

BAB II **LANDASAN TEORI**

2.1 Tinjauan Pustaka

Berikut adalah beberapa tinjauan pustaka yang digunakan sebagai referensi atau pembandingan terhadap penelitian ini :

1. Oleh Nur Fatimah, Yandria Elmasari (2018) dari Program Studi Pendidikan Teknologi Informasi, STKIP PGRI Tulung Agung dengan judul Perancangan Sistem Informasi Perpustakaan Berbasis Web Untuk SMA Islam Sunan Gunung Jati. Penelitian ini mengangkat masalah pengolahan data pada perpustakaan SMA Islam Sunan Gunung Jati, yang masih menggunakan cara manual dimana perpustakaan tersebut masih tergolong baru dan belum banyak buku. Yang semua prosesnya dilakukan secara manual, seperti halnya pencatatan proses peminjaman, pengembalian dan pendataan. Hal tersebut tidak efisien, karena boros kertas, data tidak akurat, dan pengarsipan menjadi berantakan. Penelitian ini bertujuan untuk membuat perancangan sistem informasi perpustakaan berbasis web. Sistem ini dibangun menggunakan metode penelitian *extreme programming*, dengan menggunakan bahasa pemrograman *PHP (Hypertext Preprocessor)*, *Javascript*, dan *CSS Cascading Style Sheet*). Sedangkan basis datanya menggunakan *MySQL* yang di akses menggunakan *XAMPP*. Berdasarkan hasil penelitian, sistem informasi perpustakaan berbasis web yang dikembangkan dalam penelitian ini layak digunakan dengan hasil penhujian *usability* dengan hasil 84,22% (kriteria sangat layak) pada kelompok kecil, dan 91,2% (kriteria sangat layak) pada kelompok besar, serta *performance effeciency* dengan hasil 98,64 (grade A).
2. Oleh Indra Chaidir, Dimas Wahyu Aditya, Sumarna (2020) dari Program Studi Sistem Informasi, Universitas Bina Sarana dan Program Studi Teknik Informatika, STIMK Nusa Mandiri dengan judul Rancang Bangun Sistem Informasi Berbasis Web Pada MTS Al -

Husna Depok. Penelitian ini mengangkat masalah pada sistem yang masih menggunakan cara manual dalam bentuk buku untuk melakukan pendataan baik mencari data buku, anggota, transaksi peminjaman dan pengembalian buku. Cara itu sangat tidak efektif mengingat banyak kasus kehilangan buku pada perpustakaan MTS Al - Husna Depok. Hal tersebut cukup membuat petugas perpustakaan kesulitan dalam melakukan pendataan. Penelitian ini bertujuan untuk merancang sistem informasi perpustakaan berbasis web pada MTS Al - Husna Depok, guna menjadikan sistem agar lebih terkomputerisasi. Metode yang digunakan adalah deskriptif, dengan teknik pengumpulan data menggunakan observasi, wawancara, dan studi pustaka. Dengan metode pengembangan sistem *waterfall*, *UML (Unified Modelling Language)* dan *ERD (Entity Relationship Diagram)*. Bahasa pemrograman yang digunakan adalah *HTML*, *SQL*, *CSS*, dan *PHP*. Hasil dari penelitian ini adalah petugas perpustakaan dapat melakukan pelayanan lebih cepat, akurat, aman, dan lebih efisien. Serta dapat meminimalisir kesalahan – kesalahan yang terjadi dalam pengolahan data anggota, data buku serta laporan.

3. Oleh Aji Permana, M.Kom (2018) dari Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Kuningan dengan judul Rancang Bangun Sistem Informasi Perpustakaan Berbasis Web (Studi Kasus Universitas Kuningan). Penelitian ini mengangkat masalah data buku yang ada di masing masing perpustakaan tidak dapat terkontrol, karena alur bisnis pengelolaan yang berjalan dilakukan secara terpisah dan tidak terpusat. Hal itu berakibat pada proses registrasi dan peminjaman buku masih diperbolehkan jika mahasiswa tersebut sudah meminjam pada perpustakaan fakultas, yang akan menjadi tanggung jawab pengelola buku untuk menjamin buku dikembalikan beserta denda yang telah ditentukan. Penelitian ini bertujuan untuk merancang sistem informasi perpustakaan yang mencakup data buku, transaksi peminjaman dan pengembalian buku, data peminjam dan berbagai proses yang saling berhubungan antara

