

## BAB II

### LANDASAN TEORI

#### 2.1. Kajian Pustaka

Pada bagian ini, akan dibahas dua hal penting, yaitu kajian penelitian terdahulu dan perbandingan dengan penelitian terdahulu. Adapun uraian selengkapnya adalah sebagai berikut:

##### 2.1.1. Kajian Penelitian Terdahulu

Kajian Penelitian Terdahulu berfungsi untuk mengidentifikasi dan membahas penelitian-penelitian sebelumnya yang memiliki metode serupa dalam pengembangan sistem. Tujuan dari kajian ini adalah untuk memahami pendekatan yang telah digunakan oleh peneliti lain, sehingga dapat dijadikan acuan untuk menyempurnakan perancangan dan implementasi sistem dalam penelitian yang dilakukan.

**Tabel 2. 1** *Penelitian Terdahulu*

Peneliti	Judul Penelitian	Ringkasan	Metode	Hasil Penelitian
Sri Martianingsih Jibrán, Nur Jannah, Dwi Irang Putri Rahmani.  Universitas Muhammadiyah Buton  Journal of Human And Education Volume 5, No. 1, Tahun 2025, pp 576-588 E-ISSN 2776-5857, P- ISSN 2776-7876  <b>DOI:</b>	Pengembangan Sistem Informasi Manajemen Penjualan Berbasis <i>Website</i> untuk Meningkatkan Efisiensi Operasional pada Toko Win Glowing dengan Metode <i>Waterfall</i>	Penelitian ini merancang sistem manajemen penjualan berbasis web untuk Toko Win Glowing guna meningkatkan efisiensi operasional, meminimalkan kesalahan pencatatan, dan mempercepat transaksi.	Metode <i>Waterfall</i>	Sistem yang dikembangkan mampu mengotomatisas i pencatatan penjualan, pengelolaan stok, dan pelaporan, sehingga mempercepat pelayanan dan membantu pemilik toko dalam analisis data real-time.

Peneliti	Judul Penelitian	Ringkasan	Metode	Hasil Penelitian
<a href="https://doi.org/10.31004/jh.v5i1.2225">https://doi.org/10.31004/jh.v5i1.2225</a>				
<p>Rais Akbar Wibowo, Didik indrayana, Asriyanik.</p> <p>Universitas Muhammadiyah Sukabumi, indonesia</p> <p>Syntax Admiration: p-ISSN 2722-7782   e-ISSN 2722-5356 Vol. 6, No. 1, Januari 2025</p> <p><b>DOI:</b> <a href="https://doi.org/10.46799/jsa.v6i1.2058">https://doi.org/10.46799/jsa.v6i1.2058</a></p>	<p>Implementasi Metode <i>Waterfall</i> dalam Pengembangan Aplikasi Toko Online Bebas Web di Cv Aishastore.Id</p>	<p>Penelitian ini mengembangkan aplikasi toko daring berbasis web untuk CV AishaStore.id menggunakan metode <i>Waterfall</i> dengan teknologi PHP, JavaScript, CSS, dan MySQL. Fitur utama meliputi manajemen stok, katalog produk, transaksi, dan akun pengguna.</p>	<p>Metode <i>Waterfall</i></p>	<p>Aplikasi ini meningkatkan efisiensi operasional, mengotomatiskan stok, dan mempermudah transaksi. Pengujian <i>Black Box</i> memastikan semua fitur berfungsi. Sistem ini juga memperluas pasar dan mendukung pengambilan keputusan berbasis data, dengan potensi pengembangan ke <i>mobile</i>, analitik, dan keamanan data.</p>
<p>Erika Hulu, Raudhah , Sahara Abdy.</p> <p>STMIK Logika, Medan, Indonesia</p> <p>Volume 2, Nomor 2, Juli 2025, Page 13-19 ISSN 3032-6834</p> <p><a href="https://ejurnal.stmikmethodistbinjai.ac.id/jip/article/view/75">https://ejurnal.stmikmethodistbinjai.ac.id/jip/article/view/75</a></p>	<p>Perancangan Sistem Informasi Pendaftaran Pasien Baru di RSIA Pramaliesa Batang Kuis Menggunakan Metode <i>Waterfall</i></p>	<p>Penelitian ini mengembangkan sistem pendaftaran pasien baru berbasis web di RSIA Pramaliesa menggunakan PHP dan <i>MySQL</i> dengan metode <i>Waterfall</i>. Sistem ini menggantikan proses manual yang sering menyebabkan antrean panjang,</p>	<p>Metode <i>Waterfall</i></p>	<p>Sistem yang dikembangkan mempermudah pengolahan data pasien, meminimalkan kehilangan data, serta mempercepat proses pendaftaran. Hasilnya, waktu tunggu pasien berkurang dan kepuasan layanan meningkat.</p>

Peneliti	Judul Penelitian	Ringkasan	Metode	Hasil Penelitian
		kehilangan data, dan kesalahan input.		
<p>Dewa Kurniawan Nusantoro1, Pastima Simanjuntak.</p> <p>Program Studi Teknik Informatika, Universitas Putera Batam.</p> <p>JURNAL COMASIE - VOL. 12NO. 02 (2025) ISSN (Online) 2715-6265</p> <p><b>DOI:</b> <a href="https://doi.org/10.33884/comasiejoumal.v12i2.9705">https://doi.org/10.33884/comasiejoumal.v12i2.9705</a></p>	<p>Implementasi Aplikasi <i>E-Commerce</i> Berbasis Android Menggunakan Metode <i>Waterfall</i></p>	<p>Penelitian ini mengembangkan aplikasi <i>e-commerce</i> untuk <i>Markibong</i>, toko online yang menjual berbagai souvenir khas Batam. Aplikasi ini dibangun menggunakan platform <i>Android open-source</i> untuk meningkatkan efisiensi pemasaran, penjualan, dan transaksi dengan metode <i>Waterfall</i> dalam SDLC.</p>	<p>Metode <i>Waterfall</i></p>	<p>Aplikasi ini memperluas akses pelanggan dan meningkatkan pendapatan toko, sehingga memberikan solusi efektif dalam digitalisasi bisnis souvenir.</p>
<p>Eko Setiawan, Ahmad Subhan Yazid.</p> <p>Sistem Informasi, Universitas Alma Ata, Yogyakarta, Indonesia. Informatika, Universitas Alma Ata, Yogyakarta, Indonesia.</p> <p>Jurnal Ilmiah Komputer e- ISSN: 2685-0877 p-ISSN: 0216-3284</p>	<p>Analisis Perancangan Sistem Informasi Tabungan Sekolah Dasar XYZ Menggunakan Metode <i>Waterfall</i> dan <i>PIECES</i>.</p>	<p>Penelitian ini merancang sistem informasi tabungan sekolah berbasis website untuk SD XYZ guna mengatasi pengelolaan manual yang tidak efisien. Sistem ini mendukung tabungan siswa, pembayaran</p>	<p>Metode <i>Waterfall</i></p>	<p>Hasil pengujian dengan <i>Black Box Testing</i> menunjukkan semua fitur, seperti <i>login</i>, mutasi tabungan, pembayaran, dan laporan, berfungsi dengan baik. Sistem ini meningkatkan efisiensi administrasi, memberikan informasi yang</p>

