energi yang dimasukkan ke dalam sistem, sedangkan keluaran adalah hasil pengolahan sistem. Pengolah merupakan bagian yang mengubah *input* menjadi *output*, dan sasaran sistem

adalah tujuan yang ingin dicapai. Karakteristik ini menunjukkan bahwa suatu sistem harus dirancang secara jelas agar dapat mendukung tujuan organisasi secara efektif [7].

Informasi adalah data yang telah diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya. Data pada dasarnya masih berupa fakta mentah yang belum dapat digunakan secara langsung dalam pengambilan keputusan, sedangkan informasi merupakan hasil pengolahan data yang telah memiliki nilai, arti, dan manfaat. Dalam lingkungan perusahaan, informasi memiliki peran penting karena digunakan untuk mendukung aktivitas operasional, pengawasan, evaluasi, dan pengambilan keputusan manajerial [8].

Informasi dapat dikatakan bernilai apabila mampu memberikan manfaat bagi penerimanya, tepat waktu, akurat, relevan, dan dapat dipercaya. Sumber informasi biasanya berasal dari data yang diproses melalui suatu siklus pengolahan data, yaitu dari *input*, proses, hingga *output*. Berdasarkan sifatnya, informasi dapat dibedakan menjadi beberapa jenis seperti *absolute information*, *substitutional information*, *philosophic information*, *subjective information*, *objective information*, dan *cultural information* [9].

Sistem informasi adalah sistem yang terdapat dalam organisasi untuk mempertemukan kebutuhan pengolahan transaksi harian, mendukung kegiatan operasional, mendukung fungsi manajerial dan strategi organisasi, serta menyediakan laporan bagi pihak tertentu yang membutuhkan [6]. Sistem informasi adalah kombinasi teratur dari manusia, perangkat keras, perangkat lunak, dan basis data yang berfungsi untuk mengumpulkan, mengubah, serta menyebarkan informasi dalam suatu organisasi [2].

Fungsi sistem informasi adalah untuk mengumpulkan, mengolah, menyimpan, dan menyajikan data menjadi informasi yang bermanfaat bagi pengguna dalam mendukung kegiatan operasional dan pengambilan keputusan [7], sedangkan tujuan sistem informasi adalah untuk menghasilkan informasi yang akurat, relevan, tepat waktu, dan bermanfaat bagi pengguna dalam mendukung kegiatan operasional, pengendalian, evaluasi, serta pengambilan keputusan dalam suatu organisasi.

Komponen dalam sistem informasi terdiri dari *hardware*, *software*, personil, data, dan prosedur. *Hardware* berfungsi sebagai perangkat fisik untuk menjalankan sistem, *software* menjadi alat pemroses data, personil bertugas mengoperasikan dan menggunakan sistem, data menjadi bahan utama yang diolah, dan prosedur menjadi pedoman kerja sistem. Seluruh komponen ini harus berjalan bersama agar sistem informasi dapat menghasilkan keluaran yang bermanfaat [8].

2.2 Aplikasi Berbasis Web

Aplikasi berbasis web adalah sistem aplikasi yang memiliki komponen di sisi klien dan komponen di sisi *server* untuk mengolah data. Aplikasi ini memanfaatkan layanan web dengan arsitektur *client-server*, *request-response model*, dan standar HTTP. Dalam perkembangannya, aplikasi web dibedakan menjadi web statis dan web dinamis. *Web* dinamis memungkinkan pengguna melakukan interaksi dengan sistem, sedangkan web statis hanya menampilkan informasi tanpa interaksi yang kompleks. Oleh karena itu, aplikasi berbasis web sangat sesuai digunakan dalam pengembangan sistem *payroll* karena memungkinkan pengelolaan data secara interaktif melalui *browser* [10].

Aplikasi berbasis web merupakan sistem yang dijalankan melalui jaringan internet atau *server* lokal dengan menggunakan *browser* sebagai media akses utama. Aplikasi berbasis web dipahami sebagai solusi digital untuk menggantikan proses manual yang sebelumnya menggunakan kertas, arsip fisik, atau file terpisah [10].

Secara fungsional, aplikasi berbasis web memiliki keunggulan karena mampu menyatukan proses penginputan, pengolahan, penyimpanan, pencarian, dan pelaporan data dalam satu sistem yang terintegrasi. Dalam konteks pengembangan sistem, aplikasi berbasis web umumnya dirancang dengan metode pengembangan perangkat lunak tertentu, seperti *Waterfall*, *Prototype*, *Agile*, atau *Incremental Development*. Pemilihan metode tersebut bertujuan agar proses pengembangan berjalan lebih sistematis [10].

