

## BAB II

### LANDASAN TEORI

#### 2.1 Penelitian Terdahulu

Berikut merupakan tabel penelitian terdahulu yang berhubungan dengan sistem penjualan berbasis web yang akan dijadikan referensi dan acuan dasar pada penelitian ini.

Tabel 2.1 Penelitian Terdahulu

No	Judul Jurnal	Nama Jurnal	Penulis / Tahun	Hasil	Kesimpulan
1	<i>E-Commerce System for Media Group Cooperative</i>	Ultimatics: Jurnal Teknik Informatika / Vol. XV / No. 1	Mikha Melvin Agustin, Ririn Ikana Desanti, Wella / 2023	Hasil dari penelitian ini adalah sistem ini dapat memperluas pemasaran produk yang dijual oleh koperasi dan dapat diakses dengan cepat, sehingga dapat meningkatkan penjualan dan pesanan untuk produk koperasi. Beberapa perbedaan antara koperasi sebelum dan setelah menggunakan sistem ini termasuk penjualan produk secara online, memudahkan pemantauan transaksi secara real-time, dan pencatatan penjualan yang dilakukan oleh sistem untuk memudahkan administrator melihat ringkasan penjualan produk secara real-time	Pengembangan sistem e-commerce menggunakan metode waterfall dapat membantu meningkatkan kinerja koperasi dalam melakukan transaksi online dan menyebarkan informasi kepada anggota tentang program-program terkini. Dalam proses pengembangan sistem ini, dilakukan wawancara mendalam dengan koperasi karyawan Media Group untuk memperoleh informasi yang diperlukan. Metode waterfall digunakan secara berurutan untuk menciptakan alur proses sistem yang akan diterapkan dalam koperasi. Sistem ini dirancang

No	Judul Jurnal	Nama Jurnal	Penulis / Tahun	Hasil	Kesimpulan
					untuk memudahkan penjualan dan informasi kepada anggota koperasi
2	<i>Design and Built the Kuyngebasreng Website Using the Waterfall Method</i>	<i>International Journal for Applied Information Management</i> / Vol. III / No.4	Evelyn Gina Megananda, Fitri Is'aaf Khairunisa, Septiya Nur Fadillah, Siti Saekhhah Ali, Taewoto / 2023	Hasil penelitian menunjukkan bahwa penggunaan metode Waterfall dalam pengembangan website ini dapat membantu Kuyngebasreng dalam mengatasi tantangan pemasaran yang sempit, meningkatkan layanan pelanggan, dan menyederhanakan proses pembelian produk basreng. Dengan adanya website, Kuyngebasreng dapat bersaing dalam bisnis penjualan basreng dengan menyediakan informasi yang cepat, akurat, dan menarik bagi pelanggan	Penggunaan metode Waterfall dalam pengembangan website untuk Kuyngebasreng telah terbukti efektif dalam membantu UKM mengatasi tantangan pemasaran yang sempit, mencapai target penjualan, dan memudahkan proses pembelian produk bagi konsumen. Melalui penggunaan perangkat lunak WordPress sebagai CMS, Kuyngebasreng dapat memanfaatkan fleksibilitas platform tersebut untuk mempromosikan produk basreng mereka secara online, sehingga membantu meningkatkan visibilitas dan penjualan produk mereka
3	Analisis dan Perancangan Sistem Informasi <i>Inventory</i> Berbasis Web (Studi Kasus: PT Meco Teknokonsultan Sinergi)	Jurnal Esensi Infokom / Vol. IV / No. 1	Sarip Hidayatuloh, Fahrul Firdaus / 2020	Penelitian ini menghasilkan sebuah Sistem Informasi <i>Inventory</i> berbasis web yang dapat mengelola data keluar masuk barang dengan efisien. Sistem ini dirancang untuk membantu dalam pencatatan transaksi keluar masuk barang.	Sistem Informasi <i>Inventory</i> ini mampu membantu pencatatan transaksi keluar masuk barang serta memberikan informasi akurat mengenai jumlah stok yang tersedia. Dengan adanya sistem ini, pelaporan transaksi keluar masuk barang dapat disajikan dengan cepat dan akurat. Selain itu, pemberian kode SKU pada barang memungkinkan identifikasi barang dengan mudah dan mengurangi risiko tercampurnya barang.

