

BAB II

KAJIAN LITERATUR

2.1 Konsep Dasar Sistem Informasi

2.1.1 Sistem

Sistem merupakan rangkaian tindakan atau unsur yang berkaitan satu sama lain, di mana dalam suatu sistem terdapat *input*, proses, dan *output* yang bertujuan untuk mencapai tujuan yang diinginkan [2]. Sistem juga dapat diartikan sebagai bagian komponen yang secara fisik ataupun non fisik saling terhubung dan bekerja sama untuk sebuah tujuan yang sinkron. Sistem memiliki pendekatan dalam sebuah prosedur yang tahapannya secara berurutan mulai dari tindakan apa yang harus dilakukan (*what*), pelaku yang melaksanakan (*who*), waktu pelaksanaan (*when*) dan cara pelaksanaannya (*how*). Adapun karakteristik yang dimiliki pada suatu sistem yaitu [3] :

1. Komponen (*Component*)

Dalam sistem terdapat beberapa komponen yang saling berinteraksi dan membentuk suatu kesatuan yang terdiri dari beberapa cabang sistem. Komponen-komponen tersebut saling bekerja sama untuk mencapai tujuan sistem secara efektif dan efisien.

2. Lingkungan luar sistem (*Environment*)

Lingkungan luar sistem merupakan faktor yang mempengaruhi operasi sistem sehingga harus dijaga dan dipelihara agar tidak mengganggu keberlangsungan hidup sistem tersebut.

3. Batasan sistem (*Boundary*)

Lingkungan luar dari suatu sistem dibatasi oleh ruang lingkup sistem itu sendiri dan dibatasi oleh batas sistem lain yang sesuai dengan daerahnya. Batasan sistem tersebut dapat berbentuk lingkaran atau bidang yang jelas dan dapat diidentifikasi.

4. Penghubung sistem (*Interface*)

Penghubung sistem merupakan alat bantu yang menghubungkan satu sistem dengan sub sistem lainnya. Dengan menggunakan penghubung sistem, sumber daya dapat mengalir dari suatu sub sistem ke sub sistem lainnya.

5. Masukkan sistem (*Input*)

Input sistem merujuk pada sumber daya yang dimasukkan ke dalam sistem, baik itu berupa pemeliharaan atau sinyal masuk. Pemeliharaan adalah jenis *input* sistem yang dibutuhkan agar sistem dapat berfungsi secara optimal, sedangkan sinyal masukan merupakan *input* sistem yang diproses untuk menghasilkan keluaran.

6. Keluaran sistem (*Output*)

Keluaran sistem merupakan hasil dari pemrosesan *input* yang diterima oleh sistem dan mengeluarkan sebuah energi atau informasi. Sebagai contoh, ketika sebuah sistem telah mengeluarkan energi yang diolah dan menghasilkan informasi yang dianggap sebagai keluaran energi yang berguna.

7. Pengelolaan sistem (*Process*)

Pengelolaan sistem adalah tahapan dalam proses di mana *input* diubah menjadi *output*. Salah satu contoh pengolahan sistem adalah akuntansi yang mengelola data keuangan dan menghasilkan laporan-laporan keuangan sebagai *output*.

8. Sasaran sistem

Sasaran sistem merupakan tujuan dan sasaran yang telah direncanakan, apabila sistem tidak memiliki sasaran maka sistem tersebut tidak berguna dan tidak membantu dalam menyelesaikan tugas.

2.1.2 Informasi

Informasi merupakan hasil pengolahan data yang telah di olah untuk mendapatkan pemahaman yang lebih bermanfaat dalam mencapai tujuan tertentu. Informasi dianggap sangat penting karena memberikan manfaat lebih dari pada sekedar melihat data yang ada [2]. Sumber informasi berasal dari data yang merepresentasikan kejadian yang telah terjadi pada waktu tertentu. Untuk menghasilkan informasi yang bermanfaat melewati sebuah siklus yang disebut siklus informasi (*infotmation cycle*), pada siklus ini data yang belum diolah dapat disimpan dalam bentuk basis data. Data yang disimpan nantinya dapat diambil kembali saat data akan diolah menjadi informasi. Data tersebut akan menjadi *input* dalam proses pengolahan data dengan menggunakan suatu model tertentu, sehingga menghasilkan *output* yang dapat digunakan oleh penerima untuk membuat keputusan dan melakukan tindakan [3].