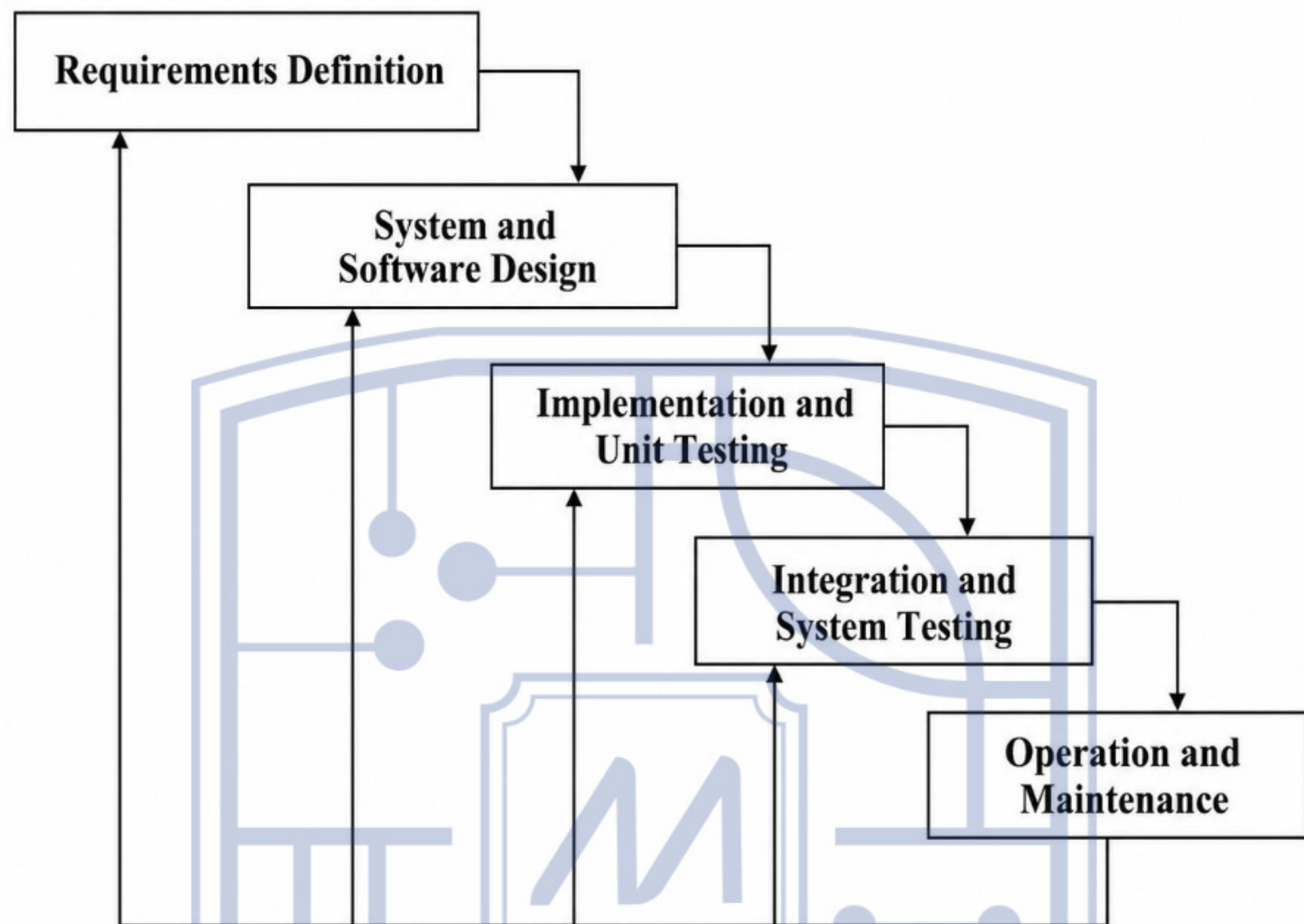
2.3 Metode Pengembangan Sistem (*System Development Life Cycle*)

System Development Life Cycle atau SDLC adalah siklus hidup pengembangan sistem yang digunakan dalam rekayasa sistem dan rekayasa perangkat lunak. SDLC merupakan proses pembuatan dan perubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem tersebut. Tahapan dalam SDLC umumnya meliputi perencanaan, analisis, desain, implementasi, pengujian, dan pemeliharaan. Salah satu metode SDLC yang banyak digunakan adalah metode *Waterfall* [11].

2.3.1 Metode *Waterfall*

Waterfall adalah model pengembangan perangkat lunak yang dilakukan secara sekuensial atau berurutan dari atas ke bawah seperti air terjun. Metode ini terdiri dari lima fase utama yang harus diselesaikan secara bertahap, yaitu *requirements definition*, *system*

and software design, implementation and unit testing, integration and system testing, serta operation and maintenance [12].



Gambar 2.1 Bagan Metode *Waterfall* [11]

1. **Requirements Definition**

Pada tahap ini dilakukan analisis kebutuhan untuk memahami kebutuhan yang diharapkan oleh pengguna dan batasan-batasan dalam pengembangan perangkat lunak. Dalam penelitian ini, kebutuhan sistem mencakup pengelolaan data karyawan, data absensi, data penggajian, perhitungan gaji, slip gaji, dan laporan penggajian. Tahap ini sangat penting karena menjadi dasar bagi seluruh proses pengembangan sistem [13].

2. **System and Software Design**

Pada tahap ini pengembang membuat desain sistem yang akan membantu menentukan kebutuhan perangkat keras, kebutuhan perangkat lunak, struktur basis data, rancangan antarmuka, dan arsitektur sistem secara keseluruhan. Dalam pengembangan aplikasi *payroll* berbasis web, tahap desain akan menghasilkan gambaran sistem yang akan memudahkan proses implementasi selanjutnya [14].

