

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Teori Umum**

Bagi kelancaran penelitian ini, landasan teori umum sangat penting sebagai dasar yang mengarahkan dan memahami konteks penelitian. Pada teori umum ini, akan dijelaskan ringkas mengenai beberapa konsep yang relevan dengan rancang bangun aplikasi penyaluran jasa asisten rumah tangga yaitu rancang bangun, sistem informasi, jasa dan asisten rumah tangga.

##### **2.1.1 Rancang Bangun**

Menurut (Josi et al., 2017) Rancang bangun adalah konsep yang sering digunakan untuk merujuk pada proses pembuatan atau perancangan suatu objek dari tahap awal perancangan hingga tahap akhir produksi. Dalam konteks ini, termasuk dalam prosesnya adalah perencanaan, pengembangan konsep, perincian desain, pelaksanaan, dan penyelesaian pembuatan objek tersebut. Proses rancang bangun ini membentuk dasar bagi penciptaan dan pengembangan berbagai jenis produk atau sistem yang dapat beragam dalam kompleksitasnya.

Perancangan adalah tahap kunci dalam mengembangkan suatu sistem, yang melibatkan proses menyusun rencana rinci berdasarkan temuan dan rekomendasi yang ditemukan selama analisis sistem. Pada tahap ini, langkah-langkah awal ditempuh untuk merinci bagaimana sistem akan dibangun, termasuk spesifikasi yang akan menjadi panduan dalam proses pengembangannya (Eko Nugroho, 2016).

### **2.1.2 Sistem Informasi**

Sistem informasi sebagai alat bantu yang digunakan untuk mendigitalisasi seluruh proses bisnis yang ada pada industri, merupakan salah satu langkah memasuki era digital ini. Dengan menggunakan sistem informasi proses pertukaran informasi antar proses bisnis dapat dengan mudah dilakukan, tidak ada lagi batasan waktu dan tempat yang menjadi hambatan. Proses pengelolaan data menjadi suatu informasi menjadi lebih mudah dan efisien, yang membuat kemudahan dalam menangkap informasi (Dharma et al., 2019).

Menurut (Silalahi & Saragih, n.d.) Sistem informasi merupakan gabungan dari elemen yang terhubung satu sama lain yang menciptakan suatu keutuhan dan saling terintegrasi yang berfungsi memproses data serta menyalurkan informasi. Sistem informasi ialah gabungan yang terdiri perangkat keras, perangkat lunak, dan pengguna yang akan mengerjakan sejumlah data-data menjadi suatu informasi. Selain itu menurut (Sudana et al., n.d.) Sistem informasi adalah gabungan dari teknologi, prosedur kerja, data, serta interaksi individu yang terstruktur untuk mencapai tujuan tertentu dalam suatu organisasi atau lingkungan bisnis.

### **2.1.3 Website**

*Website* adalah sekumpulan halaman yang dimulai dengan halaman utama yang berisi informasi, iklan, dan aplikasi. Sebaliknya, *Web Server* adalah suatu komputer yang terdiri dari perangkat keras dan perangkat lunak. *Web browser* adalah perangkat lunak yang digunakan untuk mengambil dan menampilkan informasi sumber (Nurhadi, 2018). Selain itu menurut (Mukhlis, 2023) *website* merujuk pada proses penciptaan halaman web tunggal, yang mencakup elemen

estetika seperti grafik, palet warna, jenis huruf, isi konten, dan struktur navigasi. Tujuan dari desain *website* ini adalah untuk menciptakan halaman web yang tidak hanya memiliki tampilan menarik, tetapi juga mengandung informasi yang unik, sehingga mempermudah pengunjung dalam mengakses dan memperoleh informasi yang mereka butuhkan.

*Website* adalah kumpulan halaman web yang terkait satu sama lain dan berada di *server* yang sama, yang berisi informasi yang disediakan oleh individu, kelompok, atau organisasi. *Website* dapat diakses melalui jaringan seperti Internet atau jaringan lokal (LAN) dengan menggunakan alamat Internet yang disebut URL. *Website* adalah bagian dari *World Wide Web*, yang merupakan sistem informasi global yang dapat diakses oleh publik melalui Internet (Durbin Hutagalung & Arif, 2018).

#### **2.1.4 Web**

Menurut (Melanda et al., 2023) Web merupakan suatu sistem informasi yang menyokong interaksi pengguna melalui antarmuka berbasis web. Ciri-ciri khas dari web termasuk adanya persistensi data, dukungan untuk transaksi, dan komposisi halaman web dinamis yang dapat dianggap sebagai suatu bentuk hibrida antara hypermedia dan sistem informasi.