Peneliti	Judul Penelitian	Ringkasan	Metode	Hasil Penelitian
<p><b>DOI:</b> <a href="https://doi.org/10.35889/progresif.v21i1.2548">10.35889/progresif.v21i1.2548</a></p>		<p>LKS, buku, kursus, dan kebutuhan lainnya, dikembangkan dengan metode <i>Waterfall</i> dan dianalisis menggunakan pendekatan <i>PIECES</i>.</p>		<p>akurat, serta memungkinkan guru dan orang tua memantau data secara transparan.</p>
<p>Az Zahra Dwi Nur Adiya, Dea Lili Anggraeni , Muhammad Akhmal Al Raidhan, Tarwoto.</p> <p>Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Amikom Purwokerto, Kota Purwokerto, Provinsi Jawa Tengah, Indonesia.</p> <p>Volume 9 E - I S S N : 2580 - 1 6 4 3 (2), April-June 2025, 467-476. E - I S S N : 2580 - 1 6 4 3.</p> <p><b>DOI:</b><a href="https://doi.org/10.35870/jtik.v9i2.3321">10.35870/jtik.v9i2.3321</a></p>	<p>Perancangan <i>Website</i> Pemesanan Menggunakan Metode <i>Waterfall</i> Pada Cafe Ungu Universitas Amikom Purwokerto</p>	<p>Penelitian ini mengembangkan sistem informasi pemesanan berbasis web untuk Cafe Ungu Universitas Amikom Purwokerto guna mengatasi antrean panjang dan kesalahan pencatatan pada metode pemesanan konvensional. Sistem dirancang menggunakan metode <i>Waterfall</i>, mencakup analisis kebutuhan, desain, pengembangan , pengujian, dan pemeliharaan.</p>	<p>Metode <i>Waterfall</i></p>	<p>Hasil implementasi menunjukkan bahwa sistem ini meningkatkan efisiensi operasional, mengurangi kesalahan pemesanan, dan meningkatkan kepuasan pelanggan. Dengan adanya sistem ini, Cafe Ungu dapat mengoptimalkan manajemen pemesanan di era digital.</p>

Peneliti	Judul Penelitian	Ringkasan	Metode	Hasil Penelitian
<p>Niken Noviardiana, Sulastri.</p> <p>Program Studi Sistem Informasi, Universitas Stikubank Semarang.</p> <p>VOL.7 NO.1 ISSN: 2656-7407 (Online).</p> <p><b>DOI:</b> <a href="https://doi.org/10.31849/zn.v7i1.24951">https://doi.org/10.31849/zn.v7i1.24951</a></p>	<p>Penerapan Metode <i>Waterfall</i> Dalam Rancang Bangun <i>E-Katalog</i> Berbasis Website Pada Studio <i>Verasalsa</i> Kecantikan.</p>	<p>Penelitian ini mengembangkan sistem reservasi berbasis web untuk Verasalsa Studio Kecantikan guna menggantikan sistem manual dalam reservasi, pengelolaan produk, data pelanggan, dan laporan. Sistem dibangun menggunakan <i>Framework CodeIgniter 3</i>, <i>database MySQL</i>, dan dikembangkan dengan metode <i>Waterfall</i>.</p>	<p>Metode <i>waterfall</i></p>	<p>Pengujian dengan <i>Black Box Testing</i> menunjukkan sistem berjalan dengan baik. UAT (<i>User Acceptance Testing</i>) yang melibatkan 12 responden menghasilkan tingkat kepuasan 85%, menunjukkan bahwa sistem ini berfungsi sesuai kebutuhan pengguna dan meningkatkan efisiensi operasional.</p>
<p>Daffa Nazza Pryatama, Novi Rukhviyanti.</p> <p>Sekolah Tinggi Manajemen Informatika dan Komputer IM, Indonesia.</p> <p>Vol. 7 No.01 2025 E-ISSN: 2685-6921.</p> <p><b>DOI:</b> <a href="https://doi.org/10.53863/kst.v7i01.1488">https://doi.org/10.53863/kst.v7i01.1488</a></p>	<p>Rancang Bangun Aplikasi Stok Barang dengan <i>QRcode</i> Menggunakan Metode <i>Waterfall</i> dan <i>Framwork Laravel</i> pada Konveksi Sfgiandra</p>	<p>Penelitian ini mengembangkan aplikasi stok barang berbasis web untuk Konveksi SFGiandra guna menggantikan sistem manual dalam pengelolaan inventaris, transaksi penjualan, dan pelaporan stok. Sistem dibangun</p>	<p>Metode <i>Waterfall</i></p>	<p>Pengujian menggunakan <i>Black Box Testing</i> menunjukkan sistem dapat meningkatkan efisiensi dan akurasi dalam pengelolaan stok. Aplikasi ini memungkinkan pengecekan barang lebih efektif dan cepat, sehingga membantu Konveksi</p>

Peneliti	Judul Penelitian	Ringkasan	Metode	Hasil Penelitian
		menggunakan <i>Framework Laravel</i> , dengan diagram UML sebagai perancangannya dan dikembangkan menggunakan metode <i>Waterfall</i> .		SFgiandra mengoptimalkan proses bisnisnya.
<p>Ali Mulyanto, Nur Fazal Hakim.</p> <p>Teknik Informatika, Universitas Panca Sakti Bekasi, Bekasi.</p> <p>Volume IV No.1, Februari 2025. e-ISSN: 2964- p-ISSN: 2964-9692.</p> <p><a href="https://www.journal.almuslim.ac.id/index.php/almuisy/article/download/192/135/520">https://www.journal.almuslim.ac.id/index.php/almuisy/article/download/192/135/520</a></p>	<p>Penerapan Metode <i>Waterfall</i> pada Rancang Bangun Aplikasi Penjualan Dan Manajemen Stock di Nahata Sablon</p>	<p>Penelitian ini mengembangkan aplikasi penjualan dan manajemen stok berbasis web untuk Toko Nahata Sablon, yang sebelumnya menggunakan pencatatan manual dalam buku ledger. Sistem dikembangkan menggunakan metode <i>Waterfall</i> dan <i>Kodular</i> sebagai alat pengembangan aplikasi berbasis Android.</p>	<p>Metode <i>Waterfall</i></p>	<p>Hasil pengujian menunjukkan bahwa aplikasi yang dikembangkan berjalan lancar dan efektif dalam mengelola penjualan serta stok barang. Sistem ini meningkatkan efisiensi operasional toko serta membantu pengelola dalam mengambil keputusan terkait persediaan dan penjualan.</p>
<p>Yoyakim Mandataris Duha, Raudhah, Ayu Nuriana Sebayang, STMIK Logika, Medan, Indonesia.</p>	<p>Perancangan Sistem Inventaris Barang Berbasis Web Menggunakan Metode <i>Waterfall</i> (Studi</p>	<p>Penelitian ini mengembangkan sistem inventaris barang berbasis web untuk STMIK Logika Medan, yang sebelumnya</p>	<p>Metode <i>Waterfall</i></p>	<p>Hasil penelitian menunjukkan bahwa sistem inventaris yang dikembangkan dapat diakses secara offline, memiliki fitur pencatatan barang,</p>

Peneliti	Judul Penelitian	Ringkasan	Metode	Hasil Penelitian
<a href="https://ejurnal.stmikmethodistbinjai.ac.id/jil/article/view/81">https://ejurnal.stmikmethodistbinjai.ac.id/jil/article/view/81</a>	Kasus : STMIK Logika Medan)	<p>masih menggunakan pencatatan manual. Sistem ini dikembangkan dengan metode <i>Waterfall</i>, mencakup tahap analisis kebutuhan, desain, implementasi, pengujian, dan pemeliharaan</p>		<p>pelaporan stok, dan pengelolaan data secara terpusat. Sistem ini meningkatkan efisiensi dan akurasi dalam pengelolaan inventaris di kampus.</p>
Adrian Rizki Maulana	<p>Rancang Bangun Aplikasi Berbasis Web untuk Optimalisasi Sistem Reservasi dan Manajemen Layanan Salon Kecantikan Purigeulis by Rddg Majalengka Menggunakan Metode Waterfall</p>	<p>Penelitian ini bertujuan untuk membangun sistem berbasis web yang dapat meningkatkan efisiensi dalam proses reservasi dan manajemen layanan di salon kecantikan. Sistem ini dirancang untuk menggantikan proses manual yang sering menyebabkan kesalahan pencatatan, keterlambatan layanan, dan kurangnya transparansi dalam pengelolaan data pelanggan,</p>	Metode <i>Waterfall</i>	<p>Aplikasi berbasis web yang dikembangkan dapat mempermudah pelanggan dalam melakukan reservasi layanan, mengotomatiskan pencatatan transaksi, serta membantu admin dalam mengelola menu layanan, data reservasi, dan yang lainnya. Pengujian dengan metode <i>Black Box</i> menunjukkan sistem berjalan dengan baik dan sesuai dengan kebutuhan pengguna.</p>

Peneliti	Judul Penelitian	Ringkasan	Metode	Hasil Penelitian
		karyawan, serta menu layanan.		