3. **Implementation and Unit Testing**

Pada tahap ini desain sistem diterjemahkan ke dalam kode program. Sistem dikembangkan ke dalam modul-modul kecil yang kemudian diuji secara terpisah melalui *unit testing* untuk memastikan setiap modul berjalan sesuai fungsi yang dirancang. Tahap ini menjadi inti dari proses pembangunan aplikasi karena rancangan yang sebelumnya masih berupa konsep diwujudkan menjadi program nyata [11]. Dalam aplikasi *payroll*, *Black Box Testing* dapat diterapkan pada proses *login*, *input* data karyawan, *input* data absensi, perhitungan gaji, dan pembuatan laporan [15].

4. **Integration and System Testing**

Pada tahap ini seluruh modul yang telah dibuat digabungkan menjadi satu sistem utuh, kemudian dilakukan pengujian untuk melihat apakah sistem telah berjalan sesuai desain dan kebutuhan pengguna. Pengujian sistem penting dilakukan agar pengembang dapat memastikan bahwa interaksi antarbagian sistem berjalan secara benar dan tidak menimbulkan kesalahan fungsional [16].

5. **Operation and Maintenance**

Tahap ini merupakan tahap akhir dari metode *Waterfall*. Sistem yang telah selesai dikembangkan mulai dioperasikan dan selanjutnya dilakukan pemeliharaan. Pemeliharaan mencakup perbaikan kesalahan yang belum ditemukan sebelumnya, penyesuaian sistem terhadap kebutuhan baru, serta peningkatan performa sistem agar tetap berjalan dengan baik dalam penggunaan jangka panjang [12].

2.3.2 **Blackbox Testing**

Black Box Testing merupakan metode pengujian yang digunakan untuk memastikan fungsi aplikasi telah sesuai dengan kebutuhan pengguna. Pengujian ini dilakukan dengan melihat *input* dan *output* dari setiap alur sistem, tanpa mengharuskan penguji memahami kode program yang digunakan [17]. *Black Box Testing* adalah pengujian perangkat lunak yang berfokus pada spesifikasi fungsional sistem. Metode ini memungkinkan penguji menentukan kondisi *input* dan melakukan pengujian terhadap fungsi program tanpa melihat struktur internal perangkat lunak [18].

Box Testing adalah metode pengujian yang menilai fungsionalitas sistem dari sudut pandang pengguna akhir tanpa memperhatikan struktur internal kode program. Metode ini digunakan untuk memastikan setiap fitur berjalan sesuai spesifikasi dan kebutuhan pengguna [19]. *Black Box Testing* merupakan bagian dari proses pengujian perangkat lunak yang dilakukan untuk menilai apakah sistem dapat berjalan sesuai fungsi yang diharapkan.

Pengujian ini berfokus pada proses *testing* dari sisi luar sistem sehingga dapat digunakan untuk menemukan kesalahan fungsi sebelum perangkat lunak digunakan [20].

Manfaat *Black Box Testing* secara umum adalah untuk memastikan bahwa perangkat lunak berjalan sesuai dengan fungsi dan kebutuhan pengguna tanpa perlu melihat struktur internal atau kode program. Pengujian ini membantu penguji memeriksa kesesuaian antara *input* yang diberikan dengan *output* yang dihasilkan, sehingga dapat diketahui apakah sistem telah bekerja sesuai spesifikasi yang ditentukan [17].

Selain itu, *Black Box Testing* bermanfaat untuk menemukan kesalahan pada fungsi, antarmuka, alur proses, validasi data, serta respons sistem terhadap berbagai kondisi *input*. Dengan adanya pengujian ini, potensi *error* atau *bug* dapat diketahui lebih awal sebelum perangkat lunak digunakan oleh pengguna akhir [19].

2.3.3 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) merupakan tahapan krusial dan tahap akhir dalam siklus hidup pengembangan sistem perangkat lunak sebelum sistem tersebut diluncurkan secara resmi (*deployment*). UAT dilakukan langsung oleh pengguna akhir (*end-user*) atau klien untuk memvalidasi apakah sistem yang telah dibangun dapat beroperasi dengan baik di lingkungan kerja nyata dan telah memenuhi seluruh kebutuhan bisnis serta spesifikasi awal yang disepakati [21]. Berbeda dengan *Black Box Testing* yang dilakukan oleh pengembang untuk mencari bug secara teknis fungsional, UAT lebih difokuskan pada pengujian kelayakan dan penerimaan fungsionalitas aplikasi dari sudut pandang alur kerja pengguna sesungguhnya.

Tujuan utama dari pelaksanaan UAT adalah untuk memberikan keyakinan (*confidence*) kepada klien bahwa perangkat lunak yang diserahkan tidak hanya terbebas dari kesalahan kritis, tetapi juga memberikan solusi yang praktis dan efisien terhadap masalah bisnis yang ada. Proses UAT umumnya mencakup skenario pengujian bisnis (*business test cases*) yang dijalankan pada sistem. Jika perangkat lunak berhasil melewati tahap ini tanpa adanya feedback negatif yang signifikan, maka perangkat lunak tersebut dinyatakan layak untuk diimplementasikan ke tahap *production*[22]

Dalam pelaksanaannya, evaluasi UAT sering kali diukur menggunakan instrumen kuesioner yang dibagikan kepada pengguna yang berpartisipasi dalam pengujian. Salah satu metode pengukuran respons dan tingkat kepuasan pengguna yang paling umum digunakan adalah Skala Likert. Skala Likert adalah skala psikometrik yang digunakan untuk mengukur

sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang suatu fenomena atau sistem. Dalam pengujian sistem, Skala Likert umumnya menggunakan lima tingkat penilaian, mulai dari "Sangat Tidak Setuju" (skor 1) hingga "Sangat Setuju" (skor 5), yang kemudian dikonversikan menjadi persentase tingkat kelayakan sistem [23]. Rumus yang digunakan untuk menghitung persentase kelayakan sistem berdasarkan hasil UAT adalah sebagai berikut:

$$\text{Persentase UAT} = (\text{Total Skor} \div \text{Skor Maksimum}) \times 100\% \dots\dots\dots(1)$$