Web adalah sebuah perangkat lunak yang digunakan untuk menampilkan dokumen-dokumen yang ada di web, yang merupakan kumpulan informasi yang tersedia di internet. Web memungkinkan pengguna untuk mengakses internet dengan menggunakan perangkat lunak yang terhubung dengan jaringan internet.

Web menampilkan dokumen-dokumen dalam berbagai format, seperti teks, gambar, *audio*, *video*, atau kombinasi dari semuanya. Web juga memungkinkan pengguna untuk berinteraksi dengan dokumen-dokumen tersebut, misalnya dengan mengklik tautan, mengisi formulir, atau memberikan komentar. Web adalah salah satu cara untuk mendapatkan informasi dan berkomunikasi secara *online* (Gede Surya Cipta Nugraha et al., 2021).

### **2.1.5 Jasa**

Jasa merujuk kepada segala tindakan atau pelaksanaan yang disediakan oleh satu pihak kepada pihak lain dan umumnya bersifat tak berwujud serta tidak mengakibatkan peralihan kepemilikan. Produk atau layanan jasa ini bisa terkait dengan atau tidak tergantung pada sebuah barang fisik (Refandaru et al., n.d.).

Menurut (Janiver et al., n.d.). Jasa adalah suatu bentuk aktivitas atau pekerjaan yang dilakukan oleh satu pihak untuk memberikan manfaat atau kepuasan kepada pihak lain. Ini tidak melibatkan transfer kepemilikan benda, namun lebih kepada penyediaan kemampuan atau ketrampilan untuk memenuhi kebutuhan atau keinginan pelanggan. Jasa bersifat tidak berwujud dan sering kali melibatkan interaksi langsung antara penyedia jasa dan penerima jasa. Contoh jasa mencakup layanan profesional, perbaikan, konsultasi, dan berbagai bentuk aktivitas lain yang memberikan nilai tambah kepada individu atau organisasi

### **2.1.6 Asisten Rumah Tangga**

Asisten rumah tangga adalah orang yang bekerja dalam suatu keluarga atau rumah tangga untuk membantu pekerjaan-pekerjaan yang ada di rumah. Pekerjaan

tersebut umumnya meliputi pekerjaan rumah tangga, namun terkadang juga mencakup pekerjaan lain yang berkaitan dengan keluarga. Asisten rumah tangga memiliki posisi kerja yang dipengaruhi oleh apa yang mereka kerjakan dalam suatu keluarga. Keluarga atau rumah tangga secara konseptual adalah kelompok orang yang tinggal bersama dalam satu rumah dan satu dapur. Kelompok orang tersebut biasanya terdiri dari orang tua (ayah/suami dan ibu/istri) dan anak-anak, serta mungkin juga ada satu atau lebih asisten rumah tangga dan kerabat mereka (Ilmi & Qhoiriah, 2016).

Menurut (Rudi Sanjaya et al., 2022) Asisten Rumah Tangga (ART) adalah pekerja domestik yang memegang peran signifikan dalam kehidupan masyarakat, terutama di lingkungan perkotaan. Peran ART menjadi semakin penting karena banyak individu, terutama wanita yang aktif berkarir, memiliki jadwal yang padat dan tidak dapat menangani seluruh pekerjaan rumah tangga sendiri. Untuk mengatasi keterbatasan waktu dan tenaga, masyarakat mencari bantuan dari ART untuk membantu menyelesaikan tugas-tugas rumah tangga.

## **2.2 Teori Khusus**

### **2.2.1 Sistem Informasi Jasa**

Sistem Informasi Jasa adalah suatu aplikasi atau sistem yang dirancang untuk mengelola dan mengoptimalkan proses-proses bisnis dalam penyediaan layanan jasa. Sistem ini mencakup pengumpulan, penyimpanan, pengolahan, dan penyebarluasan informasi yang berkaitan dengan berbagai jenis layanan jasa, seperti reservasi tiket, pemesanan hotel, layanan kesehatan, pendidikan, dan lain sebagainya. Tujuan utama dari Sistem Informasi Jasa adalah untuk meningkatkan

efisiensi, meningkatkan kualitas layanan, dan memenuhi kebutuhan pelanggan dalam pemanfaatan jasa-jasa yang ditawarkan. Sistem Informasi Jasa dapat berbasis web atau aplikasi, memungkinkan pelanggan untuk mengakses informasi dan melakukan transaksi dengan mudah melalui *platform* digital (Dhika & Fitriansyah, 2016).