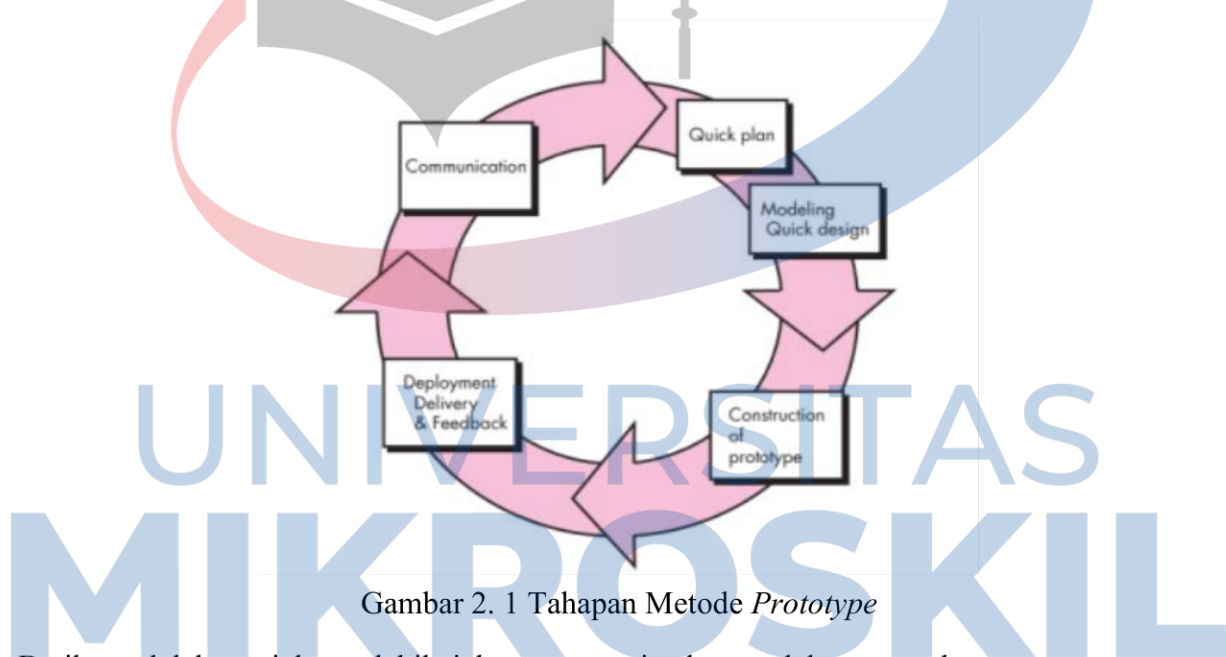
2.1.3 Sistem Informasi

Sistem informasi merupakan integrasi dari komponen-komponen yang telah di analisa dan diproses sehingga menampilkan informasi yang diperlukan untuk membantu pengguna dalam mengambil suatu keputusan [2]. Sistem informasi sering digunakan dalam suatu organisasi yang terdiri dari orang, fasilitas, teknologi, media, prosedur dan pengendalian untuk memperoleh informasi, mengelola transaksi, pengambilan keputusan dan lainnya terkait kepentingan internal ataupun eksternal [4]. Kini sudah banyak perusahaan menjadikan sistem informasi sebagai alat bantu kinerja bisnis yang lebih efektif seperti halnya dalam mengelola suatu transaksi yang awalnya dilakukan secara manual menjadi *digital* sehingga menghasilkan

laporan keuangan yang akurat [5]. Dengan bantuan sistem informasi dapat mengoptimalkan proses bisnis untuk mencapai tujuan perusahaan.