Universitas dengan data perpustakaan fakultas. Metode yang digunakan adalah Web Server Apache sebagai media interaksi antara server dengan klien lewat browser, *MySQL* untuk menampung data pada server, *PHP (Personal Home Page)* yang digunakan untuk membuat aplikasi, *Microsoft Visio* yang digunakan untuk membuat *flowchart*, *brainstorm*, dan skema jaringan. Hasil dari penelitian ini adalah laporan data pengunjung, registrasi, peminjaman, pengembalian serta denda yang dapat di sesuaikan.

4. Oleh Syamsul Bakhri, Al Bani (2019) dari Program Studi Teknologi Komputer, Universitas Bina Sarana Informatika dan Program Studi Sistem Informasi, STMIK Nusa Mandiri dengan judul Sistem Informasi Perpustakaan Berbasis Web Pada SMAN 1 Cibinong. Penelitian ini mengangkat masalah sistem pada perpustakaan yang masih dilakukan melalui lembar formulir mengakibatkan membutuhkan waktu yang lama dalam berbagai prosesnya baik dalam pencarian data buku, anggota, denda, transaksi peminjaman dan pengembalian buku. Hal itu berakibat pada proses laporan bulanan. Penelitian ini bertujuan untuk membangun sistem informasi perpustakaan berbasis website guna meningkatkan pelayanan. Metode yang digunakan dalam proses pengumpulan data adalah observasi, wawancara, studi pustaka dan *tools UML* sebagai pendokumentasian sistem. Hasil dari penelitian ini adalah sistem informasi perpustakaan berbasis website yang dirancang untuk dapat membantu memudahkan penyimpanan data dan mengurangi terjadinya kesalahan.
5. Oleh Deanna Durbin Hutagalung , Feni Arif (2018) dari Program Studi Teknik Informatika, Universitas Pamulang dengan judul Rancang Bangun Sistem Informasi Perpustakaan Berbasis Web Pada SMK Citra Negara Depok. Penelitian ini mengangkat permasalahan pada perpustakaan online yang rentan terjadi kesalahan dalam proses peminjaman buku yang dikarenakan adanya data yang belum terdokumentasi dengan baik. Penelitian ini bertujuan untuk merancang bangun sistem informasi perpustakaan

berbasis website agar dapat mengatasi permasalahan yang ada dengan pengelolaan data yang terstruktur agar peminjaman buku dapat terdokumentasi dengan baik. Metode yang dilakukan adalah *PHP*, *MySQL*, *UML*, dan *Bootstrap*. Hasil dari penelitian ini adalah pengolahan data menggunakan sistem yang telah terkomputerisasi dapat dilakukan dengan cepat dan mudah, serta mengurangi kesalahan laporan pengolahan.

Berdasarkan tinjauan pustaka ke-lima jurnal diatas, terdapat perbedaan penelitian ini dengan penelitian sebelumnya, yaitu terletak pada objek penelitian yang berada di MAN 2 Bandar Lampung. Dari beberapa jurnal yang telah disebutkan diatas, penelitian sebelumnya tidak ada yang menggunakan metode pengembangan sistem menggunakan *prototyping model*, dan sistem informasi yang akan peneliti sediakan yaitu data koleksi buku perpustakaan, transaksi peminjaman dan pengembalian buku, data pengunjung perpustakaan, laporan peminjaman dan pengembalian buku. Kelebihan dari sistem yang akan dibuat oleh penulis adalah petugas perpustakaan tidak lagi menginputkan buku secara tertulis kedalam buku besar, melainkan semua proses dilakukan dengan sistem yang terkomputerisasi. Dan dalam pengujian sistem ini menggunakan ISO 25010.

