

## **BAB II**

### **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

Penelitian yang berkaitan dengan penjualan dan monitoring servis laptop sudah banyak dilakukan oleh para peneliti terdahulu dengan tujuan dan lingkup permasalahan yang berbeda-beda. Penelitian tentang sistem informasi penjualan laptop sebelumnya dengan judul “Perancangan Sistem Informasi Penjualan Laptop Berbasis Web”. Penelitian ini bertujuan untuk mengatasi masalah yang ada pada proses transaksi penjualan dan mempermudah dalam pembuatan laporan. Metode yang digunakan dalam pengembangan sistem adalah Metode *Waterfall*. Hasil dari penelitian ini memudahkan proses transaksi penjualan, mempermudah pendataan stok barang dan penyusunan laporan yang baik [1].

Penelitian yang serupa berjudul “Rancang Bangun Sistem Informasi Penjualan Laptop Berbasis Web”. Penelitian ini bertujuan untuk mempermudah proses pemasaran dan promosi ke jangkauan yang lebih luas serta untuk mempermudah proses pendataan. Metode yang digunakan dalam penelitian ini menggunakan Metode *Waterfall*. Hasil dari penelitian ini memudahkan proses penjualan secara online, selain itu mampu membantu kinerja pada proses administrasi penjualan dan pembuatan laporan, dan mampu mempertahankan dalam hal persaingan bisnis [2].

Penelitian berikutnya mengenai servis laptop yang berjudul “Aplikasi Sistem Informasi Jasa *Service* Computer pada Toko Alfafa”. Tujuan dari penelitian tersebut adalah untuk mempermudah proses pencatatan pengolahan data layanan servis. Metode yang digunakan dalam penelitian ini yaitu Metode *Waterfall* dalam pengembangan sistemnya. Hasil yang diperoleh yaitu untuk mempermudah karyawan dalam melakukan pekerjaan, mempermudah dalam pembuatan laporan sehingga laporan yang dihasilkan bisa langsung diserahkan ke pemilik toko serta penyimpanan data menjadi lebih aman dan pencarian data lebih mudah [3].

Penelitian berikutnya dengan judul “Perancangan Sistem Informasi Penerimaan Jasa Servis pada CV. Java Multi Media Yogyakarta”. Penelitian ini bertujuan untuk meminimalisir adanya barang yang tertukar pada saat pengambilan, dan mempermudah dalam

pengelolaan data. Metode yang digunakan pada penelitian ini Metode *Rapid Application Development* (RAD)”. Penelitian ini menghasilkan tidak ada lagi barang yang tertukar saat pengambilan, pengelolaan data servis dan laporan lebih tersusun [4].

Penelitian berikutnya dengan judul “Perancangan Sistem Informasi Pemantauan Jasa Servis Laptop berbasis Web pada Toko Fentacom.Id Kota Depok”. Tujuan pada penelitian ini untuk memberikan informasi kepada pelanggan mengenai status perbaikan perangkat. Penelitian ini menggunakan metode *Web Development Life Cycle* (WDLC). Hasil dari penelitian ini Mempermudah konsumen dalam hal efektivitas waktu [5].

Penelitian kali ini, penulis bermaksud untuk membuat sistem informasi penjualan dan servis laptop berbasis web, dimana sistem ini dapat diakses dimanapun berada selama tersedia jaringan. Perbedaan sistem ini dengan penelitian sebelumnya yaitu pada sistem ini memberikan informasi mengenai spesifikasi laptop serta mempermudah konsumen memilih laptop dengan adanya fitur ulasan, pencatatan stok barang yang dapat dilihat oleh konsumen juga, memantau progres perbaikan saat servis agar tercipta komunikasi yang efektif, memberikan kemudahan kepada konsumen dan admin untuk mengetahui riwayat transaksi servis dan penjualan yang sudah pernah dilakukan. Sistem ini diharapkan dapat membantu masyarakat memudahkan proses administratif. Selain itu, bagi pemilik toko dapat membantu promosi secara lebih luas.