Sistem Informasi Jasa memanfaatkan *platform* digital untuk memudahkan penyampaian informasi layanan kepada pengguna dan memfasilitasi proses pemesanan atau penggunaan layanan secara *online*. Metode pengumpulan data yang digunakan dapat melibatkan observasi dan studi pustaka, sementara metode perancangan sistem dapat mengikuti pendekatan *waterfall* atau metode pengembangan perangkat lunak lainnya. Sistem Informasi Jasa memungkinkan penyedia layanan untuk meningkatkan efisiensi operasional, memperluas jangkauan layanan kepada pelanggan, dan memberikan pengalaman pengguna yang lebih baik melalui pemanfaatan teknologi informasi. Dengan adanya sistem ini, pelanggan dapat mengakses informasi layanan, melakukan pemesanan, atau menggunakan layanan jasa secara praktis dan efisien. Tujuan utamanya adalah memenuhi kebutuhan pelanggan dengan cara yang cepat, tepat, dan praktis melalui penggunaan teknologi modern (Bianglala Informatika & Masturah, n.d.).

### **2.2.2 Sistem Informasi Penyaluran Jasa Asisten Rumah Tangga**

Sistem Informasi Penyaluran Jasa Asisten Rumah Tangga adalah sebuah aplikasi atau sistem yang dirancang untuk membantu mengelola dan memfasilitasi proses penyaluran asisten rumah tangga kepada pelanggan. Sistem ini mencakup pengumpulan, penyimpanan, pengolahan, dan penyebarluasan informasi terkait

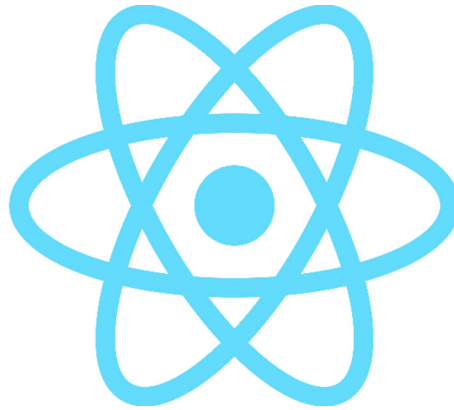
asisten rumah tangga yang tersedia, permintaan pelanggan, penempatan, jadwal, pembayaran, dan informasi lainnya yang berkaitan dengan layanan asisten rumah tangga. Tujuan dari sistem ini adalah memastikan penyaluran asisten rumah tangga berjalan efisien, akurat, dan efektif untuk memenuhi kebutuhan rumah tangga yang memerlukan bantuan tenaga kerja domestik (Refandaru et al., n.d.).

Menurut (Ali Muchtar et al., n.d.) Sistem Informasi Penyalur Jasa Asisten Rumah Tangga adalah sistem berbasis komputer yang bertujuan untuk mengelola dan memfasilitasi penyaluran tenaga kerja asisten rumah tangga serta pengasuh anak kepada pelanggan. Sistem ini mengumpulkan, menyimpan, dan mengelola informasi terkait calon asisten rumah tangga, termasuk biodata, kualifikasi, dan ketersediaan. Dengan menggunakan teknologi komputer, sistem ini memungkinkan pelanggan untuk mencari dan memilih calon pekerja rumah tangga sesuai dengan kebutuhan mereka. Penggunaan sistem informasi ini membantu menyederhanakan proses pencarian, pemilihan, dan penempatan tenaga kerja, menggantikan metode manual yang lebih kompleks dan memakan waktu. Sistem Informasi Penyalur Jasa Asisten Rumah Tangga bertujuan meningkatkan efisiensi, akurasi, dan kecepatan dalam proses penyaluran pekerja rumah tangga dan pengasuh anak.

### **2.2.3 React.js**

Menurut (Wali & Ahmad, 2018) React.js merupakan salah satu kerangka kerja web yang paling populer di dunia Node.js. Dokumentasinya sangat komprehensif, dan kemudahan penggunaannya memungkinkan kita untuk mengembangkan berbagai produk, termasuk aplikasi web dan antarmuka

pemrograman aplikasi (API) *RESTful*. React.js dapat dijadikan dasar untuk membangun kerangka kerja web yang lebih kompleks, karena keunggulan tersebut.



**Gambar 2.1** Logo ReactJS

Sumber: (<https://react.dev/>)

Dengan ReactJS, pengembang memiliki kemampuan untuk membuat komponen antarmuka pengguna (UI) yang lebih interaktif, memiliki keadaan (*stateful*), dan dapat digunakan kembali (*reusable*). Dalam konteks *model-view-controller* (MVC), ReactJS memiliki tanggung jawab terutama pada bagian tampilan (*views*) (Panjaitan & Pakpahan, 2021).