2.2 Pengertian Sistem Informasi

Sistem merupakan suatu kesatuan yang terdiri dari komponen yang saling berhubungan yang digunakan untuk memudahkan aliran informasi, materi atau energi. Informasi adalah suatu kumpulan data yang di analisis guna mencapai tujuan tertentu dengan proses pengumpulan dan penyimpanan data yang terdiri dari *input* untuk menghasilkan *output* sesuai dengan kebutuhan. Sistem informasi merupakan suatu kesatuan yang terdiri dari *hardware*, *software*, *brainware*, prosedur, yang saling berhubungan untuk mengolah data menjadi

informasi yang bermanfaat untuk mengatasi permasalahan dan mengambil keputusan. (Pramestyarani, 2020)

Berdasarkan pengertian di atas, penulis menyimpulkan bahwa sistem informasi adalah sebuah kumpulan data yang saling terintegrasi dan saling melengkapi dengan menghasilkan *output* untuk memecahkan suatu persoalan dalam hal pengambilan keputusan.

2.3 Perpustakaan

Perpustakaan adalah suatu lembaga pengelolaan berbagai bahan pustaka, baik berupa buku maupun media rekaman yang digunakan sebagai wahana pendidikan, penelitian dan pelestarian informasi yang disusun secara sistematis dengan aturan tertentu agar dapat digunakan sebagai sumber pengetahuan oleh setiap penikmatnya. (Aminullah et al., 2021)

Berdasarkan pengertian diatas, penulis menyimpulkan bahwa perpustakaan adalah sarana yang disiapkan oleh sekolah dan didalamnya terdapat koleksi buku yang dapat menambah perolehan wawasan selain dari kegiatan belajar mengajar di ruang sekolah.

2.4 Sistem Informasi Perpustakaan

Sistem informasi perpustakaan adalah suatu sistem yang digunakan untuk mengelola data yang ada pada perpustakaan. Sistem informasi perpustakaan merupakan *software* yang dapat memenuhi semua pelayanan umum pada perpustakaan meliputi peminjaman buku dan lain sebagainya. Sehingga petugas perpustakaan dapat mengelola manajemen perpustakaan tersebut dengan baik dari segi waktu, biaya, maupun usaha. (Aini et al., 2019)

Berdasarkan pengertian diatas, penulis menyimpulkan bahwa sistem informasi perpustakaan adalah suatu *software* yang dapat melayani pengunjung perpustakaan tanpa datang ke perpustakaan tersebut secara langsung. Didalam sistem informasi perpustakaan,

terdapat beberapa pelayanan yaitu meliputi peminjaman buku, pengembalian buku, *explore* koleksi buku, dan beberapa pelayanan perpustakaan secara online lainnya.

2.5 Website

Website adalah kumpulan beberapa halaman yang saling terhubung dalam sebuah domain atau subdomain. Memuat media berupa teks, dokumen, gambar, animasi dengan bantuan *web browser*. Pada saat ini, *web* bukanlah hal yang baru, karena perkembangan teknologi semakin meningkat dengan banyaknya perusahaan dan instansi yang menggunakan *gadget* membuat banyak orang tertarik untuk mengembangkan sistem informasi maupun aplikasi yang berbasis *web* yang tempatnya berada di dalam *World Wide Web* (WWW) di internet. (Fatimah & Elmasari, 2018)

Berdasarkan pengertian diatas, penulis menyimpulkan bahwa *website* adalah suatu kumpulan halaman yang digunakan untuk menampilkan informasi teks atau gambar sebagai media promosi atau pemasaran.

2.6 CodeIgniter

CodeIgniter adalah suatu aplikasi *open source* yang berisi kerangka kerja atau *framework* dan berfungsi untuk membangun website dengan bahasa pemrograman PHP. Bertujuan menguatkan pengembangan proyek agar lebih cepat dalam penulisan kode dasar atau kode terstruktur. *CodeIgniter* merupakan *framework* PHP yang dibuat sesuai dengan model *view controller* (MVC). *CodeIgniter* mempunyai *library* lengkap yang dapat mengerjakan operasi-operasi yang sering dibutuhkan oleh aplikasi berbasis *web* untuk mengakses *database* serta memvalidasi *form* sehingga sistem yang dikembangkan mudah. *CodeIgniter* memudahkan *developer* dalam pembangunan sistem karena *framework* tersebut sudah memiliki kerangka kerja sehingga tidak perlu menulis semua program dari awal. (Sallaby & Kanedi, 2020)

Berdasarkan pengertian diatas, penulis dapat menyimpulkan bahwa *CodeIgniter* adalah aplikasi *open source* yang didalamnya terdapat kerangka kerja dan berfungsi untuk membangun *website* tanpa mengetikkan kode program dari awal.