**Tabel 2.1** Perbandingan Kajian Penelitian Sebelumnya

No	Judul, Penulis, Tahun	Tujuan	Metode Pengembangan	Proses Dalam Sistem	Hasil
1	Perancangan Sistem Informasi Penjualan Berbasis Web	<ul style="list-style-type: none"> <li>Mampu mengatasi masalah yang ada pada proses transaksi penjualan</li> <li>Mempermudah dalam pembuatan laporan</li> </ul>	Menggunakan metode <i>waterfall</i> dalam pengembangan sistem	<ul style="list-style-type: none"> <li>Input registrasi member</li> <li>Login member dan admin</li> <li>Input pembelian barang</li> <li>Informasi detail pesanan</li> <li>Input konfirmasi pembayaran</li> <li>Input detail dan lihat detail produk</li> <li>Konfirmasi pesanan dan pembayaran</li> <li>Cetak laporan</li> </ul>	<ul style="list-style-type: none"> <li>Mempermudah proses transaksi penjualan</li> <li>Mempermudah pendataan stok barang</li> <li>Tidak membutuhkan waktu lama dalam pembuatan laporan</li> </ul>
2	Rancang Bangun Sistem Informasi Penjualan Berbasis Web	<ul style="list-style-type: none"> <li>Mempermudah proses penasarana dan promosi ke jangkauan yang lebih luas</li> <li>Mempermudah proses pendataan</li> </ul>	Metode yang digunakan dalam penelitian ini yaitu Metode <i>Waterfall</i>	<ul style="list-style-type: none"> <li>Login member dan admin</li> <li>Lihat katalog</li> <li>Input pesanan</li> <li>Input konfirmasi pembayaran</li> <li>Input katalog</li> <li>Konfirmasi pesanan</li> <li>Konfirmasi pembayaran</li> <li>Cetak laporan</li> </ul>	<ul style="list-style-type: none"> <li>Mempermudah proses penjualan secara online</li> <li>Tidak ada kesalahan dalam proses pendataan</li> <li>Mampu mempertahankan hal persaingan bisnis</li> </ul>

No	Judul, Penulis, Tahun	Tujuan	Metode Pengembangan	Proses Dalam Sistem	Hasil
3	Aplikasi Sistem Jasa Service Computer Pada Toko Alfaf	<ul style="list-style-type: none"> <li>Mempermudah pencatatan dan pengolahan data layanan servis</li> <li>Mempermudah pembuatan laporan</li> </ul>	Penelitian ini menggunakan Metode <i>Waterfall</i> dalam membangun sistem	<ul style="list-style-type: none"> <li>Login admin</li> <li>Input data konsumen</li> <li>Input transaksi</li> <li>Cetak nota</li> <li>Lihat laporan</li> </ul>	<ul style="list-style-type: none"> <li>Mempermudah karyawan melakukan pekerjaan</li> <li>Pembuatan laporan tidak membutuhkan waktu yang lama dan laporan yang dihasilkan jelas</li> <li>Penyimpanan data menjadi lebih aman dan proses pencarian data menjadi lebih mudah</li> </ul>
4	Perancangan Sistem Informasi Penerimaan Jasa Service Pada CV. Java Multi Media Yogyakarta	<ul style="list-style-type: none"> <li>Meminimalisir adanya barang yang tertukar pada saat pengambilan barang</li> <li>Mempermudah dalam mengelola data</li> </ul>	Metode yang digunakan pada penelitian ini yaitu Metode <i>Rapid Application Development</i> (RAD)	<ul style="list-style-type: none"> <li>Login member dan admin</li> <li>Input antrian</li> <li>Lihat antrian</li> <li>Input progress perbaikan</li> <li>Lihat progress perbaikan</li> <li>Lihat detail data servis</li> <li>Cetak laporan</li> </ul>	<ul style="list-style-type: none"> <li>Tidak ada barang yang tertukar saat pengambilan barang</li> <li>Pengelolaan data servis dan laporan servis menjadi lebih mudah</li> </ul>