#### **2.2.4 Node.js**

Menurut (Firdaus et al., 2019) Node.js adalah sebuah perangkat lunak yang telah dirancang untuk memfasilitasi pengembangan aplikasi berbasis web dan ditulis dalam bahasa pemrograman JavaScript. Sebelumnya, JavaScript dikenal sebagai bahasa pemrograman yang berfungsi hanya di sisi klien atau peramban web. Akan tetapi, Node.js hadir untuk memperluas peran JavaScript sehingga dapat digunakan sebagai bahasa pemrograman yang berjalan di sisi server, sebagaimana yang biasanya dilakukan oleh bahasa pemrograman seperti PHP, Ruby, Perl, dan



lainnya. Node.js dapat dijalankan di berbagai sistem operasi, termasuk Windows, Mac OS X, dan Linux, tanpa perlu melakukan perubahan pada kode program.



**Gambar 2.2** Node.js

Sumber: (<https://nodejs.org/>)

Node.js adalah suatu *platform runtime* berbasis JavaScript dengan tingkat skalabilitas yang tinggi, mampu menjalankan kode JavaScript di luar konteks peramban (*browser*). Keunggulan Node.js terletak pada pendekatan *non-blocking* yang memfasilitasi sistem untuk melaksanakan operasi secara bersamaan tanpa harus menunggu penyelesaian operasi sebelumnya. Karakteristik ini memungkinkan Node.js untuk mengelola sejumlah besar permintaan secara simultan, dan alat (*tools*) yang dapat mempercepat pengembangan aplikasi. Dengan demikian, Node.js telah menjadi salah satu pilihan utama bagi pengembang web untuk menciptakan aplikasi yang tangguh dan responsif di dunia digital yang terus berkembang (Nurhayati & Agussalim, 2023).

### **2.2.5 Express.js**

Express.js merupakan suatu kerangka kerja web yang tangguh dan efisien, beroperasi di atas *platform* Node.js. Kerangka kerja ini memfasilitasi pengembang untuk membangun aplikasi web dan API dengan cepat, kuat, dan dapat diandalkan.

Di sisi lain, Node.js adalah *platform* JavaScript berbasis *event-driven* yang memungkinkan penanganan banyak koneksi secara efisien dan berskala tinggi (Gelar Guntara & Azkarin, 2023)



**Gambar 2.3** Express.js

Sumber: (<https://expressjs.com/>)

Menurut (Hilmi Hamdani et al., 2023) Express.js merupakan sebuah kerangka kerja (*framework*) yang dirancang untuk beroperasi pada *platform* aplikasi Node.js yang bersifat minimalis dan sangat fleksibel. Selain itu, Express.js juga dikenal memiliki dokumentasi yang komprehensif dan penggunaannya yang relatif mudah, yang memungkinkan pengembang untuk mengembangkan berbagai jenis produk, baik itu aplikasi web maupun layanan API *RESTful*.

### **2.2.6 Database & MySQL**

MySQL adalah sebuah pengembangan perangkat *database* yang tersedia secara gratis. MySQL sering digunakan sebagai sistem *database* karena kemudahan penggunaannya dan ketersediaannya yang luas. MySQL menggunakan bahasa SQL yang saat ini telah menjadi standar. Di dalam ekosistem Linux atau Unix, MySQL merupakan salah satu perangkat lunak *database* yang paling terkenal. Keunggulan ini didukung oleh performa *query* yang dianggap paling cepat dan minim masalah pada saat itu (Wiguna et al., 2019).

Menurut (Tukino et al., 2022) MySQL digolongkan sebagai Sistem Manajemen Basis Data (DBMS). Aplikasi ini membantu pengguna dalam mengelola data dengan tingkat fleksibilitas dan kecepatan yang tinggi. Beberapa kegiatan komputasi yang didukung oleh perangkat lunak ini mencakup penyimpanan data di dalam tabel, penghapusan data dari tabel, modifikasi data dalam tabel, pengambilan data yang tersimpan dalam tabel, serta pemilihan data tertentu dan penetapan izin akses data.

### 2.2.7 Draw.io

Draw.io adalah aplikasi yang digunakan untuk membuat desain diagram tanpa perlu melakukan instalasi aplikasi, hanya dengan koneksi internet. Aplikasi ini menonjolkan fleksibilitasnya karena memungkinkan penyimpanan data di berbagai tempat dan memanfaatkan media penyimpanan pribadi. Dengan berbagai paket yang ditawarkan, aplikasi ini memfasilitasi perancangan dan pembuatan diagram, termasuk UML, *flowchart*, dan *12 Entity Relation*. Kelebihan lainnya adalah Draw.io tidak membatasi jumlah pengguna dan diagram yang dapat dibuat (Safira & Purtiningrum, n.d.).