2.7 PHP (*Personal Home Page*)

PHP adalah bahasa pemrograman yang berfungsi untuk membuat website dinamis dan interaktif. Dinamis merupakan website yang dapat berubah-ubah tampilan maupun kontennya sesuai dengan kondisi tertentu. (Gregorius Agung P, 2018) PHP merupakan singkatan dari *Hypertext Preprocessor* adalah bahasa pemrograman berbasis web yang di yang dibuat untuk pengembangan web dan digunakan sebagai penerjemah kode program menjadi kode mesin yang mudah dimengerti oleh komputer berbasis *server-side* dan dapat ditambahkan ke dalam HTML. (Sudaria et al., 2021)

Berdasarkan pengertian diatas, penulis menyimpulkan bahwa PHP adalah bahasa pemrograman yang digunakan untuk membuat website dinamis yang fleksibel karena dapat disesuaikan dengan kondisi tertentu. PHP adalah singkatan dari *Hypertext Preprocessor* yang merupakan penerjemah kode program menjadi kode mesin yang mudah dimengerti oleh komputer.

2.8 MySQL

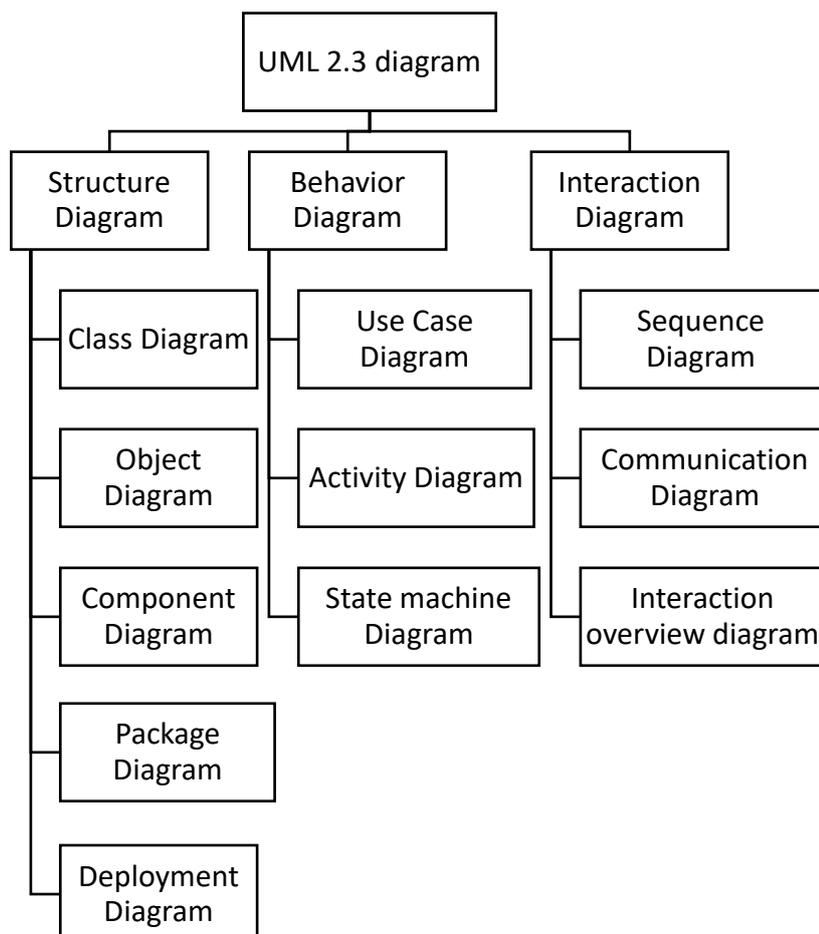
MySQL adalah server yang digunakan untuk melayani database. Database sendiri digunakan untuk menginputkan data dari user menggunakan form HTML dan selanjutnya diolah PHP agar bisa disimpan ke dalam database MySQL. (Gregorius Agung P, 2018) MySQL adalah *mutiuser database* yang menggunakan bahasa *Structural Query Language (SQL)*. Yang merupakan manajemen *database SQL* yang memiliki sifat *open source* paling sering digunakan saat ini. (Sudaria et al., 2021)