No	Judul, Penulis, Tahun	Tujuan	Metode Pengembangan	Proses Dalam Sistem	Hasil
5	Perancangan Sistem Informasi Pemanfaatan Jasa Servis Laptop Berbasis Web Pada Toko Fentacom .id Kota Depok	<ul style="list-style-type: none"> <li>Memberikan informasi kepada pelanggan mengenai status perbaikan perangkat</li> </ul>	<p>Untuk mengembangkan sistem penelti menggunakan Metode <i>Rapid Application Development (RAD)</i></p>	<ul style="list-style-type: none"> <li>Login member, admin, dan teknisi</li> <li>Input daftar servis</li> <li>Lihat data servis</li> <li>Input status perbaikan</li> <li>Lihat status perbaikan</li> <li>Lihat data detail servis</li> </ul>	<ul style="list-style-type: none"> <li>Mempermudah konsumen dalam hal efektivitas waktu karena tidak harus bolak balik ke toko untuk menanyakan perbaikan progress</li> <li>Mempermudah teknisi dalam mengerjakan perbaikan karena tidak terganggu dengan konsumen yang menanyakan progress perbaikan barang miliknya</li> </ul>

## 2.2 Landasan Teori

Dalam penelitian ini diperlukan adanya teori yang mendasar untuk menunjang proses penelitian. Adapun teori yang digunakan dalam penelitian ini adalah sebagai berikut :

### 2.2.1 Sistem Informasi

Sistem informasi merupakan suatu kombinasi teratur dari orang-orang, *software*, *hardware*, jaringan komunikasi, serta sumber daya data mengubah, mengumpulkan, dan menyebarkan informasi dalam suatu organisasi [6]. Komponen Sistem Informasi :

- a. ***Input Block (Blok Masukan)***  
Merupakan sebuah data yang masuk ke dalam sistem informasi.
- b. ***Model Block (Blok Model)***  
Merupakan kombinasi dari sebuah prosedur, logika serta metode matematis yang digunakan untuk mengolah data menjadi sebuah *output* yang diinginkan.
- c. ***Output Block (Blok Keluaran)***  
*Output* yang dihasilkan berupa informasi, produk dari sistem informasi, yang dapat digunakan untuk semua level manajemen dan semua pengguna sistem.
- d. ***Technology (Teknologi)***  
Merupakan sebuah alat yang terdapat pada sistem informasi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan keluaran serta memantau pengendalian sistem.
- e. ***Database Block (Blok Basis Data)***  
Merupakan suatu kumpulan data yang saling berkaitan satu sama lain yang tersimpan dalam sebuah komputer.
- f. ***Control Block (Blok Kendali)***  
Dilakukan dengan tujuan untuk menjaga suatu sistem agar sistem dapat mewujudkan tujuan dengan baik.

### 2.2.2 Penjualan

Penjualan merupakan kegiatan ekonomi yang dilakukan oleh perusahaan untuk mempertahankan bisnisnya, berkembang, serta untuk mendapatkan keuntungan yang diinginkan.

Penjualan merupakan kegiatan penyaluran barang dari produsen kepada konsumen dengan tujuan untuk memenuhi kebutuhan konsumen dalam menjalani kelangsungan hidup [7].

Dapat disimpulkan bahwa penjualan merupakan kegiatan ekonomi dalam hal pertukaran barang antara produsen dan konsumen untuk memenuhi kebutuhan hidup konsumen serta untuk meningkatkan perkembangan ekonomi masyarakat. Jenis-jenis penjualan :

**a. Penjualan Tunai**

Merupakan proses transaksi yang dibayarkan dalam satu kali oleh pembeli secara cash.

**b. Penjualan Kredit**

Merupakan penjualan yang proses pembayarannya dengan cara dicicil dalam jangka waktu tertentu. Cicilan yang dibayarkan sesuai dengan lamanya kredit yang diambil. Pada kasus ini dalam setiap cicilan penjual akan menambahkan bunga.

**c. Penjualan Tender**

Merupakan penjualan yang melewati proses tender yang ketentuannya dibuat menyesuaikan dengan kebutuhan antara penjual dan pembeli.

**d. Penjualan Ekspor**

Merupakan kegiatan jual beli yang pembelinya berasal dari luar negeri. Dilakukan oleh pendiri usaha bisnis yang sudah memiliki nama besar.

**e. Penjualan Konsinyasi**

Merupakan penjualan yang melalui perantara atau pihak ketiga sebelum barang atau jasa sampai kepada pembeli.

**f. Penjualan Grosir**

Merupakan penjualan produk dengan cara ecer melalui pedagang grosir.