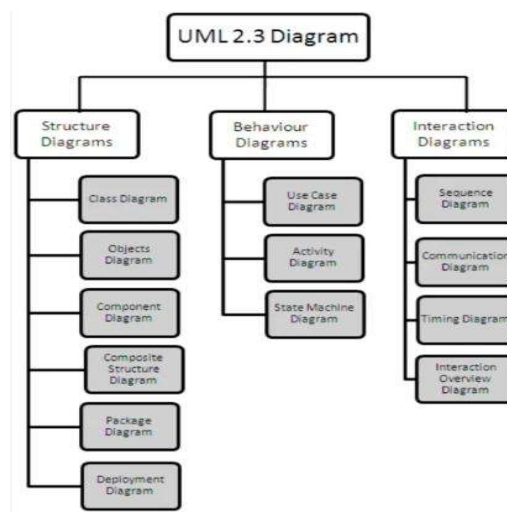
**Gambar 2.4** Draw.io

Sumber: (<https://draw.io>)

Menurut (Suharyanto et al., n.d.) Draw.io adalah sebuah *platform* yang dirancang secara khusus untuk membuat diagram secara daring. Untuk mengaksesnya, Anda hanya memerlukan peramban web yang mendukung HTML5 dan koneksi internet. Draw.io telah terintegrasi dengan Google Drive untuk penyimpanan berkas dan juga menyediakan opsi ekspor dalam format JPG/PNG/SVG/XML.

### 2.2.8 UML

Menurut (Suharyanto et al., n.d.) *Unified Modeling Language* (UML) merujuk pada teknik pemodelan yang digunakan untuk mengembangkan dan merencanakan sistem atau perangkat lunak agar dapat terstruktur dengan baik selama proses perancangan. Pemodelan ini bertujuan menyederhanakan pemahaman terhadap masalah yang kompleks. UML adalah suatu bahasa yang digunakan untuk menspesifikasi, memvisualisasikan, dan membangun dasar sistem perangkat lunak, termasuk melibatkan pemodelan aturan bisnis.



**Gambar 2.5** UML 2.3

Sumber: (<https://123dok.com/>)

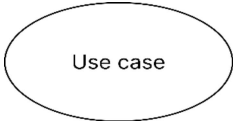
UML (*Unified Modeling Language*) merupakan salah satu bahasa standar yang sering dipakai dalam industri global untuk mengidentifikasi kebutuhan, melakukan analisis dan perancangan, serta menggambarkan struktur dalam konteks pemrograman berbasis objek (Wira et al., 2019). UML dibuat oleh *Object Management Group* (OMG) dan rancangan spesifikasi UML 1.0 diajukan ke OMG pada Januari 1997.



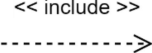
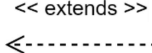
### 2.2.8.1 Use case diagram

*Use Case* merupakan tahap awal dalam proses pemodelan suatu sistem. *Use Case* digunakan untuk memodelkan kebutuhan fungsional sistem, di mana setiap *Use Case* diilustrasikan sebagai kunci dari suatu skenario yang dilakukan oleh aktor dan dibatasi dalam suatu batas sistem. Setiap *Use Case* dihubungkan dengan sebuah garis notasi (Aliman, 2021).

Komponen-komponen utama dalam *Use Case* Diagram meliputi:

**Tabel 2.1** Daftar Simbol Pada *Use Case*

Simbol	Nama	Deskripsi
	<i>Use Case</i>	Spesifikasi dari sekumpulan aksi yang dilakukan oleh sistem, yang menghasilkan hasil yang dapat diamati oleh aktor atau pemangku kepentingan lainnya. <i>Use case</i> direpresentasikan dengan gambar oval dengan label.

Simbol	Nama	Deskripsi
	<i>Actor</i>	<p>Entitas yang berinteraksi dengan sistem. Aktor direpresentasikan dengan gambar orang yang membutuhkan atau memberikan informasi kepada sistem.</p>
	Asosiasi	<p>Menghubungkan aktor dan <i>use case</i> untuk menunjukkan bahwa aktor berpartisipasi dalam <i>use case</i> tersebut. Asosiasi sesuai dengan urutan aksi antara aktor dan <i>use case</i> dalam mencapai tujuan <i>use case</i>.</p>
	<i>Include</i>	<p>Hubungan yang menunjukkan bahwa sebuah <i>use case</i> mencakup perilaku dari <i>use case</i> lainnya. <i>Include</i> digunakan untuk memodelkan fungsi umum yang digunakan oleh beberapa <i>use case</i>. <i>Include</i> direpresentasikan dengan garis putus-putus dengan label «include».</p>
	<i>Extends</i>	<p>Hubungan yang menunjukkan bahwa sebuah <i>use case</i> dapat memperluas perilaku dari <i>use case</i> lainnya pada kondisi tertentu. <i>Extend</i> digunakan untuk memodelkan variasi atau pengecualian</p>


Simbol	Nama	Deskripsi
		dari <i>use case</i> utama. <i>Extend</i> direpresentasikan dengan garis putus-putus dengan label « <i>extend</i> ».