Berdasarkan pengertian diatas, penulis menyimpulkan bahwa MySQL adalah aplikasi yang digunakan untuk membuat database yang berisi kelas-kelas untuk membuat beberapa kelas diagram. MySQL merupakan singkatan dari *Structural Query Language*.

2.9 UML (*Unified Modelling Language*)

Unified Modelling Language (UML) adalah bahasa permodelan untuk pembangunan perangkat lunak dengan menggunakan teknik pemrograman berorientasi objek. Dalam diagram UML, terdapat 13 macam diagram yang dikategorikan menjadi 3 kategori. Berikut penjelasan singkat dari pembagian kategori tersebut :

- a. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
- b. *Behavior diagrams* yaitu kumpulan diagram yang digunakan unruk menggambarkan rangkaian perubahan yang terjadi pada sistem.
- c. *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi suatu sistem dengan sistem lain. (A.S Rosa & Shalahuddin.M, 2019)



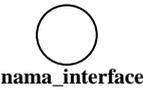
Gambar 2. 1 UML (*Unified Modelling Language*)

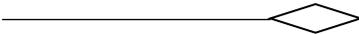
Sumber : (Rosa A.S and Shalahuddin, 2019)

2.10 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut atribut dan metode atau operasi. . (Rosa A.S and Shalahuddin, 2019). Simbol-simbol yang ada pada diagram kelas dapat dilihat pada tabel 2.1 di bawah ini :

Tabel 2. 1 *Class Diagram*

Simbol	Deskripsi			
<p>Kelas</p> <table border="1" style="margin-left: 20px;"> <tr> <td>nama_kelas</td> </tr> <tr> <td>+atribut</td> </tr> <tr> <td>+operasi()</td> </tr> </table>	nama_kelas	+atribut	+operasi()	Kelas pada struktur sistem
nama_kelas				
+atribut				
+operasi()				
<p>Antarmuka / <i>interface</i></p> <div style="text-align: center;">  </div>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.			
<p>Asosiasi/<i>association</i></p> <div style="text-align: center;">  </div>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .			
<p>Asosiasi berarah / <i>directed association</i></p> <div style="text-align: center;">  </div>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiais biasanya juga disertai dengan <i>multiplicity</i> .			
<p>Generalisasi</p> <div style="text-align: center;">  </div>	Relasi antar kelas dengan makna generalisasi- generalisasi- spesialisasi (umum khusus).			
<p>Kebergantungan / <i>dependency</i></p> <div style="text-align: center;">  </div>	Relasi antar kelas dengan makna kebergantungan antar kelas.			

Simbol	Deskripsi
Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua- bagian (<i>whole- part</i>).

Sumber : (Rosa A.S and Shalahuddin, 2019)

2.11 Use Case Diagram

Use Case adalah permodelan untuk perilaku sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. secara kasar, *use case* digunakan untuk mengetahui apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. (Rosa A.S and Shalahuddin, 2019). Simbol-simbol yang ada pada diagram *Use Case* dapat dilihat pada tabel 2.2 di bawah ini :

Tabel 2. 2 Use Case Diagram

Simbol	Deskripsi
<i>Use Case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
Aktor/ <i>actor</i> 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang.
Asosiasi/ <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor

Simbol	Deskripsi
Ekstensi/ <i>extend</i> << <i>extend</i> >> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek biasanya <i>use case</i> tambahan memiliki nama depan
Generalisasi/ <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.