2.2 Metode *Prototype*

Metode *prototype* merupakan cara untuk mengembangkan produk atau sistem dengan membuat model awal yang dapat diuji dan dievaluasi. Tujuan utama metode ini adalah untuk memperoleh umpan balik awal dari pengguna dan tim pengembang terhadap fitur, fungsi, dan desain produk sebelum produk akhir dibuat. Penggunaan metode *prototype* memberikan manfaat bagi *client* untuk mendapatkan sistem yang lebih baik karena *client* dapat selalu berinteraksi sampai akhir perancangan untuk menyesuaikan kebutuhan sistem [6]. *Prototype* bukanlah sesuatu yang lengkap, tetapi sesuatu yang harus di evaluasi dan di modifikasi kembali. Menurut Pressman pada penelitian [6] metode *prototype* memiliki beberapa tahapan pengembangan sistem yang dimulai dari proses komunikasi dengan pengguna.



Gambar 2. 1 Tahapan Metode *Prototype*

Berikut adalah penjelasan lebih jelas mengenai tahapan dalam metode *prototype* menurut Pressman 2010 [6].

1. *Communication* atau komunikasi

Tahap ini dilakukan untuk mengidentifikasi kebutuhan dan mengumpulkan data untuk merancang aplikasi yang akan dibuat dengan melibatkan pengguna terkait. Tujuannya adalah agar selama proses perancangan, hasil yang dihasilkan sesuai dengan harapan dan kebutuhan dari klien terkait.

2. *Quick plan* atau perencanaan kebutuhan

Tahap ini dilakukan perencanaan kebutuhan dengan merujuk pada spesifikasi kebutuhan pengguna yang telah dikumpulkan melalui proses komunikasi sebelumnya. Dalam langkah ini fokusnya adalah merancang kebutuhan sistem yang sesuai dengan kebutuhan yang teridentifikasi serta menentukan elemen pendukung yang diperlukan dalam proses tersebut yang dapat digambarkan menggunakan pemodelan *Unified Modelling Language* (UML).

3. *Modelling Quick Design* atau pemodelan desain

Tahap ini peneliti akan merancang model desain dengan menggunakan *software* perancangan seperti adobe XD atau Figma . Tujuannya adalah untuk secara efisien menciptakan representasi visual yang jelas tentang kebutuhan pengguna berdasarkan analisis sebelumnya. Proses ini akan memastikan bahwa semua elemen yang diperlukan dipertimbangkan dan diwakili secara tepat sebelum melangkah ke tahap implementasi.

4. Pembentukan *prototype*

Tahap ini pembuatan perangkat *prototype* termasuk pengujian dan penyempurnaan. Dalam tahap ini perancang akan membangun *prototype* perangkat lunak berdasarkan data yang telah dikumpulkan sebelumnya. Proses ini akan lebih difokuskan pada aspek inti dari perangkat lunak dengan tujuan agar perancang dapat segera mendapatkan umpan balik dari klien mengenai produk yang sedang dibuat.

5. *Deployment Delivery & Feedback*

Tahapan mengevaluasi *prototype* dan memperhalus analisis terhadap kebutuhan pengguna. Perbaikan *prototype*, yaitu pembuatan tipe yang sebenarnya berdasarkan hasil dari evaluasi *prototype* dan selanjutnya produksi akhir yaitu memproduksi perangkat secara benar sehingga dapat digunakan oleh pengguna.

2.3 Penjualan

Penjualan dapat didefinisikan sebagai aktivitas manusia yang ditujukan untuk memenuhi dan memuaskan kebutuhan serta keinginan melalui proses pertukaran. Proses penjualan juga merupakan upaya mencari pembeli, mempengaruhi mereka, dan memberikan panduan agar pembeli dapat menyelaraskan kebutuhan mereka dengan produk yang ditawarkan. Selain itu, tujuan dari kegiatan penjualan adalah untuk mencapai kesepakatan atau perjanjian mengenai harga yang menguntungkan baik bagi penjual maupun pembeli. Dengan demikian penjualan bukan hanya sekadar transaksi jual-beli, tetapi juga melibatkan upaya menciptakan hubungan yang saling menguntungkan dan memastikan kecocokan antara penawaran produk dan kebutuhan konsumen [7].