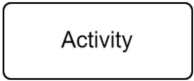
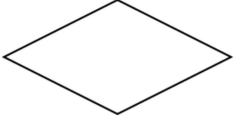

Tujuan utama dari *Use Case* Diagram adalah untuk memberikan pandangan visual yang jelas tentang interaksi antara pengguna (aktor) dan fungsionalitas sistem (*use case*). Diagram ini membantu tim pengembang dan pemangku kepentingan untuk memahami kebutuhan sistem, mengidentifikasi kasus penggunaan utama, dan merencanakan fungsionalitas sistem.

#### 2.2.8.2 Activity Diagram

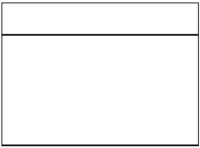
*Activity* Diagram adalah jenis diagram yang mengilustrasikan bagaimana suatu objek atau sistem beroperasi. Sebuah diagram aktivitas digambarkan dengan alur proses yang terstruktur dari suatu *use case*, mulai dari titik awal hingga titik akhir. Setiap aktivitas diilustrasikan dengan notasi-notasi yang sesuai dengan fungsinya (Wali & Ahmad, 2018). Terdapat beberapa komponen pada *Activity* diagram yaitu:

**Tabel 2.2** Daftar Simbol Pada *Activity* Diagram

Simbol	Nama	Deskripsi
	<i>Initialization/Status</i> Awal	titik mulai dari aliran aktivitas. Status awal menandakan bahwa belum ada aktivitas yang berjalan

Simbol	Nama	Deskripsi
		atau belum ada masukan yang diterima. Status awal direpresentasikan dengan lingkaran hitam kecil
	Aktivitas	Digunakan untuk menggambarkan tindakan yang dilakukan oleh sistem dan perlu dimulai dengan kata kerja.
	<i>Decision Node</i>	Percabangan dalam aliran aktivitas. <i>Decision Node</i> digunakan untuk dapat merepresentasikan pilihan atau keputusan yang harus dibuat berdasarkan kondisi atau kriteria tertentu.
	<i>Final State/ Status Akhir</i>	Titik selesai dari aliran aktivitas. Status akhir menandakan bahwa semua aktivitas telah selesai atau sudah ada keluaran yang dihasilkan. Status akhir direpresentasikan dengan lingkaran hitam besar dengan



Simbol	Nama	Deskripsi
		lingkaran hitam kecil di dalamnya.
	<i>Swimlane</i>	Digunakan untuk menguraikan bagaimana organisasi bisnis membagi tugas dan tanggung jawab dalam suatu kegiatan yang tengah berlangsung.

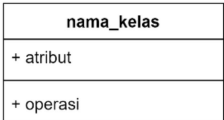
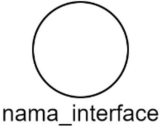

Tujuan utama dari *Activity Diagram* adalah untuk memberikan gambaran yang jelas tentang bagaimana alur kerja atau proses berlangsung dalam sistem. Diagram ini membantu dalam memahami interaksi antara aktivitas-aktivitas, pengambilan keputusan, dan jalur aliran yang berbeda.



### 2.2.8.3 *Class Diagram*

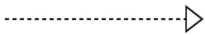
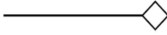
*Class diagram* adalah salah satu model penting dalam UML yang berfungsi untuk membuat model logika dari suatu sistem. Sebuah diagram kelas memperlihatkan skema arsitektur dari sistem yang sedang dirancang. Diagram kelas digambarkan dengan menggunakan kelas yang berisi atribut dan metode, di mana setiap kelas dihubungkan oleh suatu garis yang disebut asosiasi (Wali & Ahmad, 2018).