No	Judul Jurnal	Nama Jurnal	Penulis / Tahun	Hasil	Kesimpulan
4	<i>Augmented Reality</i> Pengenalan Lingkungan Kampus II Universitas Harapan Medan dengan Metode <i>Markerless</i>	Jurnal Krisnadana / Vol. II / No. 1	Hediningtias Wulaningrum, Ihsan Lubis, Septiana Dewi Andriana / 2022	Pengembangan sebuah aplikasi <i>Augmented Reality</i> untuk smartphone Android yang mampu menampilkan animasi 3D dari gedung-gedung di Kampus 2 Universitas Harapan Medan. Aplikasi ini dibangun bahasa pemrograman C#. Pengujian aplikasi menunjukkan hasil yang baik dengan metode <i>BlackBox Testing</i> .	Aplikasi <i>Augmented Reality</i> yang dikembangkan mampu menampilkan animasi 3D gedung-gedung secara nyata kepada pengguna berbasis Android, meningkatkan daya tarik dan interaktivitas media promosi kampus. Dengan demikian, penelitian ini berhasil mengimplementasikan teknologi <i>Markerless</i> dan Unity 3D secara efektif untuk menciptakan pengalaman <i>Augmented Reality</i> yang menarik dan informatif bagi calon mahasiswa Universitas Harapan Medan.
5	Perancangan Aplikasi Klaim Produk Asuransi Instividual PT. Asuransi Jiwa Inhealth Indonesia (Mandiri Inhealth)	Jurnal Maklumatika / Vol. VII / No. 7	Fajar Basoni, Mohammad Imam Shalahudin / 2021	Hasil yang didapatkan pada penelitian ini adalah sebuah aplikasi klaim produk asuransi individu berbasis <i>website</i> yang dapat mengoptimalkan proses pengajuan klaim. Aplikasi ini membantu memudahkan pegawai asuransi dalam menjalankan tugasnya dan mengurangi risiko kesalahan dalam pengecekan data pengajuan klaim, serta membuat pelaporan menjadi lebih cepat dan tepat	Aplikasi klaim produk asuransi individu yang telah berhasil dirancang dapat mempercepat proses pengajuan klaim dan mengurangi risiko kesalahan dalam pengajuan klaim. Selain itu, peserta asuransi dapat dengan mudah mengetahui status pengajuan klaim yang mereka ajukan. Integrasi aplikasi klaim dengan <i>Financial Information System (FIS)</i> pada PT. Asuransi Jiwa Inhealth Indonesia dapat memberikan manfaat yang signifikan dalam pengelolaan klaim asuransi
6	Model <i>Waterfall</i> Untuk Pengembangan	Jurnal Teknik Komputer AMIK BSI	Amrin, Mita Diah Larasati, Irawan Satriadi / 2020	Hasil dari penelitian ini adalah dapat membantu dalam aktivitas pengolahan nilai rapor siswa secara lebih efisien,	Sistem informasi ini efektif dalam mengurangi kesalahan yang sering terjadi pada sistem manual.

No	Judul Jurnal	Nama Jurnal	Penulis / Tahun	Hasil	Kesimpulan
	Sistem Informasi Pengolahan Nilai pada SMP Kartika XI-3 Jakarta Timur			akurat, dan cepat. Selain itu, sistem ini juga diharapkan dapat meminimalisasi kesalahan dan keterlambatan dalam pengolahan data rapor siswa	Implementasi sistem ini juga dapat meningkatkan kinerja pegawai dalam pengolahan data yang sebelumnya kurang efisien, tingkat keamanan data lebih terjaga
7	Model <i>Waterfall</i> dalam Rancang Bangunan Sistem Informasi Pemesanan Produk dan Outlet Berbasis Web	JII (Jurnal Inovasi Informatika) Universitas Pradita / V / No. 2	Yopi Handrianto, Budi Sanjaya / 2020	Hasil dari penelitian ini adalah sistem yang dirancang untuk mengatasi kendala-kendala yang terjadi pada proses bisnis perusahaan, seperti proses order yang lambat dan Tata Cara manual yang memakan waktu. Dengan adanya sistem informasi ini, dapat meningkatkan efektivitas dan efisiensi dalam pengambilan keputusan, serta mempercepat proses bisnis perusahaan secara keseluruhan.	Data-data pelatihan sudah dimuat dalam Proses bisnis perusahaan dapat menjadi lebih cepat dan efisien. Fasilitas <i>online</i> yang terkoneksi dan terintegrasi memungkinkan beberapa tahapan proses dapat dipermudah, sehingga membantu dalam pengambilan keputusan yang lebih efektif.
8	<i>The Development of UMS Building Catalogue Information System</i>	Ultimatics: Jurnal Teknik Informatika / Vol. XIV / No. 1	Ganno Tribuana Kurniaji, Mohammad Faqih Eza Ammar, Yusuf Sulisty Nugroho / 2022	Proses pengujian <i>Blackbox</i> dilakukan melalui 8 skenario. Semua skenario dalam pengujian <i>Blackbox</i> berjalan dengan baik, termasuk pemeriksaan gambar bangunan yang sesuai dengan nama bangunan, pengecekan urutan nama bangunan, pengecekan ejaan nama bangunan, dan validitas informasi lainnya yang terkait dengan bangunan di UMS.	Dengan menerapkan model <i>waterfall</i> , hasil pengujian <i>acceptance</i> testing menunjukkan bahwa sistem ini dapat diterima dengan skor SUS sebesar 76.17, yang berarti sistem ini dapat diimplementasikan dan mudah digunakan oleh mahasiswa baru. Sistem katalog bangunan ini membantu mahasiswa memperoleh informasi tentang lokasi bangunan di UMS dan dapat menjadi landasan untuk pengembangan aplikasi berbasis <i>mobile</i> di masa depan.