Dimana:

Total Skor = jumlah skor jawaban seluruh responden

Skor Maksimum = (jumlah pernyataan $\times$ jumlah responden $\times$ bobot tertinggi)

2.4 Fishbone Diagram

Diagram tulang ikan atau fishbone diagram adalah salah satu metode untuk menganalisa penyebab dari sebuah masalah atau kondisi. Sering juga diagram ini disebut dengan diagram sebab-akibat atau cause effect diagram. Diagram sebab akibat berkaitan dengan pengendalian proses statistik, di mana dapat mengidentifikasi penyebab suatu proses out of control. Artinya, diagram sebab akibat ini dipergunakan untuk menunjukkan faktor-faktor penyebab (sebab) dan karakteristik kualitas (akibat) yang disebabkan oleh faktor-faktor penyebab itu. Diagram sebab akibat ini sering juga disebut sebagai Diagram Tulang Ikan (*Fishbone* Diagram) karena bentuknya seperti kerangka ikan. Diagram ini juga dikenal dengan diagram Ishikawa, yang pertama kali diperkenalkan oleh Prof. Kaoru Ishikawa dari Universitas Tokyo pada tahun 1953.

Kegunaan dari diagram *Fishbone* ini adalah diagram fishbone dapat digunakan untuk menganalisis permasalahan baik pada level individu, tim, maupun organisasi. Manfaat penggunaan diagram fishbone ini antara lain: [24]

- a. Memfokuskan individu, tim, atau organisasi pada permasalahan utama. Penggunaan diagram fishbone dalam tim/organisasi untuk menganalisis permasalahan akan membantu anggota tim dalam memfokuskan permasalahan pada masalah prioritas.
- b. Memudahkan dalam mengilustrasikan gambaran singkat permasalahan tim/organisasi. Diagram fishbone dapat mengilustrasikan permasalahan utama secara ringkas sehingga tim akan mudah menangkap permasalahan utama.

- c. Menentukan kesepakatan mengenai penyebab suatu masalah. Dengan menggunakan teknik brainstorming pada anggota tim akan memberikan pendapat mengenai penyebab munculnya masalah. Berbagai pendapat ini akan didiskusikan untuk menentukan mana dari penyebab tersebut yang berhubungan dengan masalah utama termasuk menentukan penyebab yang dominan.
- d. Membangun dukungan anggota tim untuk menghasilkan solusi. Setelah ditentukan penyebab dari masalah, langkah untuk menghasilkan solusi akan lebih mudah mendapat dukungan dari anggota tim.

Fungsi dasar diagram fishbone (tulang ikan) atau cause and effect (sebab dan akibat) adalah untuk mengidentifikasi dan mengorganisasi penyebab-penyebab yang mungkin timbul dari suatu efek spesifikasi dan kemudian memisahkan akar penyebabnya. *Fishbone* diagram sendiri banyak digunakan untuk membantu mengidentifikasi akar penyebab dari suatu masalah dan membantu menemukan ide-ide untuk solusi suatu masalah. [24]

Berikut ini tahapan yang dilakukan dalam menyusun fishbone diagram yaitu sebagai berikut: [24]

- a. Membuat kerangka fishbone diagram. Kerangka meliputi kepala ikan yang diletakkan pada bagian kanan diagram dan tulang ikan pada bagian kiri diagram. Kepala ikan digunakan untuk menyatakan masalah utama. Bagian sirip digunakan untuk menuliskan kategori penyebab permasalahan. Bagian tulang digunakan untuk menyatakan penyebab masalah. Bagian duri ikan digunakan untuk menyatakan sub penyebab masalah.
- b. Merumuskan masalah utama. Masalah utama ditempatkan pada bagian kanan dari fishbone diagram atau ditempatkan pada kepala ikan.
- c. Langkah berikutnya adalah mencari faktor-faktor utama yang menyebabkan munculnya masalah utama. Kategori penyebab masalah ini ditempatkan di fishbone diagram pada bagian sirip ikan.
- d. Menemukan penyebab untuk masing-masing kategori penyebab masalah. Penyebab ini ditempatkan pada bagian duri ikan.
- e. Langkah selanjutnya setelah masalah dan penyebab atau sub penyebab masalah diketahui, dapat menggambarannya pada fishbone diagram.