### 2.1.2. Perbandingan Dengan Penelitian Terdahulu

Dalam penelitian ini, dilakukan perbandingan dengan beberapa penelitian terdahulu untuk memahami konteks dan kontribusi yang lebih jelas terhadap pengembangan sistem yang dilakukan. Perbandingan ini dirangkum dalam tabel berikut:

**Tabel 2. 2** *Perbandingan Dengan Penelitian Terdahulu*

Aspek	Penelitian Terdahulu	Penelitian yang Dilakukan
Fokus Aplikasi	Pengembangan sistem manajemen untuk toko online, pendaftaran pasien, sistem inventaris, dan aplikasi e-commerce.	Pengembangan sistem reservasi dan manajemen layanan salon kecantikan berbasis web.
Sasaran Pengguna	Pemilik toko, rumah sakit, sekolah, atau perusahaan untuk meningkatkan efisiensi operasional dan pengelolaan data internal.	Pelanggan salon dan manajemen layanan salon kecantikan untuk mempermudah proses reservasi dan pengelolaan layanan.
Proses yang Ditingkatkan	Proses manajemen stok, penjualan online, transaksi, dan pendaftaran pasien yang sebelumnya manual, untuk meningkatkan efisiensi dan mengurangi kesalahan.	Proses reservasi dan manajemen layanan salon yang sebelumnya manual, untuk mengurangi kesalahan pencatatan dan keterlambatan layanan.

## 2.2. Kajian Teori

Kajian teori merupakan landasan konseptual yang digunakan untuk mendukung penelitian dan pengembangan sistem yang dibahas dalam penelitian yang dilakukan. Kajian ini bertujuan untuk memperkuat pemahaman terhadap konsep-konsep yang digunakan dalam proses analisis, perancangan, dan implementasi sistem. Oleh karena itu, pada bagian selanjutnya dijelaskan teori-teori yang relevan sebagai dasar dalam membangun sistem aplikasi yang diusulkan.

### 2.2.1. Aplikasi

Aplikasi merupakan perangkat lunak yang dirancang untuk mendukung pengguna dalam menyelesaikan berbagai tugas atau mencapai tujuan tertentu. Dalam era teknologi modern, aplikasi hadir dalam berbagai bentuk, termasuk aplikasi *mobile*, *web*, dan *desktop*, yang semuanya bertujuan untuk memberikan kenyamanan dan efisiensi dalam penggunaan sehari-hari (Anendya, 2025)

Pada era digital saat ini, keberadaan aplikasi sangat penting dalam kehidupan sehari-hari. Aplikasi-aplikasi ini memiliki peran yang signifikan dalam mendukung berbagai aktivitas, seperti berkomunikasi, bekerja, serta memperluas wawasan. Salah satu jenis aplikasi yang banyak digunakan adalah aplikasi berbasis web, yang dapat dijumpai pada layanan seperti *Gmail*, platform *e-commerce*, *YouTube*, dan lain sebagainya (Kania, 2023)

Menurut Kania, dua jenis aplikasi yang paling umum ditemui adalah aplikasi web dan aplikasi *mobile*, yang sering diakses melalui perangkat ponsel pintar. Selain kedua jenis aplikasi tersebut, terdapat juga aplikasi *desktop*, yang tidak kalah penting untuk dipahami, terutama dalam konteks pengembangan aplikasi. (Kania, 2023)

Adapun perbedaan ketiga jenis aplikasi tersebut dapat dijelaskan sebagai berikut:

#### a. Aplikasi *Desktop*

Aplikasi *desktop* merupakan perangkat lunak yang dirancang untuk digunakan pada perangkat komputer atau laptop. Aplikasi ini umumnya dapat berjalan tanpa membutuhkan koneksi internet atau secara *offline*. Pengguna harus menginstal aplikasi ini terlebih dahulu pada sistem operasi perangkat. Contoh aplikasi desktop antara lain adalah *Microsoft Word*, *Excel*, *Adobe Photoshop*, dan *Smadav*.

#### b. Aplikasi Web

Aplikasi web adalah aplikasi yang berjalan melalui web *browser*, seperti *Chrome* atau *Firefox*, yang memerlukan koneksi internet atau intranet (jaringan lokal) untuk dapat berfungsi. Kelebihan utama dari aplikasi web adalah kemudahan

dalam pengaksesan serta implementasinya yang dapat diterapkan pada berbagai bidang. Beberapa contoh aplikasi web yang sering digunakan antara lain adalah *Gmail*, *Trello*, *Google Docs*, dan *YouTube*.

### c. Aplikasi Mobile

Aplikasi *mobile* adalah perangkat lunak yang dikembangkan untuk perangkat mobile, seperti ponsel, tablet, atau jam tangan pintar. Dengan berkembangnya teknologi, bahasa pemrograman khusus untuk aplikasi mobile terus mengalami kemajuan, seperti *Kotlin*, *Java*, *Swift*, dan *React Native*. Contoh aplikasi mobile yang banyak digunakan dalam kehidupan sehari-hari antara lain *WhatsApp*, *Grab*, *Instagram*, dan *Twitter*.

#### 2.2.2. Website

Menurut Laily (2022), dalam artikelnya yang diterbitkan di *katadata.co.id*, Gregorius (2000) menjelaskan bahwa “*website* adalah kumpulan halaman web yang saling terhubung, dengan *homepage* sebagai halaman utama yang diikuti oleh halaman-halaman terkait lainnya. Halaman-halaman tersebut, yang disebut *child page*, biasanya berisi *hyperlink* yang menghubungkan satu halaman dengan halaman lainnya di dalam web”. Selain itu, dalam artikel yang sama, Hakim Lukmanul (2004) menyatakan bahwa “*website* merupakan fasilitas internet yang menghubungkan dokumen, baik dalam lingkup lokal maupun jarak jauh, yang dapat diakses melalui *browser* seperti *Google Chrome* dan *Mozilla Firefox*”. Sementara itu, Sholehul Azis (2013) juga menambahkan bahwa “*website* adalah halaman informasi yang bisa diakses secara global, asalkan terhubung dengan jaringan internet, dan umumnya memuat teks, gambar, suara, serta animasi yang menarik”.

Berdasarkan berbagai pendapat para ahli di atas, *website* dapat diartikan sebagai sekumpulan halaman digital yang saling terhubung melalui *hyperlink* dan disusun dalam struktur hierarkis, dengan *homepage* sebagai halaman utama. *Website* berfungsi sebagai media untuk menyajikan informasi, berkomunikasi, dan berbagi data secara luas melalui jaringan internet. Selain berisi teks, *website* juga biasanya memuat elemen multimedia seperti gambar, suara, dan animasi untuk

meningkatkan interaksi dan menarik minat pengguna. Dengan adanya *website*, pengguna dapat mengakses informasi dari berbagai lokasi secara cepat dan mudah, menjadikan *website* sebagai komponen penting dalam dunia digital modern.

Berdasarkan artikel yang ditulis oleh Laily (2022) di *katadata.co.id*, yang merujuk pada buku CMM Website Interaktif MCMS Joomla (CMS), jenis-jenis *website* dapat diklasifikasikan berdasarkan sifat, tujuan, dan bahasa pemrogramannya.

Pertama, berdasarkan sifatnya, website dibagi menjadi:

1. **Website Dinamis**, yaitu *website* yang kontennya terus diperbarui, seperti portal berita daring.
2. **Website Statis**, yaitu *website* dengan konten yang jarang diubah, seperti halaman profil organisasi.

Kedua, berdasarkan tujuannya, website dapat dibedakan menjadi:

1. **Personal Website**, yang berisi informasi pribadi seseorang.
2. **Corporate Web**, yang dimiliki oleh perusahaan untuk menyajikan informasi bisnis.
3. **Portal Website**, yang menyediakan berbagai layanan seperti berita, email, dan jasa lainnya.
4. **Forum Website**, yang berfungsi sebagai tempat diskusi antar pengguna.