Tabel 2. 3 Use Case Diagram (Lanjutan)

Simbol	Deskripsi
Menggunakan/ <i>Include/uses</i> << <i>include</i> >> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini

Sumber : (Rosa A.S and Shalahuddin, 2019)

2.12 Sequence Diagram

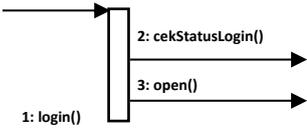
Sequence diagram adalah gambaran kelakuan objek pada *use case* dengan mendeskripsikan waktu onjek dan message yang akan dikimkan lalu diterima sesama objek. Untuk membuat sebuah diagram *sequence* maka harus mengetahui objek apa saja yang terlibat dalam suatu *use case* dengan metode yang dimiliki oleh kelas yang diinstansikan menjadi objek tersebut. Dalam pembuatan diagram *sequence* harus melihat skenario yang ada pada *use case*.

(Rosa A.S and Shalahuddin, 2019). Simbol-simbol yang ada pada diagram *Sequence* dapat dilihat pada tabel 2.3 di bawah ini :

Tabel 2. 4 *Sequence* Diagram

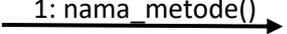
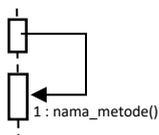
Simbol	Deskripsi
<p>Aktor</p>  <p>Atau</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;"> <p>nama aktor</p> <hr style="width: 100%;"/> </div> <p>Tanpa waktu aktif</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan dengan nama aktor.</p>

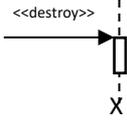
Tabel 2. 5 *Sequence* Diagram (Lanjutan)

Simbol	Deskripsi
<p>garis hidup / <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek.</p>
<p>objek</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;"> <p>nama objek : nama kelas</p> <hr style="width: 100%;"/> </div>	<p>Menyatakan objek yang berinteraksi pesan.</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p> 

	<p>Maka cekStatusLogin() dan open() dilakukan di dalam metode login()</p> <p>Aktor tidak memiliki waktu aktif</p>
<p>pesan tipe create</p> <p><<create>></p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>

Tabel 2. 6 *Sequence* Diagram (Lanjutan)

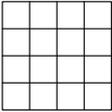
Simbol	Deskripsi
<p>Pesan tipe call</p> <p>1: nama_metode()</p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p>  <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.</p>
<p>Pesan tipe send</p> <p>1 : masukan</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>

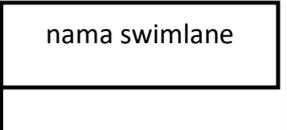
<p>Pesan tipe return</p> <p>1 : keluaran</p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>
<p>Pesan tipe destroy</p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>.</p>

2.13 Activity Diagram

Diagram aktifitas atau *Activity Diagram* berfungsi untuk menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. (Rosa A.S and Shalahuddin, 2019). Simbol-simbol yang ada pada diagram aktivitas dapat dilihat pada tabel 2.3 di bawah ini :

Tabel 2. 7 Activity Diagram

Simbol	Deskripsi
<p>Status awal</p> 	<p>Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.</p>
<p>Aktivitas</p> 	<p>Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.</p>
<p>Percabangan/<i>decision</i></p> 	<p>Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu</p>
<p>Penggabungan/<i>join</i></p> 	<p>Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu</p>
<p>Tabel</p> 	<p>Suatu file komputer dari mana data bisa dibaca atau direkam selama kejadian bisnis</p>