2.5 Performance, Information, Economics, Control Efficiency, Services (PIECES)

Analisa PIECES merupakan suatu alat dalam menganalisis sistem informasi berbasis komputer, di mana terdiri dari poin-poin penting yang berguna untuk dijadikan pedoman

atau acuan dalam menganalisis sistem tersebut. Terdapat 6 kriteria pada analisis PIECES yaitu: [6]

a. *Performance*

Keandalan suatu sistem merupakan variabel pertama dari PIECES di mana memiliki peranan penting untuk melihat sejauh mana dan seberapa handal suatu sistem informasi memproses atau mengelola data untuk menghasilkan informasi dan tujuan yang diharapkan.

b. *Information*

Informasi dan data yang disajikan ataupun dibutuhkan oleh perusahaan merupakan salah satu faktor penting untuk kemajuan perusahaan. Informasi yang dihasilkan sistem informasi harus memiliki nilai yang berguna untuk pengambilan keputusan oleh manajemen perusahaan.

c. *Economics*

Ekonomi menjadi parameter apakah dengan pengorbanan perusahaan untuk mengaplikasikan sistem informasi sepadan dengan hasil yang diperoleh perusahaan.

d. *Controls*

Suatu sistem jika tidak disertai dengan pengendalian dan pengamanan yang baik, akan menjadi suatu sistem yang sangat lemah sehingga pihak dari luar sistem sangat mudah untuk masuk dan mengacaukan sistem tersebut. Oleh karena itu, perlu adanya suatu pengendalian dan pengamanan terhadap suatu sistem informasi dengan memperhatikan hal-hal yang terkait pengendalian dan pengamanan sistem.

e. *Efficiency*

Sistem informasi yang digunakan secara mutlak harus memiliki nilai keunggulan jika dibandingkan dengan penggunaan sistem secara manual. Keunggulan tersebut terletak pada tingkat keefisienan saat sistem informasi tersebut beroperasi.

f. *Services*

Layanan merupakan aspek penting yang diberikan oleh sistem. Maka dari itu layanan yang disediakan sistem apakah memiliki kekurangan yang perlu diperbaiki atau ditambahkan layanan baru sesuai dengan kebutuhan saat ini.

2.6 Konsep Penggajian

2.6.1 Gaji

Gaji adalah balas jasa dalam bentuk uang yang diterima karyawan sebagai konsekuensi atas statusnya sebagai pekerja yang memberikan kontribusi bagi pencapaian tujuan perusahaan. Gaji juga dapat dipahami sebagai bayaran tetap yang diterima seseorang karena kedudukannya dalam perusahaan. Dalam konteks penelitian ini, gaji merupakan elemen utama yang dikelola dalam aplikasi *payroll* karena berkaitan langsung dengan hak karyawan dan kewajiban perusahaan [3].

Dalam upah atau gaji terdapat beberapa komponen utama, antara lain gaji pokok, tunjangan, potongan, uang lembur, dan insentif. Gaji pokok merupakan dasar gaji yang ditentukan sesuai jabatan atau tugas tertentu. Tunjangan merupakan pelengkap dari gaji pokok. Potongan adalah pengurangan penghasilan karyawan yang dapat berasal dari pajak, pinjaman, asuransi, atau absensi. Uang lembur merupakan tambahan gaji karena bekerja di luar jam kerja resmi. Sementara itu, insentif adalah pemberian kompensasi tambahan oleh perusahaan untuk memotivasi pegawai agar meningkatkan prestasi kerjanya [25].

2.6.2 Aturan Jam Kerja

Aturan jam kerja merupakan ketentuan mengenai lamanya waktu kerja yang wajib dipenuhi oleh pekerja dalam satu hari atau satu minggu sesuai peraturan ketenagakerjaan. Pada umumnya waktu kerja dibedakan menjadi tujuh jam per hari dan empat puluh jam per minggu untuk enam hari kerja, atau delapan jam per hari dan empat puluh jam per minggu untuk lima hari kerja. Ketentuan ini menjadi dasar penting dalam perhitungan kehadiran, keterlambatan, dan lembur pada sistem penggajian [26].

Selain waktu kerja normal, terdapat pula ketentuan mengenai kerja lembur. Pekerja yang bekerja melebihi waktu kerja normal harus memenuhi syarat tertentu, seperti adanya persetujuan pekerja dan adanya kewajiban perusahaan untuk membayar upah kerja lembur. Dalam pengembangan aplikasi *payroll*, aturan jam kerja dan lembur perlu diperhatikan agar sistem dapat menghitung hak karyawan secara benar dan sesuai ketentuan yang berlaku [27].

2.6.3 Absensi

Absensi merupakan daftar kehadiran karyawan yang berisi catatan jam datang dan jam pulang. Absensi sangat penting dalam sistem penggajian karena menjadi dasar untuk

mengetahui kehadiran, ketidakhadiran, keterlambatan, dan waktu lembur karyawan. Dengan adanya data absensi yang akurat, perusahaan dapat menghitung gaji secara lebih tepat dan objektif [28].

Dalam perusahaan, absensi juga menggambarkan hubungan antara kewajiban kerja karyawan dan hak yang akan diterimanya. Tingkat absensi akan berpengaruh pada potongan ataupun tambahan penghasilan tertentu. Oleh sebab itu, pengelolaan absensi yang semula dilakukan secara manual sebaiknya didukung oleh sistem yang terkomputerisasi agar proses penggajian menjadi lebih cepat, tertib, dan akurat [24]

2.7 Pemodelan Sistem

Pemodelan sistem merupakan tahapan yang penting dalam pengembangan perangkat lunak karena digunakan untuk menggambarkan kebutuhan sistem, alur proses, interaksi pengguna, serta struktur data yang akan dibangun. Melalui pemodelan sistem, pengembang dapat memperoleh gambaran yang jelas mengenai bagaimana sistem akan bekerja sebelum masuk ke tahap implementasi, sehingga proses pengembangan dapat dilakukan secara lebih terarah dan sistematis [9].