### 2.2.3. Sistem

Sistem dapat diartikan sebagai suatu kesatuan yang terdiri dari berbagai komponen atau elemen yang saling berhubungan untuk mendukung kelancaran aliran informasi, materi, atau energi demi mencapai tujuan tertentu. Dalam perspektif yang lebih luas, sistem juga mencakup kumpulan unsur yang saling berkaitan dan mempengaruhi satu sama lain dalam suatu proses yang dilakukan secara kolektif untuk mencapai target yang telah ditetapkan. Oleh karena itu, sistem dapat dipahami sebagai suatu entitas yang tersusun secara sistematis dan membentuk kesatuan yang harmonis. (Effendy *et al.*, 2023).

Beberapa ahli memberikan definisi yang berbeda mengenai sistem, namun pada dasarnya memiliki esensi yang serupa. Ludwig von Bertalanffy sebagai salah satu ahli teori sistem, menjelaskan bahwa sistem terdiri dari unsur-unsur yang saling berhubungan dan terikat dalam suatu keterkaitan, baik dengan unsur lainnya maupun dengan lingkungannya. Raport juga mengemukakan bahwa sistem merupakan kumpulan kesatuan yang memiliki perangkat hubungan yang saling terkait. Sementara itu, Ackof menyatakan bahwa sistem dapat berbentuk konseptual maupun fisik, yang terdiri atas berbagai bagian yang saling bergantung satu sama lain. Dari berbagai pandangan tersebut, dapat disimpulkan bahwa sistem merupakan kumpulan elemen yang terorganisir dan bekerja sama dalam suatu kesatuan yang terstruktur. (Frisdayanti *et al.*, 2019).

Berdasarkan berbagai konsep tentang sistem, dapat disimpulkan bahwa sistem merupakan suatu kesatuan yang terdiri dari elemen-elemen yang saling terhubung serta bekerja secara kolektif untuk mencapai tujuan yang telah ditetapkan. Keberadaan sistem tidak terlepas dari keterkaitannya dengan lingkungan eksternal, di mana sistem harus mampu beradaptasi dengan perubahan agar tetap berjalan secara optimal. Selain itu, sistem juga memiliki elemen-elemen yang saling bergantung, sehingga menjaga keseimbangan dan kesinambungan antar elemen menjadi hal yang penting agar sistem dapat berfungsi dengan baik.

#### **2.2.4. Basis Data**

Basis data merupakan sekumpulan data yang diatur secara sistematis dalam suatu sistem komputer. Biasanya, basis data terdiri dari berbagai tabel yang saling terhubung melalui relasi tertentu, sehingga memungkinkan penyimpanan, pengelolaan, dan pengambilan data secara efisien untuk mendukung berbagai kebutuhan aplikasi maupun sistem informasi dalam skala kecil hingga besar (Gunawan *et al.*, 2023). Dengan adanya basis data, proses manajemen data menjadi lebih terstruktur, aman, dan dapat diakses oleh banyak pengguna secara bersamaan tanpa mengganggu integritas data. Sedangkan menurut Rozikin, basis data merupakan sekumpulan data yang terorganisir secara terstruktur, berisi informasi yang saling terkait, sehingga memudahkan proses pengambilan data secara cepat

dan akurat sesuai dengan permintaan pengguna serta mendukung pengambilan keputusan yang efektif (Rozikin et al., 2023). Dari dua pengertian di atas, dapat disimpulkan bahwa basis data adalah kumpulan data yang saling berhubungan sedemikian rupa sehingga mudah disimpan, dimanipulasi, serta dipanggil oleh pengguna agar dapat dimanfaatkan kembali untuk berbagai keperluan, baik administratif maupun analitis.

Basis data tidak hanya menyimpan data dalam jumlah besar, tetapi juga menyediakan mekanisme untuk menjaga konsistensi, integritas, dan keamanan data. Dalam praktiknya, sebuah sistem basis data dikelola menggunakan perangkat lunak khusus yang disebut *Database Management System (DBMS)*. *DBMS* memungkinkan pengguna untuk membuat, membaca, memperbarui, dan menghapus data dengan cara yang terorganisir serta menyediakan berbagai fitur tambahan seperti *backup*, *recovery*, hingga kontrol akses untuk menjaga kerahasiaan data.

Selain itu, basis data juga memainkan peran penting dalam mendukung operasional berbagai bidang, seperti bisnis, pendidikan, kesehatan, dan pemerintahan. Misalnya, pada sistem reservasi salon kecantikan, basis data digunakan untuk menyimpan informasi pelanggan, jadwal layanan, karyawan, hingga riwayat transaksi. Dengan adanya basis data, informasi tersebut dapat diakses dan diolah secara cepat sehingga meningkatkan efisiensi pelayanan kepada pelanggan.

Struktur basis data biasanya dirancang menggunakan model tertentu, seperti model relasional, hierarki, jaringan, atau model objek. Model relasional adalah yang paling umum digunakan karena menyajikan data dalam bentuk tabel yang mudah dipahami dan dikelola. Setiap tabel memiliki baris (*record*) dan kolom (*field*), di mana hubungan antar tabel dapat diatur melalui kunci primer (*primary key*) dan kunci tamu (*foreign key*) untuk memastikan integritas referensial antar data.

Perkembangan teknologi basis data juga telah melahirkan berbagai pendekatan modern, seperti basis data *NoSQL* yang mendukung penyimpanan data

tidak terstruktur untuk aplikasi berskala besar dan dinamis. Namun, untuk sistem berskala menengah seperti aplikasi reservasi salon, basis data relasional dengan DBMS seperti *MySQL* atau *PostgreSQL* sudah sangat memadai. Pemilihan jenis basis data yang tepat menjadi faktor penting agar sistem dapat bekerja optimal, terutama dalam hal kecepatan akses dan kestabilan penyimpanan data.

### 2.2.5. *MySQL*

*MySQL* adalah sistem manajemen basis data (DBMS) berbasis SQL yang bersifat *multithread* dan *multi-user*, dengan jutaan instalasi di seluruh dunia. *MySQL* didistribusikan di bawah lisensi GPL (*General Public License*), namun juga tersedia dalam lisensi komersial untuk keperluan yang tidak sesuai dengan ketentuan GPL (Dhika *et al.*, 2019).

Sedangkan menurut Hidayah, *MySQL* adalah salah satu jenis basis data yang sering digunakan dalam pengembangan aplikasi web dinamis. *MySQL* termasuk dalam kategori RDBMS (*Relational Database Management System*) dan mendukung penggunaan bahasa pemrograman PHP, dengan sintaks SQL (*Structured Query Language*) yang sederhana dan karakter pelolosan (*escape character*) yang serupa dengan PHP (Hidayah *et al.*, 2019).

Adapun menurut Alviano, *MySQL* adalah jenis *database server* yang sangat populer dan banyak digunakan untuk membangun aplikasi web yang memerlukan basis data sebagai sumber dan pengelolaan informasi. *MySQL* bersifat *open source* dan menggunakan bahasa SQL (*Structured Query Language*) untuk mengelola datanya, serta dapat berjalan di berbagai platform seperti *Windows*, *Linux*, dan lainnya (Alviano *et al.*, 2023).

Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa *MySQL* adalah sistem manajemen basis data (DBMS) berbasis SQL yang bersifat *multithread* dan *multi-user*, serta banyak digunakan untuk mengelola data dalam aplikasi web dinamis. *MySQL* didistribusikan secara *open source* di bawah lisensi GPL (*General Public License*), namun juga tersedia dalam lisensi komersial untuk keperluan yang tidak sesuai dengan ketentuan GPL. Selain itu, *MySQL* termasuk

dalam kategori RDBMS (*Relational Database Management System*) yang mendukung penggunaan bahasa pemrograman PHP dan memiliki sintaks SQL (*Structured Query Language*) yang sederhana, serta dapat berjalan di berbagai platform seperti *Windows*, *Linux*, dan lainnya.

#### **2.2.6. Metode *Waterfall***

Metode *Waterfall* adalah pendekatan yang sering digunakan dalam pengembangan perangkat lunak, yang dikenal dengan sifatnya yang linier dan terstruktur. Terdapat beberapa pendapat yang menjelaskan karakteristik dan penerapan metode ini dalam proses pengembangan sistem.