### 2.2.3 Servis

Pengertian servis merupakan tindakan, kinerja, manfaat serta aktivitas yang cara penyerahannya dilakukan melalui satu pihak ke pihak lain, yang sifatnya tidak berwujud dan tidak bersifat kepemilikan apapun. Sedangkan kata servis mengacu pada konteks reparasi, misalnya servis peralatan elektronik atau sepeda motor [8].

### 2.2.4 Laptop

Laptop atau disebut juga dengan komputer tentang merupakan komputer yang praktis dibawa kemana saja karena penggunaannya tidak perlu menancapkan banyak kabel dan mempunyai ukuran yang fleksibel [9]. Fungsi laptop antara lain :

- a. Sebagai sarana untuk berkomunikasi.
- b. Sarana penunjang pendidikan.
- c. Membantu dalam perancangan desain grafis.
- d. Menyimpan data-data penting.
- e. Membantu menyelesaikan masalah lebih cepat.

### 2.2.5 Website

Menurut Abdullah, *website* merupakan sekumpulan halaman yang terdiri atas beberapa laman yang berisi informasi dalam bentuk data digital, baik berupa teks, gambar, video, audio, dan animasi lainnya yang disediakan melalui alur koneksi internet [10]. Adapun jenis-jenis *website* antara lain :

**a. Website statis**

Adalah *website* yang berisi sebuah data dan informasi yang tetap atau tidak berubah-ubah. Dokumen web yang telah dikirim oleh *client* isinya sama dengan web server.

**b. Website dinamis**

Adalah sebuah *website* dengan data dan informasi yang berbeda-beda tergantung dengan input yang disampaikan oleh *client*.

### 2.2.6 Rekayasa Perangkat Lunak (RPL)

Rekayasa Perangkat Lunak (RPL) adalah suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal yaitu analisa kebutuhan pengguna, menentukan spesifikasi dari kebutuhan pengguna, desain, pengkodean, pengujian, sampai pemeliharaan sistem setelah dikembangkan [11]. Rekayasa perangkat lunak memiliki beberapa pokok bahasa penting yaitu :

**a. Domain Engineering**

Kemampuan untuk memahami suatu permasalahan yang muncul dan dijadikan suatu proyek perangkat lunak.



**b. Requirement Engineering**

Kemampuan untuk memahami kebutuhan seorang pengguna sekaligus dalam memecahkan permasalahan.

**c. Software Design**

Kemampuan untuk memahami dan mengimplementasikan dalam perancangan perangkat lunak yang didalamnya terdapat aspek HCI (*Human Computer Interaction*).

**d. Development**

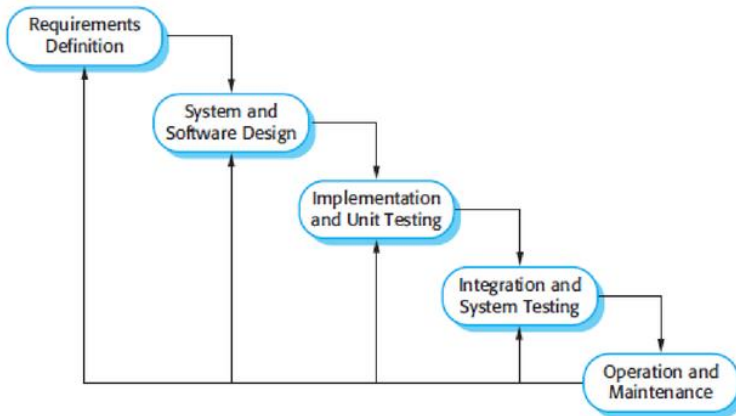
Proses pengembangan yang melibatkan pembelajaran mengenai algoritma, bahasa pemrograman yang di terapkan, dan teknik yang didalamnya berkaitan dengan dengan basis data dan sistem informasi.

**e. Maintenance**

Pemeliharaan dan pengembangan suatu sistem yang berguna untuk banyak orang [12].