No	Judul Jurnal	Nama Jurnal	Penulis / Tahun	Hasil	Kesimpulan
9	Rancang Bangun Aplikasi <i>e-commerce Dropship</i> Berbasis Web	Ultimatics: Jurnal Teknik Informatika / Vol. XII / No. 2	Alexander Waworuntu / 2020	Dalam perancangan sistem ini, hasilnya adalah aplikasi <i>e-Commerce Dropship</i> berbasis web yang telah berhasil dirancang dan dikembangkan menggunakan bahasa pemrograman PHP dengan <i>framework</i> Laravel serta basis data MariaDB. Proses pengujian sistem dilakukan menggunakan metode <i>Black Box Testing</i> untuk memastikan keseluruhan fungsionalitas sistem berjalan dengan baik dan benar. Hasil dari uji coba ini memberikan umpan balik yang menjadi dasar bagi proses analisis, desain, dan implementasi pada iterasi berikutnya.	Tahapan perencanaan, analisis, desain, dan implementasi telah dilalui dengan baik untuk mengakomodir kebutuhan pengguna, seperti manajemen produk, melihat transaksi penjualan, mencetak <i>invoice</i> , dan fitur lainnya. Proses implementasi dilakukan dengan uji coba sistem menggunakan metode <i>Black Box Testing</i> untuk memastikan fungsionalitas sistem berjalan dengan baik. Evaluasi dilakukan setelah implementasi dengan menggunakan kuesioner berdasarkan <i>Technology Acceptance Model (TAM)</i> untuk mengukur <i>Perceived Ease of Use</i> dan <i>Perceived Usefulness</i> aplikasi, yang menunjukkan tingkat penerimaan yang tinggi dari pengguna
10	<i>The Development of Web-based Sales Reporting Information Systems using Rapid Application Development Method</i>	Ultima Infosys: Jurnal Ilmu Sistem Informasi / Vol. XIII / No. 2	Suryasari, Jansen Wiratama, Ririn Ikana Desanti / 2022	Hasil pengujian sistem menunjukkan skor rata-rata keseluruhan sebesar 82,5%, menandakan kinerja yang sangat baik. Proses pengujian fungsionalitas sistem dilakukan sebelum implementasi sistem di PT Artindo Pratama Sejahtera menggunakan metode <i>User Acceptance Test (UAT)</i> dengan model pengujian <i>black box</i> . Hasil pengujian tersebut disesuaikan dengan kebutuhan perusahaan untuk memastikan bahwa	Kesimpulan dari proses pembuatan rancangan <i>website</i> pada file ini adalah bahwa penelitian telah dilakukan untuk mengembangkan Sistem Informasi berbasis kebutuhan digitalisasi dalam proses bisnis pelaporan penjualan di PT Artindo Pratama. Desain Sistem Informasi Pelaporan Penjualan berbasis web berhasil diimplementasikan dengan menggunakan Metode <i>Rapid Application Development (RAD)</i> . Sistem ini memberikan kontribusi

No	Judul Jurnal	Nama Jurnal	Penulis / Tahun	Hasil	Kesimpulan
				sistem dapat mengatasi masalah yang telah diidentifikasi sebelumnya.	signifikan bagi perusahaan, terutama bagi divisi penjualan, dalam melakukan pelaporan performa penjualan.



Berdasarkan penelitian terdahulu yang telah dituliskan pada tabel 2.1 terdapat artikel jurnal terdahulu yang dijadikan referensi untuk melakukan pengembangan sistem penjualan berbasis *website* menggunakan metode *waterfall*, bahasa pemrograman C#, dan pengujian menggunakan *blackbox*. Seperti, artikel jurnal nomor 1 yang ditulis oleh dosen-dosen FTI UMN dengan judul *E-Commerce System for Media Group Cooperative* [6], kemudian jurnal internasional yang terdapat pada nomor 2 dengan judul *Design and Built the Kuyngebasreng Website Using the Waterfall Method* [7], selanjutnya jurnal nomor 4 yang berjudul *Augmented Reality Pengenalan Lingkungan Kampus II Universitas Harapan Medan dengan Metode Markerless* [8], jurnal nomor 5 yang berjudul *Perancangan Aplikasi Klaim Produk Asuransi Individual PT. Asuransi Jiwa Inhealth Indonesia (Mandiri Inhealth)* [9], kemudian pada jurnal nomor 6 dengan judul *Model Waterfall Untuk Pengembangan Sistem Informasi Pengolahan Nilai pada SMP Kartika XI-3 Jakarta Timur* [10], selanjutnya terdapat jurnal nomor 7 yang berjudul *Model Waterfall dalam Rancang Bangunan Sistem Informasi Pemesanan Produk dan Outlet Berbasis Web* [11], dan terdapat jurnal nomor 8 yang berjudul *The Development of UMS Building Catalogue Information System* [12], akan dijadikan referensi untuk metodologi penelitian yang digunakan pada penelitian ini, yaitu menggunakan metode *waterfall*.