Pertama, menurut Supiyandi, model *Waterfall* sering disebut sebagai model tradisional atau klasik, dan dikenal sebagai model sekuensial linier. Prosesnya dimulai dengan analisis kebutuhan, kemudian berlanjut ke tahap desain, pengkodean, pengujian, dan akhirnya tahap pendukung. Keunggulan utama dari metode ini adalah setiap tahap dilakukan secara berurutan, dengan setiap langkah yang sebelumnya diselesaikan sebagai dasar untuk melanjutkan ke tahap berikutnya. Pendekatan ini memudahkan dalam pemantauan dan pengelolaan proyek karena jelas dan mudah dipahami (Supiyandi *et al.*, 2022).

Kedua, menurut Salam dan Septanto, *Waterfall* adalah metode yang mengharuskan pengembangan dilakukan secara berurutan atau linear. Setiap tahap dalam proses hanya dapat dilanjutkan setelah tahap sebelumnya selesai, sehingga ada ketergantungan yang kuat antar tahapan. Dalam praktiknya, hal ini memastikan bahwa pekerjaan yang dilakukan tidak tumpang tindih, tetapi memerlukan waktu yang lebih lama untuk setiap tahapan dan bisa lebih kaku jika ada perubahan di tengah jalan (Salam dan Septanto, 2024).

Ketiga, Firmansyah menjelaskan bahwa metode *Waterfall* adalah pendekatan yang sistematis, dimulai dengan analisis kebutuhan sistem dan pengguna. Setelah itu, perencanaan seperti desain dan pengkodean dilakukan, yang diikuti oleh pengujian dan pemeliharaan. Keuntungan dari pendekatan ini adalah adanya proses yang jelas dan terstruktur, namun kekurangannya adalah kurang

fleksibel dalam menghadapi perubahan kebutuhan atau masalah yang muncul setelah tahap awal (Firmansyah *et al.*, 2023).

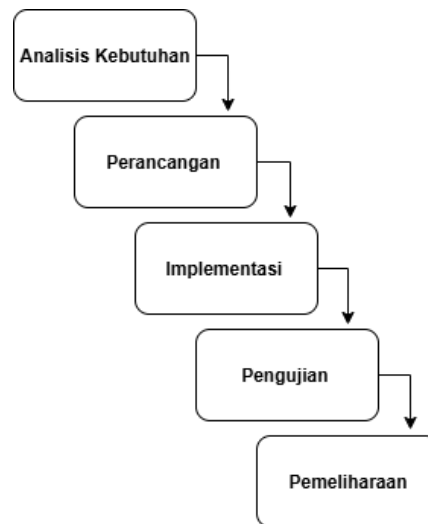
Berdasarkan beberapa pendapat di atas, metode *Waterfall* menawarkan pendekatan pengembangan perangkat lunak yang sistematis dan terstruktur, dengan tahapan yang dilaksanakan secara berurutan. Hal ini memungkinkan setiap fase untuk diselesaikan dengan jelas sebelum melanjutkan ke fase berikutnya. Namun, metode ini juga memiliki kekurangan terkait fleksibilitasnya, karena perubahan yang terjadi di tengah jalan dapat mengganggu kelancaran proses. Oleh karena itu, metode *Waterfall* lebih cocok diterapkan pada proyek yang memiliki kebutuhan yang sudah jelas dan tidak banyak berubah seiring waktu. Meskipun demikian, model ini mungkin kurang sesuai untuk proyek yang dinamis, di mana perubahan kebutuhan bisa terjadi selama pengembangan.

Sebagai perbandingan, berikut ini adalah tabel perbedaan antara metode *Waterfall*, *Agile*, dan *Spiral*:

**Tabel 2. 3** Perbandingan Metode *Waterfall* Dengan Metode Lainnya

Aspek	Waterfall	Agile	Spiral
<b>Pendekatan</b>	Linear dan sekuensial	Iteratif dan incremental	Iteratif dengan fokus pada manajemen risiko
<b>Fleksibilitas</b>	Rendah, sulit menerima perubahan setelah tahap dimulai	Tinggi, mudah beradaptasi terhadap perubahan kebutuhan	Sedang, fleksibel dengan pengulangan setiap fase
<b>Keterlibatan Klien</b>	Rendah, hanya pada awal dan akhir proyek	Tinggi, klien terlibat aktif di setiap iterasi	Sedang, klien dilibatkan pada setiap siklus
<b>Cocok Untuk</b>	Proyek dengan kebutuhan stabil dan jarang berubah	Proyek dinamis dengan kebutuhan sering berubah	Proyek besar/kompleks dengan risiko tinggi
<b>Keunggulan</b>	Proses jelas dan mudah dipantau	Cepat menghadirkan hasil, adaptif, kolaboratif	Fokus pada identifikasi risiko dan pengendalian biaya
<b>Kelemahan</b>	Tidak fleksibel, sulit mengatasi perubahan	Membutuhkan komitmen tinggi dari tim dan klien	Bisa lebih mahal dan memakan waktu jika banyak iterasi

Metode Waterfall ini terdiri dari beberapa tahapan utama yang harus dilakukan secara berurutan, yaitu sebagai berikut:



**Gambar 2. 1** Tahapan Dalam Metode Waterfall (Firmansyah, 2023)

1. **Analisis Kebutuhan** : Mengidentifikasi dan menganalisis kebutuhan sistem berdasarkan hasil observasi dan wawancara.
2. **Perancangan Sistem** : Membuat rancangan sistem yang akan diimplementasikan.
3. **Implementasi** : Mengembangkan aplikasi berbasis web berdasarkan desain yang telah dibuat.
4. **Pengujian** : Melakukan uji coba terhadap sistem untuk memastikan bahwa *fitur* yang dikembangkan berfungsi sesuai dengan kebutuhan pengguna.
5. **Pemeliharaan** : Mengevaluasi serta memperbaiki sistem berdasarkan hasil pengujian dan masukan dari pengguna.

### 2.2.7. *Unified Modeling Language (UML)*

Unified Modeling Language (UML) merupakan sebuah pendekatan yang digunakan dalam perancangan sistem untuk pengembangan aplikasi berbasis *Object Oriented Programming (OOP)*. UML berfungsi untuk menganalisis proses bisnis dan memodelkannya menggunakan berbagai notasi yang dirancang untuk memvisualisasikan entitas dalam OOP serta hubungan di antara entitas tersebut.

Selain itu, UML mampu menggambarkan sistem dalam aspek dinamis maupun statis (Aminuddin, 2019)

Dalam penelitian ini, UML yang digunakan meliputi *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram*. Diagram-diagram ini berfungsi untuk memodelkan berbagai aspek sistem, mulai dari interaksi pengguna, alur proses, urutan pesan, hingga struktur kelas dalam aplikasi yang dikembangkan. Berikut ini penjelasan mengenai masing-masing diagram:

**a. *Use Case Diagram***

Menurut Juliarto, *Use Case Diagram* adalah salah satu jenis diagram dalam *Unified Modeling Language (UML)* yang berfungsi untuk menggambarkan interaksi antara aktor dan sistem. Diagram ini mendeskripsikan jenis interaksi yang terjadi antara pengguna sistem dengan sistem itu sendiri (Juliarto, 2021).



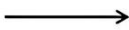

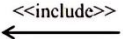
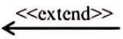
Beberapa fungsi utama *Use Case Diagram* meliputi:

1. Memperlihatkan urutan aktivitas dalam sistem secara jelas.
2. Menggambarkan proses bisnis dan menampilkan langkah-langkah aktivitas dalam suatu proses.
3. Berperan sebagai penghubung antara pengembang dan pengguna untuk mendeskripsikan sistem secara lebih mudah.