Penjualan atau yang dikenal dengan istilah pemasaran merupakan suatu kegiatan atau aspek bisnis yang terkait dengan penawaran dan pembelian produk atau jasa. Definisi umum dari penjualan adalah kegiatan jual-beli yang dilakukan oleh dua pihak atau lebih, melibatkan alat pembayaran yang sah. Dengan kata lain, penjualan melibatkan proses interaksi antara penjual dan pembeli yang bertujuan untuk menukar produk atau jasa dengan pembayaran yang telah ditentukan. Aktivitas ini merupakan inti dari kegiatan bisnis yang melibatkan transaksi komersial dan berperan penting dalam ekosistem ekonomi [8].

Kini metode penjualan semakin berkembang yang di mana proses jual-beli dilakukan melalui *platform digital* tanpa bertemu secara langsung. Dalam konteks ini, transaksi dilakukan secara virtual melalui sebuah media atau alat yang menghubungkan antara penjual yang menawarkan produk kepada calon pembeli. Dengan kata lain, penjualan *online* memungkinkan pembeli untuk memperoleh informasi dan melakukan transaksi pembelian secara elektronik melalui internet tanpa harus berinteraksi secara fisik dengan penjual [9].

Penjualan *online* merupakan kegiatan perdagangan yang melibatkan proses mulai dari mencari calon pembeli hingga mempromosikan produk atau barang menggunakan fasilitas internet dan perangkat elektronik sebagai sarana untuk terhubung dengan jaringan *online*. Fenomena penjualan *online* telah menjadi semakin populer di masyarakat, bahkan menjadi preferensi utama karena berbagai keuntungan yang ditawarkan, seperti diskon besar-besaran dan penghematan waktu [10].

2.4 Website

Website adalah sebutan bagi sekelompok halaman *webpage* yang umumnya merupakan bagian dari suatu nama domain atau sub domain yang terkoneksi di internet. *Website* yang bersifat statis maupun dinamis memiliki 1 rangkaian bangunan yang saling terkait dan dihubungkan dengan jaringan atau disebut dengan *hyperlink*. Bersifat statis apabila isi informasi tetap, jarang berubah dan informasinya searah hanya dari pemilik *website*. Bersifat dinamis apabila isi informasi *website* selalu berubah-ubah. Situs *website* yang memiliki fungsi informasi pada umumnya lebih menekankan pada kualitas bagian kontennya, karena tujuan situs tersebut adalah menyampaikan isinya

Berdasarkan pengertian di atas maka disimpulkan bahwa *Website* adalah sekelompok halaman yang saling terkait dalam sebuah domain dan dapat diakses melalui internet. Setiap halaman dalam *website* umumnya berisi konten seperti teks, gambar, video, dan elemen lainnya yang bertujuan untuk menyampaikan informasi sesuai kebutuhan pengguna [11].

2.5 Metode Antrean First Come First Served

Metode *First Come First Served* adalah metode untuk mengurutkan pemesanan yang pertama kali masuk untuk dilayani terlebih dahulu. Proses pengurutan terjadi ketika pelanggan melakukan pemesanan melalui sistem maka informasi pemesanan akan masuk ke data antrean. Setelah itu sistem akan mengurutkan pesanan yang masuk pertama kali terletak diposisi antrean yang paling atas. Posisi antrean yang paling teratas akan diproses terlebih dahulu kemudian dilanjut ke antrean selanjutnya sehingga setiap pelanggan dapat terlayani dengan teratur dalam penerimaan pesanan [12].