Selain itu, referensi untuk membantu proses pengembangan *website* ini yaitu penggunaan bahasa pemrograman yang cocok dalam penelitian ini. Terdapat beberapa jurnal artikel yang menjadi bahan referensi penggunaan bahasa pemrograman C#, yang pertama jurnal nomor 3 dengan judul *Analisis dan Perancangan Sistem Informasi Inventory Berbasis Web (Studi Kasus: PT Meco Teknokonsultan Sinergi)* [13], selanjutnya jurnal nomor 4 yang berjudul *Augmented Reality Pengenalan Lingkungan Kampus II Universitas Harapan Medan dengan Metode Markerless* [8], dan jurnal nomor 5 yang berjudul *Perancangan Aplikasi Klaim Produk Asuransi Individual PT. Asuransi Jiwa Inhealth Indonesia (Mandiri Inhealth)* [9].

Sedangkan referensi untuk *use case scenario* yang akan digunakan pada penelitian ini yaitu diambil dari penelitian terdahulu nomor 7 dengan judul Sistem Informasi Pelatihan pada Lembaga Ilmu Pengetahuan Indonesia Pusat Pengembangan Teknologi Tepat Guna Berbasis Web. Referensi untuk pengujian menggunakan metode *blackbox* diambil dari penelitian terdahulu yang terdapat pada nomor 4 dengan judul penelitiannya yaitu *Augmented Reality* Pengenalan Lingkungan Kampus II Universitas Harapan Medan dengan Metode *Markerless* [8], selanjutnya penelitian nomor 7 dengan judul Model *Waterfall* dalam Rancang Bangunan Sistem Informasi Pemesanan Produk dan Outlet Berbasis Web [11], kemudian terdapat pada penelitian nomor 8 yang berjudul *The Development of UMS Building Catalogue Information System* [12], selanjutnya yaitu penelitian nomor 9 yang diteliti oleh salah satu dosen dari UMN dengan judul Rancang Bangun Aplikasi *e-Commerce Dropship* Berbasis Web [14], dan terdapat pada penelitian nomor 10 yang juga merupakan hasil penelitian dari dosen-dosen UMN dengan judul penelitiannya yaitu *The Development of Web-based Sales Reporting Information Systems using Rapid Application Development Method* [15]. Berdasarkan referensi-referensi dari jurnal artikel tersebut akan dijadikan acuan untuk melakukan pengembangan sistem penjualan berbasis web menggunakan metode *waterfall* dan bahasa pemrograman C# pada PT Mitra Beton Mandiri.

## 2.2 Sistem Informasi Penjualan

Penjualan merupakan pembelian suatu barang atau jasa dari satu pihak ke pihak lain dengan tujuan memperoleh keuntungan atau keuntungan dengan membayar imbalan berupa uang dari pembeli [16]. Proses penjualan terdiri dari beberapa tahapan, dimulai dari persiapan pra penjualan, penentuan lokasi calon pembeli, pendekatan pelatihan produk atau jasa, proses jual beli, dan layanan purna jual [17]. Dikarenakan kendala sistem, seluruh proses penjualan masih bersifat *manual* sehingga menimbulkan risiko kesalahan pencatatan [18]. Perubahan lingkungan bisnis yang cepat, seperti globalisasi dan kemajuan teknologi, juga mengubah cara perusahaan melakukan aktivitas bisnis. Dunia usaha harus mampu beradaptasi

terhadap perubahan-perubahan ini dan memanfaatkan teknologi informasi untuk mendapatkan keunggulan kompetitif.

Sistem informasi penjualan adalah sistem yang membantu mengolah data dari seluruh tahapan penjualan untuk mencapai tujuan bisnis [19]. Membangun sistem informasi penjualan memerlukan perancangan, implementasi, dan modifikasi terlebih dahulu jika diperlukan, sehingga tahap konstruksi awal memakan waktu dan mahal. Namun, sistem informasi penjualan bisa sangat membantu dalam membuat keputusan berdasarkan data, memastikan bahwa tindakan yang diambil efektif dan tepat sasaran. Kebijakan ini diharapkan dapat meningkatkan kinerja penjualan di setiap tahapannya. Sistem informasi penjualan juga menyediakan akses terhadap data terkini, sehingga memungkinkan Anda untuk merangkul waktu perpindahan data dari satu bagian ke bagian lain dan meminimalkan ketidaksesuaian data yang sering terjadi akibat proses *transfer* data yang lama.

Salah satu bentuk sistem informasi penjualan adalah e-bisnis. E-bisnis adalah konsep aktivitas bisnis yang dilakukan melalui internet, yang juga mencakup pembelian, penjualan, pelayanan, mendukung pelanggan, dan bekerja sama dengan mitra bisnis individu dan organisasi [20]. Selanjutnya terdapat sistem informasi penjualan berupa perdagangan elektronik yang memerlukan sebuah *website* untuk melakukan transaksi jual beli sebagai kegiatan usahanya [21]. Sistem informasi penjualan sangat membantu dalam pengambilan keputusan berdasarkan data sehingga kebijakan yang dibuat efektif dan tepat sasaran. Aktivitas penjualan melibatkan banyak proses dan interaksi antara pelanggan, mitra bisnis, dan departemen internal. Mengelola semua aspek ini secara manual bisa jadi sulit dan memakan waktu.