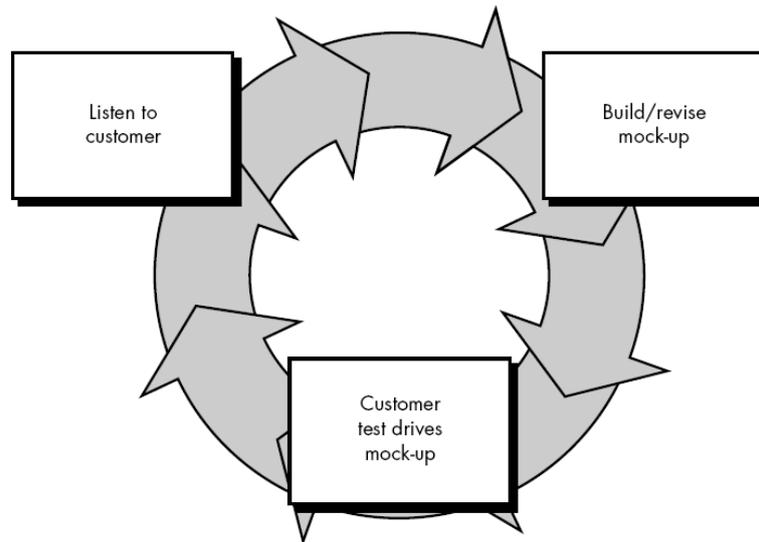
Simbol	Deskripsi
Dokumen 	Menunjukkan dokumen sumber atau laporan
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber : (Rosa A.S and Shalahuddin, 2019)

2.14 Metode Prototype

Sering pelanggan (*customer*) membayangkan kumpulan kebutuhan yang diinginkan tapi tidak terspesifikasi secara detail dari segi masukan (*input*), proses, maupun keluaran (*output*). Di sisi lain seorang pengembang perangkat lunak harus menspesifikasikan sebuah kebutuhan secara detail dari segi teknis dimana pelanggan sering kurang mengerti mengenai hal teknis ini.

Model *prototype* dapat digunakan untuk menyambungkan ketidakpahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak. Model *prototype* (*prototyping model*) dimulai dari mengumpulkan kebutuhan pelanggan terhadap perangkat lunak yang akan dibuat. Lalu dibuatlah program *prototype* agar pelanggan lebih terbayang dengan apa yang sebenarnya diinginkan. Program *prototype* biasanya merupakan program yang belum jadi. Program ini biasanya menyediakan tampilan dengan simulasi alur perangkat lunak sehingga tampak seperti perangkat lunak yang sudah jadi. Program perangkat lunak ini dievaluasi oleh pelanggan atau *user* sampai ditemukan spesifikasi yang sesuai dengan keinginan pelanggan atau *user*. Model *prototype* dapat dilihat pada gambar 2.2



Gambar 2. 2 Model *Prototype*

Sumber : (Rosa A.S and Shalahuddin, 2019)

Mock-Up adalah sesuatu yang digunakan sebagai model desain yang digunakan untuk mengajar, demonstrasi, evaluasi desain, promosi, atau keperluan lain. Sebuah *mock-up* disebut sebagai *prototype* perangkat lunak jika menyediakan atau mampu mendemonstrasikan sebagian besar fungsi sistem perangkat lunak dan memungkinkan pengujian desain sistem perangkat lunak. Iterasi terjadi pada pembuatan *prototype* sampai dengan keinginan pelanggan atau *user*. Sistem perangkat lunak yang sebenarnya dikembangkan bersamaan dengan mengembangkan *prototype* sehingga sesuai dengan kebutuhan pelanggan atau *user*.

Model *prototype* memiliki kelemahan sebagai berikut :

- a. Pelanggan dapat mengubah atau mengambah spesifikasi kebutuhan karena menganggap aplikasi sudah dengan cepat dikembangkan, karena adanya iterasi ini dapat menyebabkan pengembang banyak mengalah dengan pelanggan karena perubahan atau penambahan spesifikasi kebutuhan perangkat lunak.
- b. Pengembang lebih sering mengambil kompromi dengan pelanggan untuk mendapatkan *prototype* dengan waktu yang cepat sehingga pengembang lebih sering melakukan segala cara (tanpa identitas) guna menghasilkan *prototype* untuk mendemonstrasikan. Hal ini

dapat menyebabkan kualitas perangkat lunak yang kurang baik atau bahkan menyebabkan interaktif tanpa akhir.

Metode *prototype* memiliki permasalahan yang dapat diatasi dengan melakukan perjanjian antara pengembang perangkat lunak dengan *user* supaya model *prototype* yang dibuat hanya digunakan untuk kebutuhan *user* mendefinisikan spesifikasi kebutuhan perangkat lunak, dan tidak digunakan untuk seluruh proses pengembangan sistem perangkat lunak. Metode *prototype* cocok digunakan untuk menjabarkan kebutuhan pelanggan dengan lebih detail karena sering kali pelanggan atau *user* kesulitan dalam menyampaikan kebutuhan sistem yang diperlukan secara detail tanpa melihat gambaran yang jelas.