Selain itu, *Use Case Diagram* juga memiliki beberapa manfaat, antara lain:

1. Memfasilitasi verifikasi kebutuhan sistem.
2. Menyajikan gambaran antarmuka dari sistem, karena setiap sistem yang dibangun harus memiliki antarmuka yang jelas.
3. Mengidentifikasi siapa saja yang dapat berinteraksi dengan sistem dan apa saja yang bisa dilakukan oleh sistem.
4. Memastikan kebutuhan sistem telah dipahami dengan baik, sehingga mengurangi potensi kebingungan.
5. Memudahkan komunikasi antara *domain expert* dan pengguna akhir (*end user*).

*Use Case Diagram* memiliki simbol-simbol yang berfungsi untuk merepresentasikan berbagai elemen dalam pemodelan interaksi antara pengguna dan sistem. Berikut merupakan simbol-simbol yang terdapat pada *Use Case Diagram*:

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

**Gambar 2. 2** Simbol-simbol *use case diagram* (Juliarto, 2021)

#### b. *Activity Diagram*



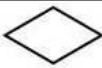



Menurut Sugiarti, *Activity Diagram* adalah jenis diagram yang digunakan untuk memodelkan berbagai proses dalam suatu sistem, dengan urutan aktivitas yang digambarkan secara vertikal. Diagram ini merupakan pengembangan dari *Use Case Diagram* dan berfungsi untuk memperlihatkan alur aktivitas atau langkah-langkah dalam proses bisnis yang ada pada sistem tersebut (Sugiarti, 2024).

*Activity Diagram* memiliki beberapa tujuan, diantaranya sebagai berikut:

1. Menggambarkan urutan aktivitas dalam suatu proses.
2. Memodelkan alur proses bisnis dalam dunia bisnis.

3. Memudahkan pemahaman terhadap keseluruhan proses dalam sebuah sistem.
4. Menyediakan metode perencanaan yang terstruktur, mirip dengan *flowchart* atau *Data Flow Diagram* (DFD).
5. Menunjukkan aktivitas yang dilakukan pengguna berdasarkan *use case* atau diagram sebelumnya.

*Activity Diagram* memiliki simbol-simbol yang berfungsi untuk merepresentasikan berbagai elemen dalam pemodelan alur aktivitas dalam suatu sistem. Berikut merupakan simbol-simbol yang terdapat pada *Activity Diagram*:

Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

**Gambar 2. 3** Simbol-simbol Activity Diagram (Sugiarti, 2024)

### c. *Sequence Diagram*

Menurut Mufid, *Sequence Diagram* adalah jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk memvisualisasikan alur interaksi antara objek atau komponen dalam suatu sistem. Diagram ini berfokus pada urutan pesan yang dikirim antar objek untuk menyelesaikan suatu proses atau kasus penggunaan (*use case*). *Sequence Diagram* menggambarkan interaksi dalam dua dimensi: vertikal untuk menunjukkan waktu dan horizontal untuk menunjukkan

objek yang terlibat dalam komunikasi. Diagram ini biasanya melibatkan aktor, objek, dan pesan untuk merepresentasikan komunikasi antara komponen sistem (Mufid, 2023).







Tujuan dari *Sequence Diagram* meliputi:

1. **Representasi Visual** - Memperjelas aliran proses atau interaksi sistem untuk memudahkan pemahaman fungsionalitas sistem.
2. **Memodelkan Langkah-Langkah** - Memetakan setiap langkah dalam proses untuk mengidentifikasi potensi masalah dan menyusun alur logis yang efisien.
3. **Memantau Proses** - Memudahkan pemantauan interaksi antar objek dalam sistem, membantu menemukan area yang memerlukan perbaikan.
4. **Memahami Kebutuhan Sistem** - Mengidentifikasi kebutuhan sumber daya tambahan dan persyaratan sistem berdasarkan skenario yang ada.

Komponen utama dalam *Sequence Diagram* antara lain:

1. **Aktor** - Simbol manusia atau pengguna yang berinteraksi dengan sistem, biasanya digambarkan dengan figur manusia.
2. **Activation Box** - Menunjukkan periode waktu ketika suatu objek aktif, digambarkan dengan persegi panjang vertikal.
3. **Lifeline** - Garis putus-putus yang menghubungkan objek dengan *activation box* untuk menunjukkan durasi objek hidup dalam sistem.
4. **Objek** - Komponen yang terlibat dalam proses, digambarkan sebagai kotak dengan nama objek di bawahnya.
5. **Messages** - Garis panah yang menunjukkan pesan yang dikirim antar objek, menandakan urutan interaksi dalam diagram.

Untuk menunjukkan hubungan waktu dengan objek dan perilakunya, *sequence diagram* memakai simbol-simbol khusus. Beberapa daftar simbol *sequence diagram* adalah sebagai berikut:

Gambar	Nama	Keterangan
	Entity Class	Gambaran sistem sebagai landasan dalam menyusun basis data
	Boundary Class	Menangani komunikasi antar lingkungan sistem
	Control Class	Bertanggung jawab terhadap kelas-kelas terhadap objek yang berisi logika
	Recursive	Pesan untuk dirinya
	Activation	Mewakili proses durasi aktivasi sebuah operasi
	Life Line	Komponen yang digambarkan garis putus terhubung dengan objek

**Gambar 2. 4** Simbol-simbol Sequence Diagram (Mufid, 2023)

**d. Deployment Diagram**

*Deployment diagram* merupakan salah satu jenis diagram dalam *Unified Modeling Language (UML)* yang digunakan untuk memvisualisasikan penempatan komponen perangkat lunak ke dalam elemen perangkat keras dalam sistem. Diagram ini memfokuskan pada bagaimana perangkat keras saling terhubung serta bagaimana proses aplikasi dijalankan pada masing-masing node (Prameswari, 2024).

Dalam sistem yang kompleks, *deployment diagram* sangat bermanfaat untuk menggambarkan relasi antara perangkat keras, jaringan, dan perangkat lunak secara terstruktur dan menyeluruh. Diagram ini memperjelas bagaimana elemen-elemen tersebut berkolaborasi dalam sebuah arsitektur sistem.

Adapun komponen umum dalam *deployment diagram*, menurut Prameswari (2024), terdiri dari:

1. **Client (Perangkat Pengguna):** Node ini mewakili perangkat seperti laptop atau smartphone yang digunakan pengguna akhir untuk mengakses aplikasi web. Akses dilakukan melalui web *browser* atau aplikasi klien lainnya.
2. **Application Server:** Node ini berfungsi sebagai tempat utama di mana aplikasi web dijalankan. Komponen inti dari aplikasi berada di *server* ini, termasuk pengelolaan permintaan yang masuk dari *client*.
3. **Database Server:** Berperan dalam menyimpan serta mengelola data aplikasi. Sistem basis data seperti *MySQL*, *PostgreSQL*, atau *Oracle* biasanya ditempatkan pada server ini, yang akan diakses oleh server aplikasi untuk proses query data.
4. **Jaringan (Network):** Komunikasi antar node berlangsung melalui jaringan, umumnya menggunakan protokol HTTP/HTTPS untuk komunikasi antara client dan application server, serta protokol SQL antara *application server* dan *database server*.

#### 2.2.8. Entity Relationship Diagram

*Entity Relationship Diagram* (ERD) merupakan alat penting dalam perancangan basis data yang menggambarkan hubungan antara entitas dalam sistem informasi. Berdasarkan beberapa pendapat, ERD adalah teknik yang digunakan untuk merancang dan memodelkan hubungan antara entitas dan atribut yang ada dalam *database*.

Pulungan menjelaskan bahwa ERD berfungsi untuk merepresentasikan data secara grafis dalam bentuk entitas yang saling berhubungan, dan sangat berperan dalam proses awal perancangan basis data. Dalam hal ini, ERD tidak hanya sebagai alat untuk mendesain *database* tetapi juga untuk menganalisis kebutuhan data pengguna (Pulungan *et al.*, 2023).








Afiifah menambahkan bahwa ERD digunakan dalam tahap awal pembuatan *database* untuk memastikan bahwa data yang akan disimpan dalam sistem dapat dikelola dengan baik. Mereka menjelaskan bahwa dengan menggunakan notasi yang jelas, seperti entitas, atribut, dan relasi, ERD memberikan gambaran yang jelas

mengenai struktur data yang akan digunakan dalam sistem. Dengan kata lain, ERD berfungsi untuk memetakan kebutuhan data dengan cara yang lebih sistematis (Afiifah *et al.*, 2022).