Oleh karena itu, sistem informasi penjualan dan pemasaran membantu mengotomatisasi metode, meningkatkan efisiensi, dan mengurangi kesalahan manusia [22]. Sistem memungkinkan tugas-tugas berupa pengolahan data dilakukan dengan lebih mudah dibandingkan cara manual, seperti menjual, mencatat, dan melaporkan hasil [23]. Sistem penjualan memudahkan pengguna dalam mencari informasi produk yang dicarinya dan penjual dalam mengolah data

pesanan barang konsumen [24]. Selain itu, keamanan data menjadi salah satu permasalahan yang sering ditemui ketika menggunakan sistem informasi penjualan. Sistem informasi penjualan rentan terhadap serangan *cyber* dan pelanggaran data dan harus tetap aman dan dipersiapkan dengan baik [25].

### 2.3 Sistem Informasi Berbasis Web

*Website* adalah serangkaian halaman yang terhubung satu sama lain dan berisi dokumen serta gambar yang disimpan di *server web*. [26]. Pada prinsipnya sistem informasi berbasis web tidak berbeda jauh dengan sistem informasi pada umumnya. Bedanya, sistem informasi berbasis web menggunakan arsitektur *client/server* dan bersifat *online* [22]. Sistem informasi berbasis *website* merupakan suatu komputer yang mempunyai berbagai fungsi tergantung kebutuhan yang digunakan untuk menginput data tertentu, dengan tujuan untuk mempermudah, mempercepat dan mencatat data yang diproses secara akurat, karena merupakan sistem yang terstandar maka banyak digunakan [21]. *Client-server* merupakan suatu bentuk jaringan dengan *client* yang bertugas meminta data dan *server* yang bertugas memberikan informasi atau data. Sistem informasi berbasis web ini banyak digunakan untuk penyimpanan data, pengumpulan, dan tampilan informasi [22].

### 2.4 Framework Model View Controller (MVC)

*Framework* adalah kumpulan pernyataan yang diorganisasikan ke dalam suatu *class* dan *function*, masing-masing dengan memiliki fungsi yang unik dan memudahkan *user* untuk memanggilnya tanpa harus menuliskan *syntax* program yang sama secara berulang kali [24]. Ada banyak manfaat yang didapat saat membangun sebuah sistem menggunakan *framework*. Salah satu keuntungannya yaitu *framework* itu sendiri memiliki *library* dan fungsi yang dapat digunakan oleh *user* nya secara langsung saat melakukan pengembangan suatu sistem [25]. Hal tersebut berdampak kepada proses pembuatan sistem yang menjadi lebih cepat dan mudah.

Salah satu *framework* yang banyak digunakan adalah *Model View Controller* (MVC). MVC adalah suatu konsep yang diperkenalkan oleh Trygve Reenskaug

selaku penemu Smalltalk [27]. Penerapan model MVC melibatkan pembagian struktur program dengan representasi yang berbeda yang dibagi menjadi tiga bagian, yaitu *Model* yang mewakili struktur data, *View* bertanggung jawab atas presentasi sistem, dan *Controller* bertindak sebagai penghubung atau perantara antara *View* dan *Model*, memproses data dan menampilkan hasilnya pada sistem.

## 2.5 Basis Data

Basis Data (*database*) adalah kumpulan data yang saling terhubung yang mencerminkan suatu fakta-fakta yang terkandung dalam suatu organisasi [28]. *File-file* ini dapat mencakup *file* tentang internal organisasi, pemangku kepentingan yang menjadi mitra bisnis, hingga masyarakat luas yang menjadi pelanggan atau pengguna jasa dan barang yang diproduksi oleh organisasi [29]. Untuk memberikan informasi yang lebih detail, data perlu disimpan dalam basis data. Data yang terdapat di dalam suatu *database* harus diorganisasikan sedemikian rupa sehingga menghasilkan hasil informasi yang berkualitas tinggi. Penggunaan basis data dalam sistem informasi dikenal sebagai sistem basis data, yang merupakan sebuah sistem informasi yang mengelola kumpulan data yang saling terkait. [30].

*Database* dirancang untuk digunakan dengan menyimpan dan mengakses data. Pada penelitian ini digunakan *software* Microsoft SQL Server untuk menyimpan *database*. SQL Server merupakan salah satu sistem manajemen *database* berjenis *relational database management* (RDBMS) yang dikembangkan oleh perusahaan Microsoft [31]. SQL Server menggunakan bahasa SQL (*Structured Query Language*) sebagai bahasa kueri utamanya. Pengguna dapat menggunakan SQL untuk pengambilan, pembaruan, dan penghapusan data dari *database*, serta membuat dan mengelola struktur *database*. SQL Server menyediakan berbagai fitur untuk menyimpan, mengelola, dan menganalisis data.