**1) Metode Pengembangan Sistem**

Metode yang digunakan adalah metode *waterfall* menurut Ian Sommerville. Metode *waterfall* adalah suatu proses pengembangan perangkat lunak secara berurutan dan diibaratkan seperti air terjun karena kemajuannya dipandang terus mengalir kebawah melewati fase perencanaan, pemodelan, implementasi, dan pengujian [13]. Dalam proses implementasi metode *waterfall*, sebuah langkah dimulai dari tahapan pertama yang harus diselesaikan sebelum melanjutkan ke tahap berikutnya. Penggunaan metode *waterfall* dapat memungkinkan adanya sedikit perubahan terhadap proyek yang sedang berlangsung. Metode ini mengusulkan sebuah pendekatan kepada pengembangan *software* yang sistematis dan sekuensial yang dimulai dari tingkat kemajuan sistem pada seluruh analisis, desain, kode, pengujian dan pemeliharaan. Metode *waterfall* dapat dilihat pada Gambar 2.1.



**Gambar 2.1** Metode *Waterfall* Menurut Ian Sommerville

Adapun penjelasan penjelasan dari tahapan-tahapan metode *waterfall* yaitu :

a) ***Requirement Analysis and Definition***

Tahap ini dilakukan untuk menganalisa terhadap kebutuhan sistem. Pengumpulan data dapat diperoleh melalui penelitian, wawancara atau *study literature*. Seorang sistem analis akan menggali informasi sebanyak-banyaknya dari *user* sehingga akan tercipta sebuah sistem komputer yang sesuai dengan keinginan *user* tersebut. Pada tahap ini akan menghasilkan dokumen *user requiruitment* atau data yang berhubungan dengan keinginan *user* dalam pembuatan sebuah sistem serta digunakan sebagai acuan sistem analisis untuk menterjemahkan kedalam bahasa pemrograman.

b) ***System and Software Design***

Dalam tahap *System and Software Design* akan dibentuk suatu arsitektur sistem berdasarkan dengan persyaratan yang telah ditetapkan. Selain itu, pada tahap ini juga dilakukan identifikasi dan penggambaran mengenai abstraksi dasar sistem perangkat lunak beserta hubungan-hubungannya.

c) ***Implementation and Unit Testing***

Pada tahap *implementation and unit testing* hasil dari desain perangkat lunak akan direalisasikan sebagai satu set program serta

dilakukan pengujian pada setiap unitnya untuk mengetahui spesifikasinya sudah memenuhi atau belum.

**d) *Integration and System Testing***

Dalam tahap ini setiap unit program akan diintegrasikan antara satu dengan yang lain dan diuji sebagai satu sistem yang utuh untuk memastikan bahwa sistem sudah memenuhi persyaratan yang ada kemudian, sistem akan dikirimkan ke pengguna sistem.

**e) *Operation and Maintenance***

*Operation and maintenance* merupakan tahap dimana sistem diinstal dan mulai digunakan selain itu juga, memperbaiki kesalahan (*error*) yang tidak ditemukan pada tahap pembuatan. Pada tahap ini juga dilakukan pengembangan sistem seperti penambahan fitur dan fungsi baru.

**2) *Metode Pengujian Sistem***

Metode pengujian sistem yang digunakan yaitu metode *black box testing*. Metode *black box testing* merupakan pengujian kualitas perangkat lunak yang berfokus pada fungsionalitas perangkat lunak. Pengujian *black box testing* bertujuan untuk menemukan fungsi yang tidak benar, kesalahan antarmuka, kesalahan pada struktur data, kesalahan performansi, kesalahan inisialisasi dan terminasi [14]. Ciri-ciri *black box testing* :

- a) *Black box testing* fokus pada kebutuhan fungsional berdasarkan pada spesifikasi kebutuhan dari *software*.
- b) *Black box testing* merupakan pendekatan pelengkap dalam mencakup *error* dengan kelas yang berbeda dari metode *white box testing*.
- c) Sistem pengujian metode *black box testing* dilakukan tanpa pengetahuan detail struktur internal dari sistem maupun komponen yang diuji.