Komponen-komponen dari *Class diagram* adalah:

Tabel 2.3 Daftar Simbol Pada *Class Diagram*

Simbol	Nama	Deskripsi
	<i>Class</i>	Kelas merepresentasikan sebuah konsep, objek, atau entitas dalam sistem dan memiliki atribut dan operasi yang mendefinisikan karakteristik dan perilakunya.
	<i>Interface</i>	Kumpulan operasi yang dideklarasikan tanpa implementasinya. <i>Interface</i> digunakan untuk menentukan perilaku yang harus dimiliki oleh kelas yang mengimplementasikannya
	<i>Association</i>	Hubungan struktural antara dua atau lebih kelas yang menunjukkan bahwa kelas-kelas tersebut saling terkait atau berkolaborasi. <i>Association</i> direpresentasikan dengan garis

Simbol	Nama	Deskripsi
		lurus antara kelas-kelas yang terhubung.
	<i>Directed Association</i>	<p>Menunjukkan arah aliran informasi atau kontrol antara kelas-kelas yang terhubung.</p> <p><i>Directed Association</i> direpresentasikan dengan garis lurus dengan panah pada salah satu ujungnya.</p>
	<i>Generalization</i>	<p>Hubungan spesialisasi antara dua kelas yang menunjukkan bahwa kelas anak (<i>subclass</i>) adalah kasus khusus dari kelas induk (<i>superclass</i>).</p> <p><i>Generalization</i> direpresentasikan dengan garis lurus dengan panah kosong pada ujungnya. Panah menunjukkan arah dari <i>subclass</i> ke <i>superclass</i>.</p>

Simbol	Nama	Deskripsi
	<i>Dependecy</i>	<p>Hubungan fungsional antara dua kelas yang menunjukkan bahwa perubahan pada satu kelas akan mempengaruhi kelas lainnya. <i>Dependency</i> direpresentasikan dengan garis putus-putus dengan panah pada salah satu ujungnya. Panah menunjukkan kelas yang bergantung pada kelas lainnya.</p>
	<i>Aggregation</i>	<p>Menunjukkan hubungan “bagian-dari” antara dua kelas. <i>Aggregation</i> menunjukkan bahwa satu kelas adalah koleksi atau wadah dari kelas lainnya, tetapi tidak memiliki kendali penuh atas siklus hidupnya. <i>Aggregation</i> direpresentasikan dengan garis lurus dengan berlian kosong pada salah satu ujungnya. Berlian menunjukkan kelas agregat yang memiliki</p>

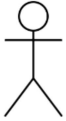
Simbol	Nama	Deskripsi
		kelas lainnya sebagai bagian dari komposisinya.

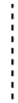


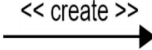
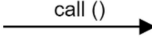
#### 2.2.8.4 *Sequence Diagram*

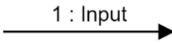
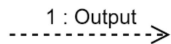
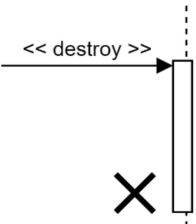
*Sequence* diagram adalah suatu diagram yang menggambarkan interaksi dinamis antara sejumlah objek dalam suatu sistem. Fungsinya adalah untuk menunjukkan urutan pesan yang dikirim antara objek, serta interaksi yang terjadi di antara objek-objek tersebut (Musthofa & Adiguna, 2022).

Komponen-komponen pada *Sequence Diagram*:

**Tabel 2.4** Daftar Simbol Pada *Sequence Diagram*

Simbol	Nama	Deskripsi
	Aktor	Entitas yang berinteraksi dengan sistem. Aktor bisa berupa individu, organisasi, atau bahkan sistem lain yang berhubungan dengan sistem yang sedang digambarkan. Biasanya, dalam diagram ini, kita menggunakan ikon gambar orang untuk merepresentasikan aktor

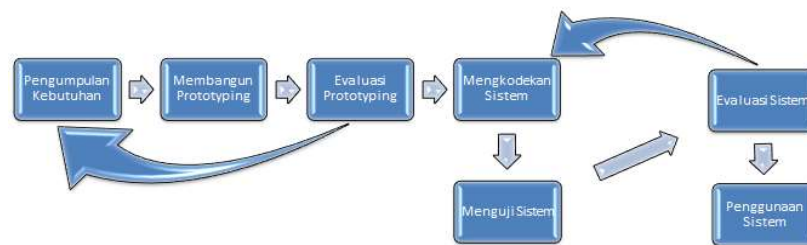
Simbol	Nama	Deskripsi
	<i>Lifeline</i>	Garis vertikal yang menunjukkan sepanjang waktu objek tersebut aktif atau "hidup" dalam interaksi. Garis ini menunjukkan kapan objek dibuat dan kapan objek tersebut dihancurkan dalam konteks interaksi
	<i>Object</i>	Entitas yang berperan dalam interaksi. Objek ini bisa berupa kelas, aktor, atau sistem lain yang terlibat dalam alur kerja.
	<i>Activation</i>	Menunjukkan periode waktu di mana objek tersebut aktif atau sedang melakukan atau menerima pesan. Ini membantu dalam menunjukkan kapan objek berpartisipasi dalam interaksi.
	<i>Create</i>	Pesan khusus yang menunjukkan bahwa sebuah objek diciptakan oleh objek lain. Pesan ini direpresentasikan dengan garis panah dengan label "create."
	<i>Call</i>	Pesan yang menunjukkan bahwa sebuah objek memanggil operasi atau metode dari objek lain. Pesan ini direpresentasikan