## 2.6 Pemodelan Berorientasi Objek

Pendekatan berorientasi objek, juga dikenal sebagai metodologi OO (*Object Oriented*), merupakan pendekatan pengembangan perangkat lunak yang kuat dan efisien. Pendekatan ini memainkan peran penting dalam menciptakan sistem

perangkat lunak berkualitas tinggi. Setiap tahap berkontribusi terhadap keberhasilan keseluruhan proses pengembangan perangkat lunak. Pada tahap awal OOA, analisis menyeluruh dilakukan untuk mengidentifikasi kebutuhan perangkat lunak. Hal ini melibatkan identifikasi berbagai objek dan kelas yang relevan dengan permasalahan yang dihadapi. Setelah tahap OOA selesai, langkah selanjutnya adalah tahap OOD. Pada tahap ini, arsitektur sistem, komponen data, dan *User Interface* dirancang dengan cermat [32]. Proses ini berfokus pada penciptaan landasan yang kuat untuk sistem perangkat lunak, memastikan ketahanan dan skalabilitasnya. OOD membangun informasi yang dikumpulkan pada tahap sebelumnya dan mengubahnya menjadi desain sistem yang komprehensif. Setelah tahap OOD, proses desain direalisasikan ke dalam kode program dengan menggunakan bahasa pemrograman yang berorientasi objek yang biasa disebut OOP. Pada pendekatan berorientasi objek, dapat memungkinkan *user* memilih metode pengujian yang akan dilakukan [33].

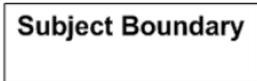
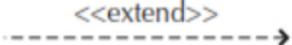
## **2.7 Unified Modeling Language (UML)**

*Unified Modeling Language* atau UML adalah sebuah bahasa/*model* yang biasa digunakan untuk memvisualisasikan, merancang, dan mendokumentasikan perancangan suatu sistem berdasarkan grafik/gambar dari sebuah sistem berbasis *Object-Oriented*. UML merupakan standar penulisan atau *blueprint* sebuah sistem yang akan dibuat, termasuk deskripsi konsep proses bisnis, skema basis data, dan penulisan kelas-kelas yang dibutuhkan dalam sebuah sistem [34] [35]. UML memvisualisasikan elemen-elemen dan hubungannya dalam bentuk diagram, yang membuatnya lebih mudah untuk melakukan pemahaman terhadap suatu sistem dari berbagai sudut pandang yang berbeda. Berikut diagram yang digunakan pada UML, antara lain:

### **2.7.1 Use Case Diagram**

*Use Case Diagram* adalah representasi grafis yang digunakan untuk menggambarkan interaksi antara aktor dan sistem dalam suatu lingkungan untuk menunjukkan aktivitas pada sistem dan hubungannya dengan aktor yang terlibat. Notasi *use case diagram* dijelaskan pada tabel 2.2 [36].

Tabel 2.2 Notasi *Use Case Diagram*

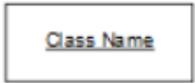
Notasi	Deskripsi
<p><i>Actor</i></p> 	Seseorang yang melakukan interaksi terhadap sistem untuk melakukan suatu tindakan (pertukaran informasi).
<p><i>Use Case</i></p> 	Sebuah rangkaian tindakan yang saling terkait, di mana setiap tindakan tersebut berhubungan dalam proses, baik secara manual maupun otomatis.
<p><i>Subject Boundary</i></p> 	Mewakili ruang lingkup sistem yang berisikan nama dari sistem yang batasannya ditentukan ( <i>boundary</i> )
<p><i>Association Relationship</i></p> 	Garis penghubung yang menunjukkan keterkaitan dua arah antara <i>actor</i> dan <i>use case</i>
<p><i>Include Relationship</i></p> 	Memasukkan satu <i>use case</i> dalam <i>use case</i> lainnya
<p><i>Extend Relationship</i></p> 	Memperluas <i>use case</i> untuk memasukkan perilaku opsional
<p><i>Generalization</i></p> 	Mewakili <i>use case</i> khusus untuk <i>use case</i> yang lebih umum

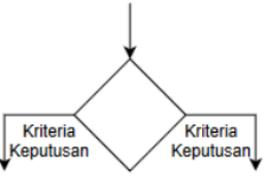
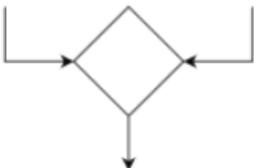
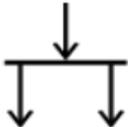
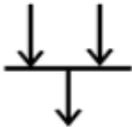
### 2.7.2 *Activity Diagram*

*Activity Diagram* merepresentasikan alur kerja (*workflow*) dari aktivitas kerja dalam suatu sistem [37]. Aktivitas dalam diagram ini membantu untuk memvisualisasikan kumpulan tindakan proses bisnis dan memahami kumpulan urutan aktivitas dalam sebuah proses tersebut [36]. *Activity diagram* dibuat untuk menggambarkan aliran berbagai aktivitas dalam sistem yang Anda rancang, dimana aktivitas tersebut dimulai, kemungkinan apa yang terjadi, bagaimana aktivitas tersebut berakhir, dan mendeskripsikan proses yang

dilakukan oleh beberapa proses yang dibuat berdasarkan satu atau lebih *use case* dalam suatu *use case diagram*. Berikut adalah notasi dari *activity diagram* dijelaskan pada tabel 2.3 [36]:

Tabel 2.3 Notasi *Activity Diagram*

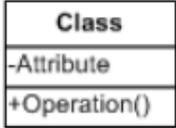
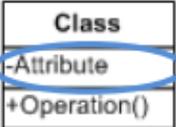
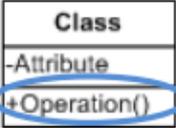
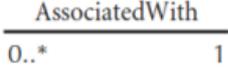
Notasi	Deskripsi
<p><i>Action</i></p> 	Perilaku yang sederhana dan tidak dapat dijelaskan
<p><i>Activity</i></p> 	Kumpulan tindakan atau <i>action</i>
<p><i>Object Node</i></p> 	Mewakili sebuah objek yang terkait dengan serangkaian objek lainnya
<p><i>Control Flow</i></p> 	Menunjukkan urutan eksekusi
<p><i>Object Flow</i></p> 	Menunjukkan perpindahan suatu objek dari satu aktivitas ke aktivitas lainnya
<p><i>Initial Node</i></p> 	Menggambarkan awal dari seluruh rangkaian tindakan atau kegiatan
<p><i>Final-activity Node</i></p> 	Menghentikan semua arus kontrol dan arus objek dalam suatu rangkaian aktivitas
<p><i>Final-flow Node</i></p> 	Menghentikan arus kontrol atau arus objek tertentu

Notasi	Deskripsi
<p data-bbox="448 331 614 360"><i>Decision Node</i></p> 	<p data-bbox="724 322 1350 416">Digunakan untuk mewakili kondisi pengujian guna memastikan bahwa arus kontrol atau arus objek hanya mengikuti satu jalur</p>
<p data-bbox="461 589 601 618"><i>Merge Node</i></p> 	<p data-bbox="724 580 1265 609">Menyatukan kembali berbagai jalur <i>decision node</i></p>
<p data-bbox="470 846 592 875"><i>Fork Node</i></p> 	<p data-bbox="724 837 1350 900">Digunakan untuk membagi perilaku menjadi serangkaian aktivitas yang paralel atau bersamaan</p>
<p data-bbox="474 1070 588 1099"><i>Join Node</i></p> 	<p data-bbox="724 1061 1350 1124">Menyatukan kembali serangkaian arus aktivitas yang paralel</p>
<p data-bbox="477 1294 585 1323"><i>Swimlane</i></p> 	<p data-bbox="724 1285 1350 1402">Membagi diagram menjadi baris dan kolom guna menetapkan kegiatan individu (atau tindakan) kepada individu atau objek yang bertanggung jawab melaksanakannya</p>

### 2.7.3 Class Diagram

*Class Diagram* memvisualisasikan struktur sistem dan kaitannya dengan definisi kelas-kelas yang akan dibuat untuk membangun sebuah sistem, yang ini termasuk atribut, operasi, dan hubungan antara kelas yang ada. Berikut adalah tabel 2.5 mengenai notasi yang terdapat pada *class diagram* [36].

Tabel 2.4 Notasi *Class Diagram*

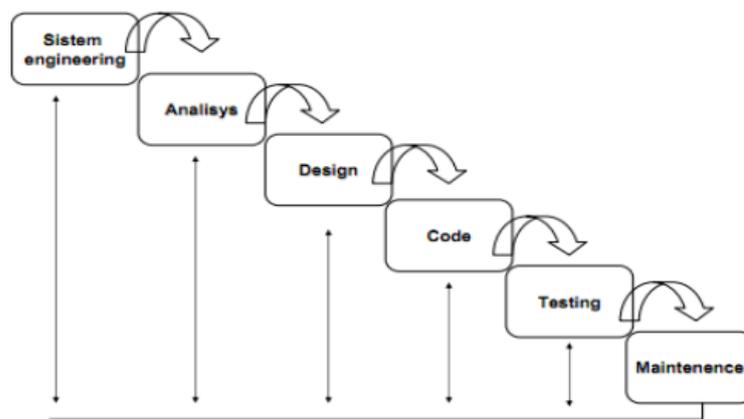
Notasi	Deskripsi
<p><i>Class</i></p> 	Mewakili orang, tempat, atau benda yang diperlukan sistem untuk menangkap dan menyimpan informasi
<p><i>Attribute</i></p> 	Menggambarkan keadaan suatu objek
<p><i>Operation</i></p> 	Tindakan atau fungsi yang dapat dilakukan oleh <i>class</i>
<p><i>Generalization</i></p> 	Mewakili hubungan sejenis antara beberapa kelas
<p><i>Association</i></p> 	Mewakili hubungan antara beberapa kelas atau antara kelas dengan dirinya sendiri
<p><i>Aggregation</i></p> 	Merupakan hubungan sebagian yang logis di antara beberapa kelas atau antara suatu kelas dengan kelas itu sendiri
<p><i>Composition</i></p> 	Menggambarkan koneksi fisik antara sejumlah kelas atau antara satu kelas dengan dirinya sendiri