Selain itu, Latukolan mengungkapkan bahwa ERD sering digunakan untuk mendukung pemodelan data otomatis dengan notasi *Crow's Foot* yang mudah dipahami. Notasi ini diakui sebagai salah satu yang paling efektif karena memungkinkan pengembang untuk menggambarkan hubungan antar entitas dalam bentuk yang lebih jelas dan sederhana. Ini juga penting dalam menghindari kesalahan desain yang dapat muncul di kemudian hari (Latukolan *et al.*, 2019).

Dari berbagai sumber tersebut, dapat disimpulkan bahwa ERD adalah sebuah teknik yang sangat penting dalam perancangan *database* karena memberikan gambaran yang jelas mengenai struktur data dan hubungan antar data yang akan dikelola. Penggunaan ERD dapat mengurangi kesalahan dalam desain *database* dan mempermudah pengembang dalam merancang sistem yang lebih efisien.

Berikut ini adalah simbol-simbol dalam ERD yang digunakan untuk menggambarkan entitas, hubungan, dan kardinalitas dalam diagram.

Gambar	Nama	Keterangan
	Entity	Objek diluar sistem yang berkomunikasi dengan sistem sebagai sumber maupun pemakai informasi
	Relationship line	Garis penghubung yang di gunakan pada diagram
	Option symbol	Di gunakan untuk relasi fungsional
	One Symbol	Di gunakan pada relasi menunjukkan 1(One)
	Many	Di gunakan pada relasi untuk menunjukkan banyak(many)
	Composite Entity	Menggambarkan sebuah entitas yang komposit
	Weak Entity	Entitas yang kemunculannya tergantung dari entitas lain yang lebih kuat

**Gambar 2. 5** Simbol -Simbol ERD (Bahri, 2015)

### 2.2.9. *User Interface (UI)*

User Interface (*UI*) merupakan elemen penting dalam desain interaktif yang berfungsi sebagai penghubung antara pengguna dan sistem. Menurut Prasetyo dan Ariesta, *UI* adalah cara program dan pengguna berkomunikasi. Mereka menjelaskan bahwa istilah *user interface* atau *interface* sering digunakan sebagai pengganti istilah *HCI (Human-Computer Interaction)*, yang mencakup semua aspek interaksi antara pengguna dan komputer. Semua elemen yang terlihat di layar, seperti teks, ikon, dan tombol, serta interaksi pengguna menggunakan perangkat input seperti keyboard dan mouse, merupakan bagian dari *UI* (Prasetyo dan Ariesta, 2023).

Selain itu, Sabilissalam dalam penelitiannya mengenai aplikasi mobile menekankan bahwa *UI* adalah komponen yang menggabungkan berbagai fitur interaktif dan visual untuk meningkatkan pengalaman pengguna. *UI* memainkan peran penting dalam membuat aplikasi seluler lebih mudah diakses dan lebih menyenangkan untuk digunakan. Mereka juga menambahkan bahwa *UI* harus dirancang dengan mempertimbangkan kenyamanan dan kebutuhan pengguna, dengan tujuan menciptakan interaksi yang efisien dan efektif (Sabilissalam et al., 2023).

Kuntoro dan Mogi lebih lanjut menjelaskan bahwa *UI* adalah tampilan aplikasi yang mencakup elemen-elemen visual, seperti tulisan, gambar, dan warna, yang dirancang untuk menarik perhatian pengguna dan mempermudah interaksi. Mereka menyatakan bahwa *UI* yang baik dapat meningkatkan kepuasan pengguna dengan memfasilitasi navigasi yang lebih mudah dan intuitif di dalam aplikasi, baik itu aplikasi berbasis web maupun mobile (Kuntoro dan Mogi, 2022).

Berdasarkan pendapat dari beberapa ahli di atas, dapat disimpulkan bahwa *User Interface (UI)* bukan hanya tentang tampilan visual, tetapi juga mencakup keseluruhan elemen yang memungkinkan interaksi pengguna dengan sistem. Desain *UI* yang baik memerlukan perhatian terhadap fungsionalitas, kemudahan penggunaan, dan pengalaman pengguna secara keseluruhan. *UI* harus dirancang

dengan tujuan untuk memfasilitasi interaksi yang efisien dan memberikan pengalaman yang menyenangkan bagi penggunanya.

Selain aspek visual, desain *UI* juga harus memperhatikan konsistensi antar elemen agar pengguna dapat dengan mudah mengenali fungsi setiap komponen. Konsistensi ini mencakup penggunaan warna, tipografi, ikonografi, serta tata letak yang seragam di seluruh halaman aplikasi. Dengan konsistensi yang baik, pengguna tidak perlu belajar ulang saat berpindah dari satu halaman ke halaman lain, sehingga meningkatkan efektivitas interaksi dan meminimalkan kesalahan penggunaan.

*UI* yang dirancang dengan baik juga berperan dalam meningkatkan aksesibilitas aplikasi. Aksesibilitas ini penting untuk memastikan bahwa aplikasi dapat digunakan oleh semua kelompok pengguna, termasuk mereka yang memiliki keterbatasan fisik atau kognitif. Misalnya, penggunaan kontras warna yang cukup, ukuran font yang dapat diubah, serta dukungan terhadap teknologi pembaca layar menjadi bagian penting dalam desain *UI* yang inklusif.

Dalam konteks aplikasi berbasis web, *UI* juga harus dioptimalkan untuk berbagai ukuran layar dan perangkat. Konsep desain responsif (*responsive design*) memungkinkan elemen *UI* menyesuaikan diri secara otomatis dengan resolusi layar yang berbeda, mulai dari perangkat mobile hingga desktop. Dengan demikian, pengalaman pengguna tetap optimal tanpa harus membuat versi aplikasi yang berbeda untuk setiap perangkat.

Selain itu, keterlibatan pengguna dalam proses desain *UI* melalui metode seperti *user testing* atau *feedback loop* sangat penting. Pendekatan ini memungkinkan desainer untuk memahami kebutuhan dan kebiasaan pengguna secara langsung, sehingga hasil desain *UI* benar-benar sesuai dengan harapan pengguna. Desain yang berbasis pada masukan pengguna cenderung lebih intuitif dan dapat meningkatkan tingkat kepuasan serta loyalitas pengguna terhadap aplikasi.

### 2.2.10. *User Experience (UX)*

*User Experience (UX)* merujuk pada pengalaman yang diperoleh pengguna saat berinteraksi dengan suatu produk. *UX* mencakup aspek kemudahan penggunaan, kenyamanan saat berinteraksi, serta efektivitas produk dalam membantu pengguna mencapai tujuannya. Dalam hal ini, *UX* memiliki peran krusial dalam menentukan tingkat kepuasan pengguna serta kesuksesan suatu produk di pasar (Hendra et al., 2023).

Berdasarkan pendapat di atas, dapat disimpulkan bahwa *UX* adalah tentang bagaimana sebuah produk atau layanan mampu memberikan pengalaman yang optimal kepada penggunanya, baik dari segi fungsionalitas maupun kepuasan emosional. Ini mencakup seluruh perjalanan pengguna, mulai dari interaksi awal hingga pasca-penggunaan, dengan fokus pada kemudahan, efektivitas, dan kenyamanan.

*UX* tidak hanya berfokus pada desain visual atau antarmuka, tetapi juga melibatkan seluruh proses interaksi antara pengguna dengan sistem, termasuk persepsi, emosi, dan respons pengguna. Faktor-faktor seperti kecepatan sistem, kemudahan navigasi, dan keandalan fitur juga sangat mempengaruhi kualitas *UX*. Oleh karena itu, desain yang baik harus mempertimbangkan kebutuhan dan harapan pengguna sejak awal pengembangan produk.