2.6 Teknik Perancangan


2.6.1 Unified Modelling Language (UML)




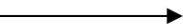
Unified Modelling Language (UML) merupakan teknik pemodelan yang menggunakan konsep berorientasi objek untuk spesifikasi, visualisasi, membangun dan mendokumentasikan dari sistem perangkat lunak [13]. Perancangan dan pengembangan aplikasi atau perangkat lunak berbasis objek yang dimodelkan menggunakan UML. Tujuan UML dapat memberikan gambaran model yang jelas dan mempermudah *developer* untuk mengetahui alur atau *flow* sistem yang dibutuhkan *user*. UML terdiri dari 4 jenis diagram yang sering digunakan yaitu *use case diagram*, *class diagram*, *activity diagram* dan *sequence diagram* [14].

1. *Use case diagram*

Use case merupakan diagram yang menggambarkan suatu tindakan yang dilakukan aktor terhadap sistem yang akan dikembangkan. *Use case* memberikan pandangan yang jelas tentang apa yang seharusnya dilakukan pengguna terhadap sistem dan siapa saja yang berhak menggunakan fungsi *use case* tersebut. Setiap *Use case* mendeskripsikan tiga hal utama yaitu aktor yang memulai suatu kejadian, kejadian yang memicu *use case*, dan tindakan yang dilakukan oleh sistem dalam menjalankan *Use case* tersebut [15].

Tabel 2. 1 Simbol *Use Case Diagram*

Simbol	Keterangan
	<i>Use case</i> digunakan untuk menyatakan fungsi-fungsi atau proses-proses yang disediakan pada aplikasi sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

 Actor	<i>Actor</i> , proses atau aplikasi lain yang berinteraksi dengan aplikasi yang akan dibuat itu sendiri. Jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu orang.
Association 	Komunikasi antara <i>actor</i> dan <i>Use case</i> , dimana <i>Use case</i> memiliki interaksi dengan aktor.
<< extend >> 	Relasi <i>Use case</i> tambahan ke sebuah <i>use case</i> , dimana <i>Use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>Use case</i> tambahan.
<<include>> 	Relasi <i>Use case</i> tambahan ke <i>use case</i> , merupakan syarat dijalankan <i>Use case</i> tersebut, dimana <i>Use case</i> yang ditambahkan memerlukan <i>Use case</i> lain untuk menjalankan fungsinya.

a. *Use Case Narrative*

Use case narrative yaitu uraian deskripsi dari *use case* diagram sehingga pengguna UML bisa mengetahui detail dari proses yang ada pada *use case* diagram. Berikut elemen-elemen yang digunakan pada *use case narrative* [16].

Tabel 2. 2 Contoh *Use Case Narrative*







Nama Use Case	Pesanan	
Aktor	Admin	
Pre-Condition	Admin menambah menu dalam sistem	
Post Condition	Menu berhasil ditambah	
Skenario Normal		
	Aksi Aktor	Reaksi Sistem
	1. Admin memilih menu makanan	2. Sistem menampilkan menu makanan dalam sistem
	3. Admin tambah menu	4. Sistem menampilkan <i>form</i> menu
	5. Admin mengisi <i>form</i>	6. Sistem menyimpan data menu

2. *Activity diagram*

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas suatu sistem atau proses bisnis atau menu yang terdapat pada perangkat lunak. Perlu diperhatikan bahwa

diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor melainkan aktivitas yang dapat dilakukan sistem [15].

Tabel 2. 3 Simbol *Activity Diagram*

Nama Komponen	Simbol	Keterangan
<i>Initial</i>		Titik awal untuk memulai suatu aktivitas.
<i>Final</i>		Titik akhir untuk mengakhiri aktivitas.
<i>Activity</i>		Menandakan suatu aktivitas.
<i>Decision</i>		Pilihan untuk mengambil keputusan.
<i>Fork atau Join</i>		Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
<i>Swimlane</i>		Untuk mengelompokkan <i>activity</i> berdasarkan aktor.