Pemodelan sistem juga membantu meminimalkan kesalahan dalam tahap perancangan karena kebutuhan pengguna telah divisualisasikan ke dalam bentuk model yang mudah dipahami. Selain itu, pemodelan sistem dapat mempermudah komunikasi antara pengembang dengan pengguna, karena rancangan sistem dapat dijelaskan secara visual dan terstruktur [29].

2.7.1 Use Case Diagram

Use Case Diagram merupakan diagram yang digunakan untuk menggambarkan hubungan antara aktor dengan sistem serta fungsi-fungsi utama yang dapat dijalankan oleh masing-masing aktor. Diagram ini berperan dalam menunjukkan kebutuhan sistem dari sudut pandang pengguna, sehingga sangat membantu dalam tahap analisis kebutuhan aplikasi yang akan dibangun [30].

Use Case Diagram adalah salah satu alat pemodelan sistem yang digunakan untuk menggambarkan interaksi antara pengguna dan sistem. Penggunaan Diagram *Use Case* dapat memberikan gambaran yang efektif dalam menjelaskan konteks suatu sistem dengan jelas, sehingga memperlihatkan batasan sistem tersebut dengan jelas. Adapun, fungsi dari *Use Case Diagram* sebagai berikut [29]:

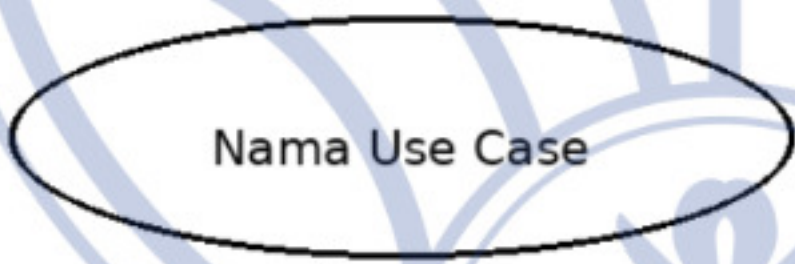
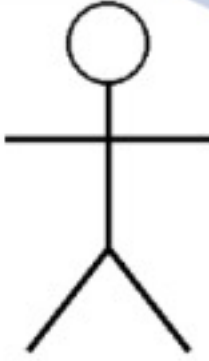
1. Berguna memperlihatkan proses aktivitas secara urut dalam sistem.
2. Mampu menggambarkan proses bisnis, bahkan menampilkan urutan aktivitas pada sebuah proses.
3. Sebagai *bridge* atau jembatan antara pembuat dengan konsumen untuk mendeskripsikan sebuah sistem.

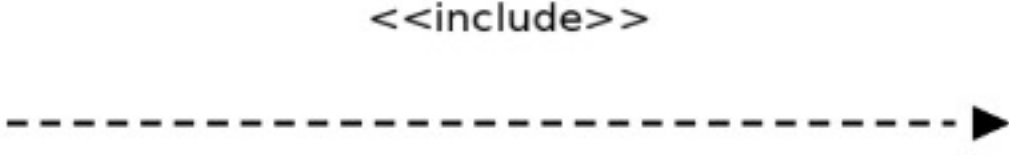
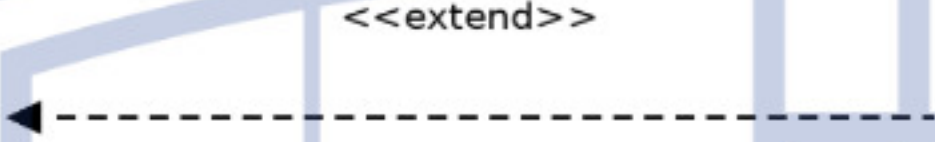
Use Case Diagram akan sangat membantu kita dalam menyusun kebutuhan-kebutuhan (*requirement analisis*) sebuah sistem. *Use case diagram* akan digunakan untuk mengkomunikasikan rancangan sistem dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem [31].

Manfaat dari *Use Case* di antaranya [30]:

1. Menggunakannya sebagai kebutuhan verifikasi.
2. Menjadi gambaran *interface* dari sebuah sistem karena setiap sistem yang dibangun haruslah memiliki *interface*.
3. Mengidentifikasi siapa saja orang yang dapat berinteraksi dengan sistem, serta apa yang dapat dilakukan oleh sistem.
4. Memberikan kepastian mengenai kebutuhan sistem, sehingga tidak membingungkan.
5. Memudahkan proses komunikasi antara *domain expert* dan *end user*.

Tabel 2.1 Simbol Use Diagram

Nama	Simbol	Deskripsi
<i>Use Case</i>		Simbol yang menggambarkan fungsi dari sistem, umumnya diekspresikan dalam bentuk kata kerja di awal setiap frasa.
Aktor		Simbol yang mewakili pengguna yang terlibat dalam <i>Use Case</i> , biasanya menyatakan orang-orang yang berinteraksi dengan sistem.
Asosiasi		Simbol yang menghubungkan antara <i>Use Case</i> dan aktor, menggambarkan komunikasi antara keduanya. Asosiasi

		digunakan untuk menggambarkan hubungan statis antara elemen-elemen tersebut.
<i>Include</i>		Simbol ini menggambarkan bahwa <i>Use Case</i> secara keseluruhan merupakan bagian dari fungsionalitas <i>Use Case</i> lainnya.
<i>Extend</i>		Simbol yang serupa dengan <i>include</i> , namun menunjukkan bahwa sebuah <i>Use Case</i> tambahan dapat berdiri sendiri tanpa membutuhkan <i>Use Case</i> yang ditambahkan.
Sistem		Simbol yang membatasi bahwa setiap interaksi dengan simbol lain harus berada di dalamnya.