## 2.8 System Development Life Cycle (SDLC)

*System Development Life Cycle* atau yang biasa disebut SDLC, merupakan serangkaian proses yang terstruktur dan terorganisir untuk mengembangkan perangkat lunak mulai dari perencanaan, analisis, desain, implementasi, pengujian, hingga pemeliharaan [38]. Ada beberapa jenis model dari SDLC, seperti model *waterfall*, *V shaped model*, *evolutionary prototyping model*, *spiral model*, *iterative*

and incremental model, dan agile model [39]. Dari banyaknya jenis SDLC tersebut, diperlukan pemilihan model SDLC yang tepat sesuai dengan permasalahan yang ada agar tujuan dari suatu proyek bisa tercapai.

*Waterfall model* adalah salah satu metodologi pengembangan *software* yang paling banyak digunakan karena sangat sederhana. Dalam model ini, setiap fase pengembangan harus selesai sebelum fase berikutnya dimulai, seperti air mengalir turun melalui tingkat yang berbeda dalam sebuah air terjun, dari atas ke bawah [29]. *Waterfall model* berbentuk sekuensial, dimana kemajuan terlihat mengalir terus ke bawah di mulai dari fase analisis, desain, implementasi, pengujian, serta *monitoring*. Tujuan dari penggunaan metode *waterfall* ini adalah agar kualitas sistem yang dihasilkan bisa maksimal, karena proses pengerjaannya yang dilakukan secara bertahap. Berikut merupakan gambaran proses pengembangan sistem dengan metode pendekatan SDLC *waterfall* [30]:



Gambar 2.1 SDLC Model *Waterfall*

Berdasarkan gambaran SDLC *waterfall* diatas, berikut uraian proses pengembangan sistem menggunakan pendekatan SDLC *waterfall*:

1. Analisis (*Analysis*): Pada tahap ini, kebutuhan sistem dipelajari dan dianalisis secara menyeluruh. Tujuannya adalah untuk memahami apa yang diinginkan oleh pengguna akhir.
2. Perancangan (*Design*): Setelah kebutuhan dikumpulkan, tim pengembangan merancang struktur sistem yang akan memenuhi kebutuhan tersebut. Ini

mencakup desain arsitektur sistem, desain *database*, desain *User Interface*, dan desain lainnya.

3. Implementasi (*Coding*): Pada tahap ini, kode perangkat lunak yang sebenarnya dibuat berdasarkan desain yang telah disetujui. Pengembang menulis kode berdasarkan spesifikasi yang telah ditentukan dalam tahap perancangan.
4. Pengujian (*Testing*): Setelah implementasi selesai, sistem diuji secara menyeluruh untuk memastikan kelayakan dan kelancaran sistem. Pengujian dapat mencakup pengujian fungsional, pengujian non-fungsional, dan pengujian integrasi.
5. Pemeliharaan (*Maintenance*): Setelah sistem diterapkan dan diuji, mungkin ditemukan kekurangan atau kebutuhan perbaikan. Tahap pemeliharaan melibatkan perbaikan *bug*, peningkatan performa, dan penyesuaian lainnya agar sistem tetap relevan dan berfungsi sebagaimana mestinya.

## 2.9 Scenario Testing

*Scenario testing* adalah metode pengujian perangkat lunak yang menggunakan skenario pengguna untuk memastikan semua fitur dalam sistem bekerja sesuai dengan yang diharapkan [40]. Metode ini sangat efektif karena mencerminkan situasi nyata yang mungkin dihadapi oleh pengguna, sehingga dapat mendeteksi masalah yang mungkin tidak terlihat dalam pengujian tradisional. Setiap fitur diuji melalui serangkaian langkah yang telah dirancang untuk memastikan bahwa semua fungsi berjalan dengan baik dan memberikan hasil yang diharapkan. Dengan demikian, *scenario testing* dapat membantu meningkatkan kualitas dan keandalan sistem. Berikut adalah langkah-langkah dalam *scenario testing* [41]:

1. Persiapan Skenario Uji:
  - Identifikasi fitur yang akan diuji.
  - Tentukan aktor yang terlibat dalam skenario.
  - Definisikan langkah-langkah uji secara rinci.
2. Pelaksanaan Uji:
  - Pastikan aktor telah terotorisasi sesuai dengan peran mereka.

- Ikuti langkah-langkah uji sesuai dengan skenario yang telah disusun.
  - Catat hasil yang diperoleh pada setiap langkah.
3. Evaluasi Hasil:
- Bandingkan hasil aktual dengan hasil yang diharapkan.
  - Tandai setiap langkah sebagai "PASS" jika hasil sesuai dengan harapan atau "FAIL" jika terjadi penyimpangan.
  - Buat catatan tambahan jika diperlukan.
4. Pelaporan:
- Buat laporan uji yang mencakup semua skenario yang diuji.
  - Sertakan hasil evaluasi dan catatan untuk setiap skenario.
  - Rekomendasikan tindakan perbaikan jika ditemukan masalah