Alasan peneliti menggunakan metode *prototype* adalah, karena sesuai dengan alur kegiatan yang akan dilakukan oleh penulis, yaitu :

1. *Listen To Customer*

Pada tahap ini, peneliti melakukan analisis ke sekolah, seperti wawancara dan pengamatan langsung pada MAN 2 Bandar Lampung dengan mendengarkan keluhan yang disampaikan oleh narasumber terkait permasalahan yang terjadi.

2. *Build and Revise Mock-Up*

Pada tahap ini, peneliti melakukan desain menggunakan UML (*Unified Modelling Language*) sesuai dengan kebutuhan pelanggan seperti *usecase*, *class diagram*, dan *activity diagram*.

3. *Customer Test Drives Mock-Up*

Pada tahap ini, peneliti melakukan pengujian sistem, apakah sudah sesuai dengan kebutuhan yang diinginkan oleh pelanggan atau belum. Apabila sistem belum terpenuhi,

maka peneliti akan melakukan perbaikan sistem lalu selanjutnya akan melakukan pengujian pada sistem tersebut menggunakan ISO 25010.

2.15 ISO/IEC 25010

ISO/IEC 25010 adalah metode evaluasi dengan fokus evaluasi pada aspek *functional suitability, performance efficiency, portability, dan usability*. (Tangkudung et al., 2019) ISO/IEC 25010 merupakan pengganti ISO 9126, digunakan untuk tiga model kualitas yang berbeda untuk perangkat lunak yang digunakan sebagai standar yang di rilis untuk menggantikan ISO 9126 yang sudah tidak relevan pada teknologi yang berkembang pada saat ini. (Murdiani & Umar, 2020).

Beberapa product perangkat lunak yang digunakan, antara lain :

- a. Model kualitas yang digunakan
- b. Model kualitas produk
- c. Model kualitas data

Definisi karakteristik ISO/IEC 25010 adalah sebagai berikut :

1. *Functional suitability* adalah pengukuran untuk melihat kualitas produk yang dikembangkan apakah sudah memenuhi kebutuhan saat menggunakan produk dalam keadaan tertentu. Dalam hal ini perangkat lunak dapat memenuhi kelayakan dari sebuah fungsi untuk melakukan pekerjaan yang spesifik bagi pengguna dan dapat memberikan hasil yang tepat dan ketelitian terhadap tingkat kebutuhan pengguna.
2. *Reliability* adalah pengukuran untuk melihat kualitas produk dimana produk aplikasi dapat mempertahankan kinerja pada level tertentu ketika digunakan dalam rentang waktu tertentu.

3. *Performance efficiency* adalah pengukuran kinerja penggunaan sumber daya dimana produk aplikasi menyediakan performa yang baik dengan jumlah resource yang digunakan.
4. *Usability* adalah pengukuran terhadap kelayakan produk yang mudah dimengerti, dipakai dan menarik untuk digunakan.
5. *Security* adalah pengukuran terhadap tingkat keamanan produk aplikasi menyediakan layanan untuk melindungi akses, penggunaan, modifikasi, pengrusakan, atau pengungkapan yang berbahaya.
6. *Compatibility* adalah pengukuran terhadap suatu komponen aplikasi atau lebih untuk bertukar informasi dan melakukan fungsi yang dibutuhkan ketika digunakan pada *hardware* atau lingkungan perangkat lunak yang sama.
7. *Maintainability* adalah pengukuran terhadap aspek perawatan produk. Dalam hal ini modifikasi adalah perbaikan, perubahan atau penyesuaian perangkat lunak untuk dapat berubah pada lingkungan , kebutuhan dan fungsionalitas yang spesifik.
8. *Portability* adalah pengukuran kemampuan produk yang dihasilkan suatu perpindahan antara perangkat yang satu dan yang lain dari lingkungan yang satu ke lingkungan yang lain.