2.7.2 Activity Diagram

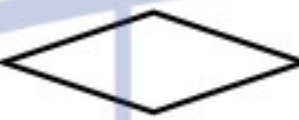
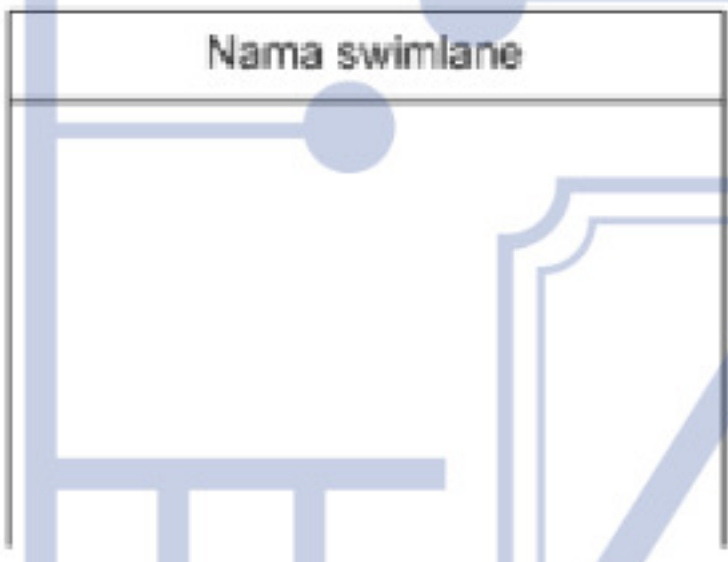
Activity Diagram adalah diagram yang digunakan untuk menggambarkan alur aktivitas atau urutan proses dalam suatu sistem mulai dari awal hingga akhir. Diagram ini membantu dalam memahami urutan kegiatan yang dilakukan oleh pengguna maupun sistem, sehingga sangat berguna untuk menjelaskan alur kerja aplikasi yang akan dibangun [29].

Activity diagram digunakan untuk menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Activity diagram sangat bermanfaat dalam memodelkan sebuah proses untuk membantu dalam memahami proses tersebut secara keseluruhan. Diagram aktivitas atau activity diagram menggambarkan workflow (alirankerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang adapada perangkat lunak [29].

Activity diagram merupakan bentuk khusus dari state machine yang bertujuan memodelkan komputasi-komputasi dan aliran-aliran kerja yang terjadi dalam sistem/perangkat lunak yang sedang dikembangkan [29].

Tabel 2.2 *Activity Diagram*

Nama	Simbol	Deskripsi
------	--------	-----------

<i>Initial Node</i>		Simbol yang menunjukkan titik awal dimulainya suatu aktivitas dalam <i>Activity Diagram</i> .
<i>Activity / Action</i>		Simbol yang menggambarkan aktivitas atau tindakan yang dilakukan dalam suatu proses.
<i>Control Flow</i>		Simbol yang menunjukkan arah alur proses atau urutan aktivitas dari satu kegiatan ke kegiatan berikutnya.
<i>Decision Node</i>		Simbol yang digunakan untuk menunjukkan percabangan keputusan berdasarkan kondisi tertentu.
<i>Swimlane</i>		Simbol yang digunakan untuk memisahkan tanggung jawab aktivitas berdasarkan aktor, bagian, atau entitas yang terlibat dalam proses.
<i>Final Node</i>		Simbol yang menunjukkan akhir dari suatu aktivitas atau proses dalam <i>Activity Diagram</i> .

2.7.3 Entity Relationship Diagram (ERD)

diagram *Entity Relationship Diagram* (ERD) adalah suatu pemodelan basis data yang digunakan untuk menggambarkan struktur data serta relasi antar entitas (tabel) di dalam sebuah sistem. Diagram ini berfungsi untuk menunjukkan bagaimana data diorganisasikan secara logis dan saling terhubung satu sama lain dalam sebuah basis data relasional (RDBMS) [6].

Komponen utama pembentuk ERD menurut Al Fatta [6] antara lain:

1. **Entitas (Entity):** Kumpulan dari objek, orang, atau kejadian yang dapat diidentifikasi secara unik di dalam sistem. Simbol entitas umumnya digambarkan dengan bentuk persegi Panjang [6].
2. **Atribut (Attribute):** Karakteristik atau properti dari entitas yang menjelaskan detail data tersebut (misalnya nama, NIP, alamat). Atribut yang menjadi kunci utama (Primary Key) selalu ditandai dengan garis bawah [6].

3. **Relasi (Relationship):** Hubungan logika yang terjadi antara satu atau beberapa entitas (misalnya karyawan memiliki jabatan). Relasi umumnya digambarkan dengan bentuk belah ketupat [6].
4. **Garis (Link):** Digunakan untuk menghubungkan atribut dengan entitas, serta menghubungkan entitas dengan relasi [6].