Dalam praktiknya, *UX* sering diukur melalui beberapa indikator seperti kepuasan pengguna (*user satisfaction*), tingkat keberhasilan tugas (*task success rate*), efisiensi (*efficiency*), dan tingkat kesalahan pengguna (*error rate*). Penggunaan metode evaluasi seperti *usability testing* dan survei kepuasan pengguna menjadi penting untuk memastikan bahwa *UX* yang dihasilkan sesuai dengan ekspektasi target pengguna.

Selain itu, *UX* memiliki hubungan yang erat dengan *UI (User Interface)*. Meskipun keduanya sering dianggap sama, sebenarnya *UI* lebih berfokus pada elemen visual dan interaksi langsung, sedangkan *UX* mencakup keseluruhan pengalaman pengguna dengan produk. Sebuah aplikasi dapat memiliki *UI* yang

menarik tetapi gagal memberikan *UX* yang baik jika pengguna merasa sulit atau tidak nyaman saat menggunakannya. Dengan demikian, keseimbangan antara desain *UI* yang estetik dan *UX* yang fungsional menjadi kunci utama dalam pengembangan aplikasi.

Di era teknologi saat ini, *UX* menjadi salah satu faktor kompetitif yang membedakan sebuah aplikasi dengan aplikasi lainnya. Pengguna cenderung memilih aplikasi yang memberikan pengalaman penggunaan yang menyenangkan, intuitif, dan bebas hambatan. Oleh karena itu, perusahaan teknologi besar seperti Google, Apple, dan Microsoft selalu menempatkan *UX* sebagai prioritas utama dalam proses pengembangan produk mereka untuk mempertahankan loyalitas pengguna.

Selain aspek teknis, *UX* juga dipengaruhi oleh faktor psikologis dan budaya pengguna. Misalnya, preferensi warna, gaya desain, dan pola navigasi dapat berbeda antara pengguna di negara Barat dan Asia. Desainer *UX* perlu memahami konteks ini agar aplikasi yang dikembangkan mampu memenuhi ekspektasi pengguna dari berbagai latar belakang. Penyesuaian desain berbasis lokalitas atau *localization* menjadi penting untuk memastikan pengalaman pengguna tetap relevan dan nyaman.

### **2.2.11. Black Box Testing**

*Black box testing* adalah metode pengujian perangkat lunak yang fokus pada pengujian fungsionalitas aplikasi tanpa mempertimbangkan struktur internal atau kode aplikasi. Pengujian ini menggunakan deskripsi eksternal perangkat lunak, termasuk spesifikasi, persyaratan, dan desain untuk merancang kasus uji. Tes ini dapat mencakup pengujian fungsional atau non-fungsional, meskipun lebih sering dilakukan untuk menguji fungsionalitas. Pengujian ini tidak memerlukan pengetahuan tentang kode internal aplikasi atau pemrograman secara umum (Putra *et al.*, 2020).

Metode ini juga menguji aplikasi dengan mencoba memasukkan data pada berbagai form atau antarmuka yang ada, untuk memastikan bahwa *software*

berfungsi sesuai dengan ekspektasi. Pengujian dilakukan tanpa memperhatikan detail internal *software*, melainkan dengan fokus pada bagaimana aplikasi merespons input yang diberikan. Hal ini memungkinkan pengujian dilakukan pada berbagai tingkat pengujian perangkat lunak, mulai dari unit, integrasi, fungsional, hingga penerimaan sistem (Febriyanti *et al.*, 2021)

Selain itu, terdapat beberapa teknik dalam metode *black box testing*, seperti *Sample Testing*, *Boundary Value Analysis*, dan *Equivalence Partitions* yang dapat digunakan untuk meningkatkan efektivitas pengujian. Teknik-teknik ini membantu dalam mengidentifikasi potensi masalah dan memastikan bahwa aplikasi berjalan sesuai dengan fungsinya (Febrian *et al.*, 2020).

Berdasarkan beberapa definisi tersebut, dapat disimpulkan bahwa *black box testing* adalah metode pengujian yang sangat bergantung pada hasil pengujian eksternal untuk memastikan bahwa perangkat lunak berfungsi sesuai dengan tujuan yang telah ditetapkan. Metode ini memastikan aplikasi diuji dari perspektif pengguna akhir, tanpa memerlukan pemahaman mendalam tentang struktur internal aplikasi itu sendiri.

#### **2.2.12. Skala Likert**

Simamora menjelaskan bahwa *skala Likert* merupakan sebuah instrumen pengukuran yang digunakan untuk mengukur sikap, opini, atau persepsi seseorang terhadap suatu objek atau fenomena tertentu. Instrumen ini terdiri dari serangkaian pernyataan yang disusun secara sistematis, di mana responden diminta untuk memberikan tingkat kesetujuannya terhadap setiap pernyataan tersebut menggunakan skala bertingkat. Simamora menekankan bahwa *skala Likert* memungkinkan pengukuran sikap secara kuantitatif dengan variasi tingkat persetujuan yang jelas dan terstruktur, biasanya dalam rentang lima hingga tujuh poin. Hal ini membuat *skala Likert* sangat populer digunakan dalam berbagai penelitian karena kemampuannya untuk mengubah data kualitatif menjadi data yang dapat dianalisis secara statistik (Simamora, 2022).

Selain itu, Mawardi menambahkan bahwa skala Likert digunakan untuk mengukur sikap individu melalui sejumlah pernyataan yang harus dijawab responden berdasarkan tingkat kesetujuannya. Ia menegaskan bahwa penyusunan item pernyataan dalam *skala Likert* harus dilakukan dengan cermat dan terstruktur agar data yang diperoleh memiliki validitas dan reliabilitas yang tinggi. Proses ini meliputi pembuatan pernyataan yang jelas, tidak ambigu, serta relevan dengan variabel yang akan diukur. Lebih lanjut, Mawardi juga menekankan pentingnya melakukan uji coba instrumen sebelum digunakan secara luas untuk memastikan bahwa responden memahami setiap item dan dapat memberikan jawaban yang akurat dan konsisten (Mawardi, 2019).

Berdasarkan dua definisi tersebut, dapat disimpulkan bahwa skala Likert merupakan instrumen pengukuran sikap yang menggunakan serangkaian pernyataan dengan pilihan jawaban bertingkat untuk mencerminkan tingkat kesetujuan responden secara kuantitatif. Kedua pendapat tersebut sepakat bahwa *skala Likert* umumnya menggunakan lima hingga tujuh poin dalam rentang skala jawaban, yang memungkinkan tingkat perbedaan sikap yang lebih rinci dan terukur. Selain itu, validitas dan reliabilitas instrumen menjadi aspek penting yang harus diperhatikan dalam penyusunan skala Likert agar data yang diperoleh dapat dipercaya dan digunakan untuk analisis lebih lanjut dalam penelitian. Dengan demikian, penggunaan skala Likert menjadi pilihan yang tepat dalam pengukuran sikap yang membutuhkan data kuantitatif dengan tingkatan jawaban yang sistematis dan terstruktur.

Untuk memberikan gambaran yang lebih jelas mengenai keunggulan dan keterbatasan skala Likert, berikut disajikan tabel yang merangkum kelebihan dan kelemahannya:

**Tabel 2. 4** *Kelebihan Dan Kelemahan Pengujian Menggunakan Skala Likert*

Kelebihan	Kelemahan
Memudahkan responden dalam memahami dan menjawab pernyataan.	Rentan bias responden, seperti memilih jawaban tengah untuk menghindari opini ekstrem.
Data yang dihasilkan bersifat kuantitatif sehingga mudah dianalisis dengan metode statistik.	Hasil dapat terpengaruh oleh interpretasi responden terhadap pernyataan yang ambigu.
Fleksibel dan dapat digunakan untuk berbagai jenis penelitian sosial maupun eksperimental.	Tidak semua responden memberikan tingkat kejujuran yang sama pada setiap item pernyataan.

Dalam penelitian ini, skala Likert digunakan sebagai instrumen untuk mengukur tingkat kepuasan pengguna terhadap sistem reservasi dan manajemen layanan berbasis web di Purigeulis. Penggunaan skala ini memungkinkan konversi persepsi pengguna menjadi data kuantitatif yang dapat dianalisis untuk mengevaluasi efektivitas sistem yang dikembangkan.