Simbol	Nama	Deskripsi
		dengan garis panah dengan nama operasi atau metode di atasnya
	<i>Send</i>	Pesan yang digunakan untuk menunjukkan bahwa sebuah objek mengirimkan informasi atau sinyal ke objek lain. Pesan ini direpresentasikan dengan garis panah dengan nama informasi atau sinyal di atasnya
	<i>Return</i>	Pesan yang menunjukkan bahwa sebuah objek mengembalikan nilai atau hasil ke objek lain. Pesan ini direpresentasikan dengan garis panah putus-putus dengan nama nilai atau hasil di atasnya.
	<i>Destroy</i>	menunjukkan bahwa sebuah objek dihancurkan oleh objek lain. Pesan ini direpresentasikan dengan garis panah dengan label " <i>destroy</i> " dan simbol X pada ujungnya.

### 2.2.9 Metode *Prototype*

Terdapat beberapa jenis metode SDLC dan salah satunya adalah Metode *prototype (prototyping model)*. Menurut (Siswidiyanto et al., 2020) Metode

*Prototype* merupakan salah satu pendekatan dalam pengembangan perangkat lunak yang secara langsung menunjukkan bagaimana suatu perangkat lunak atau komponennya akan beroperasi dalam lingkungannya sebelum tahap konstruksi aktual dilaksanakan. Model prototipe berperan sebagai penanda gambaran yang akan diwujudkan di masa mendatang dan membedakan dua fungsi utama, yaitu eksplorasi dan demonstrasi.



**Gambar 2.6** Metode *Prototype*

Sumber: (<https://medium.com/dot-intern/sdlc-metode-prototipe-8f50322b14bf>)

Menurut (Saptia Kurnia & Risyda, n.d.) terdapat beberapa tahapan-tahapan dalam metode *prototype*:

a. Pengumpulan Kebutuhan

Langkah awal dalam metode prototipe melibatkan pengidentifikasian semua perangkat dan isu yang relevan. Analisis dan identifikasi kebutuhan keseluruhan sistem menjadi tahapan yang sangat krusial. Dari sini, dapat diketahui permasalahan yang perlu dipecahkan.

b. Membangun *Prototype*



Langkah berikutnya dalam metode prototipe adalah membangun model awal yang fokus pada penyajian, seperti menciptakan *input* dan *output* sistem. Fase ini hanya mencakup pembuatan prototipe, sementara langkah selanjutnya akan melibatkan tindak lanjut yang perlu diimplementasikan.

c. Evaluasi *Prototype*

Sebelum melanjutkan ke langkah berikutnya, evaluasi prototipe menjadi tahap yang wajib. Hal ini mencakup pemeriksaan langkah 1 dan 2 karena ini menjadi penentu keberhasilan dan proses yang sangat penting. Kesalahan atau kekurangan pada tahapan awal akan menyulitkan langkah-langkah berikutnya.

d. Mengkodean Sistem

Sebelum melakukan pengkodean, penting untuk memahami bahasa pemrograman yang akan digunakan. Dalam tahap ini, perancangan, pembangunan, dan implementasi web atau aplikasi disesuaikan dengan kebutuhan dalam bentuk kode program.

e. Pengujian Sistem

Setelah pengkodean, dilakukan pengujian program. Terdapat berbagai metode pengujian, seperti menggunakan metode *white box* atau *black box*. Metode *white box* menguji kode program, sementara metode *black box* menguji fungsi tampilan apakah sesuai dengan aplikasi atau tidak.

f. Evaluasi Sistem

Evaluasi dilakukan terhadap semua langkah yang telah dilalui. Diperiksa apakah sistem sudah sesuai dengan kebutuhan atau masih memerlukan revisi. Jika masih diperlukan revisi, maka dapat kembali ke tahap 1 dan 2.

g. Menggunakan Sistem

Setelah sistem selesai diimplementasikan, perlu dilakukan upaya pemeliharaan agar sistem tetap terjaga dan berfungsi optimal. Hal ini bertujuan untuk meningkatkan produktivitas dan kinerja sistem.