2.8 Profil PT Star Samudera Logistik

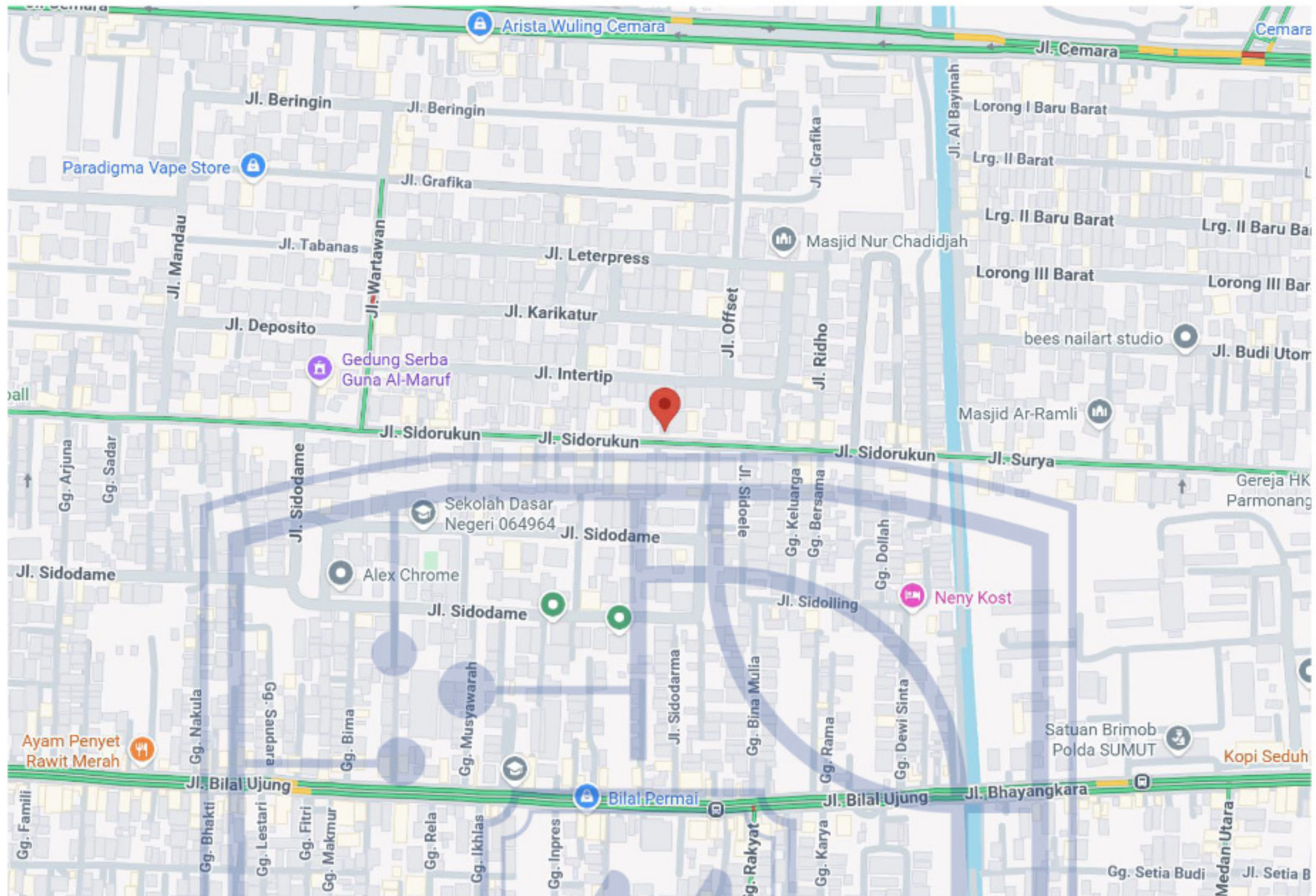
PT Star Samudera Logistik merupakan perusahaan yang berdiri sejak tahun 2020 dan berfokus pada penyediaan layanan jasa logistik darat, laut, dan udara secara komprehensif. Perusahaan ini menawarkan berbagai layanan utama yang mencakup Ekspedisi Muatan Kapal Laut (EMKL), pengurusan pabean (custom clearance), serta layanan pendukung ekspor dan impor untuk kelancaran operasional bisnis pelanggan.



Gambar 2.2 Logo PT Star Samudera Logistik (Sumber: Dokumen Perusahaan)

Dengan pengalaman yang dimiliki, PT Star Samudera Logistik telah memiliki jaringan mitra bisnis yang luas yang tersebar di hampir seluruh wilayah Indonesia serta mencakup jaringan internasional seperti Tiongkok, Korea, India, Dubai, Singapura, dan Vietnam. Layanan yang ditawarkan dirancang untuk memenuhi standar logistik modern, yang meliputi pengiriman domestik (trucking domestic) antar kota dan pulau, Pejabat Pembuat Jasa Kepabeanan (PPJK), pengiriman kontainer Less than Container Load (LCL) dan Full Container Load (FCL), hingga layanan pengiriman pintu ke pintu (door to door).

Saat ini, kantor operasional PT Star Samudera Logistik berlokasi di Jl. Kapten Sumarsono No. 32, Kelurahan Sunggal, Kecamatan Medan Sunggal, Kabupaten Deli Serdang, Sumatera Utara. Lokasi yang strategis ini mempermudah perusahaan dalam mengelola alur distribusi dan logistik pelanggan. Adapun titik lokasi kantor PT Star Samudera Logistik dapat dilihat pada pemetaan geografis berikut:



Gambar 2.3 Peta Lokasi PT Star Samudera Logistik (Sumber: Google Maps)