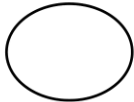


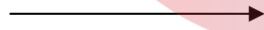


3. *Class diagram*

Merupakan sebuah diagram yang digunakan untuk menampilkan kelas-kelas pada suatu sistem yang akan digunakan. Diagram ini dapat memberikan gambaran struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem [15]. *Class diagram* mempunyai 3 relasi yang digunakan yaitu [17] :

1. *Association* adalah hubungan yang menunjukkan adanya interaksi antar *class*.
2. *Generalization* adalah hubungan antar *class* yang bersifat khusus ke umum.
3. *Constraint* adalah hubungan yang memberikan batasan sistem sehingga terdapat aspek yang tidak fungsional.

Tabel 2. 4 Simbol *Class Diagram*

Simbol	Keterangan
--------	------------




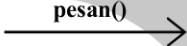


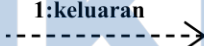
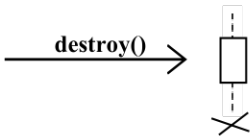
<p>Kelas/<i>class</i></p> <table border="1"> <tr> <td>Nama kelas</td> </tr> <tr> <td>-Atribut</td> </tr> <tr> <td>+Operasi</td> </tr> </table>	Nama kelas	-Atribut	+Operasi	<p>Kelas menggambarkan dari struktur sistem yang akan ditampilkan dalam sistem.</p> <p>Atribut menggambarkan keadaan dari suatu objek didalam kelas</p> <p>Oprasi menggambarkan fungsi yang terdapat dalam kelas.</p>
Nama kelas				
-Atribut				
+Operasi				
<p>Antar muka/<i>interface</i></p> 	<p>Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.</p>			
<p>Asosiasi/<i>Assosiation</i></p> 	<p>Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplecities</i>.</p>			
<p>Asosiasi berarah/<i>directed association</i></p> 	<p>Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>.</p>			
<p>Generalisasi</p> 	<p>Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)</p>			
<p>Agregasi/<i>aggregation</i></p> 	<p>Relasi antar kelas dengan makna semua bagian.</p>			
<p>Kebergantungan/<i>dependency</i></p> 	<p>Relasi antar kelas dengan makna ketergantungan antar kelas.</p>			

4. *Sequence diagram*

Merupakan diagram yang menggambarkan interaksi antara kelas atau proses objek dari waktu ke waktu. Diagram ini sering digunakan dalam analisis kasus penggunaan dan juga dalam desain sistem untuk mendapatkan pemahaman yang lebih baik tentang interaksi, hubungan, dan metode yang digunakan oleh objek-objek dalam sistem [15].

Tabel 2. 5 Simbol *Sequence Diagram*

Simbol	Keterangan
Aktor	Orang yang berinteraksi dengan sistem.

	
<p>Objek</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> Nama objek : nama jelas </div>	<p>Mewakili aktor, kelas, objek, atau instance yang terlibat dalam interaksi. Aktor atau objek awal ditempatkan di sebelah kiri dan interaksi berlanjut ke kanan.</p>
<p>Garis hidup/<i>lifeline</i></p> 	<p>Menyatakan objek yang berinteraksi pesan dan diakhiri dengan tanda X pada titik dimana kelas tidak berinteraksi.</p>
<p>Waktu aktif</p> 	<p>Menunjukkan fokus kontrol saat objek sedang aktif dan berinteraksi dalam suatu tahapan.</p>
<p>Pesan tipe <i>call</i></p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, arah panah mengarah pada objek yang memiliki operasi/metode.</p>
<p>Pesan tipe <i>create</i></p> 	<p>Menyatakan suatu objek membuat objek lain, arah panah mengarah pada objek yang dibuat.</p>
<p>Pesan tipe <i>send</i></p> 	<p>Menyatakan bahwa suatu objek mengirimkan data ke objek lainnya, arah panah mengarah pada objek yang dikirimkan.</p>
<p>Pesan tipe <i>return</i></p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>
<p>Pesan tipe <i>destroy</i></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>.</p>