**2.2.7 *Pemrograman Berorientasi Objek (PBO)***

Metodologi pemrograman berorientasi objek (PBO) merupakan pemrograman yang berorientasikan kepada objek, dimana semua data dan fungsi dibungkus dalam kelas-kelas atau objek-objek. Setiap objek dapat menerima pesan, memproses data, mengirim, menyimpan dan memanipulasi data. Beberapa objek berinteraksi dengan saling memberikan informasi satu terhadap yang lainnya [15]. Beberapa konsep

dasar yang harus dipahami mengenai metodologi pemrograman berorientasi objek :

**a. Kelas (*class*)**

*Class* adalah *prototype*, atau *blueprint*, atau rancangan yang mendefinisikan variabel dan metode-metode pada seluruh objek tertentu. *Class* berfungsi untuk menampung isi dari program yang akan dijalankan, didalamnya berisi atribut / *type* data dan *method* untuk menjalankan suatu program.

**b. Objek (*object*)**

Objek adalah *instance* dari *class*. Jika *class* secara umum mempresentasikan (*templates*) sebuah objek, sebuah *instance* adalah representasi nyata dari *class* itu sendiri.

**c. Metode (*method*)**

Metode adalah merupakan suatu operasi berupa fungsi-fungsi yang dapat dikerjakan oleh suatu objek. *Method* didefinisikan pada *class* akan tetapi dipanggil melalui objek.

**d. Atribut (*attribute*)**

Atribut merupakan nilai data yang terdapat pada suatu objek yang berasal dari *class*. *Attribute* mempresentasikan karakteristik dari suatu objek.

**e. Abstraksi (*abstract*)**

Prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

**f. Enkapsulasi (*encapsulation*)**

Pembungkusan *attribute* data dan layanan (operasi-operasi) yang dimiliki objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya.

**g. Pewarisan (*inheritance*)**

Mekanisme yang memungkinkan suatu objek mewarisi sebagian atas seluruh definisi objek lain sebagai bagian dari dirinya.

**h. Antarmuka (*interface*)**

Antarmuka atau *interface* sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah *interface* dapat diimplementasikan oleh kelas lain.


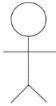



Pada pemrograman berorientasi objek, *Unified Modeling Language* (UML) digunakan sebagai pemodelan sistem. UML

merupakan bahasa yang digunakan untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan *artifacts* (bagian dari informasi guna sebagai proses pembuatan perangkat lunak berupa model, deskripsi atau perangkat lunak) dari suatu sistem perangkat lunak.

### 1) *Use Case Diagram*

*Use case diagram* merupakan deskripsi fungsi dari suatu sistem berdasarkan prespektif pengguna. Cara kerja *use case diagram* yaitu dengan mendeskripsikan tipikal interaksi antara *user* dengan sistem melalui sebuah cerita berdasarkan alur penggunaan sistem. Istilah pengguna dalam *use case diagram* disebut dengan *actor* sedangkan urutan langkah-langkah yang menjelaskan antara pengguna dengan sistem disebut *scenario*. Simbol *use case diagram* dapat dilihat pada Tabel 2.2.

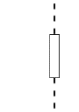
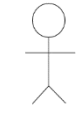
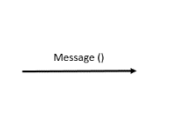
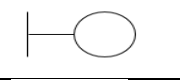


**Tabel 2.2** Simbol *Use Case Diagram*

No	Simbol	Nama	Keterangan
1.		<i>Use case</i>	Deskripsi dari urutan aksi – aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
2.		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
3.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
4.		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
6.		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas

## 2) *Sequence Diagram*

Digunakan untuk menggambarkan perilaku pada sebuah skenario. *Sequence diagram* juga akan menampilkan sebuah perintah atau pesan yang dikirim. Komponen utama *sequence diagram* terdiri dari objek yang dituliskan dengan kotak segiempat bernama. Pesan atau *message* dilambangkan dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical*. Tabel 2.3 merupakan simbol-simbol *sequence diagram*.

**Tabel 2.3** Simbol *Sequence Diagram*

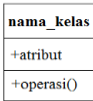





No	Simbol	Nama	Keterangan
1.		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2.		<i>Actor</i>	Menggambarkan <i>user</i> atau pengguna.
3.		<i>Message</i>	Spesifikasi dari komunikasi antar <i>objek</i> yang memuat informasi – informasi tentang aktivitas yang terjadi.
4.		<i>Boundary</i>	Menggambarkan sebuah <i>form</i> .
5.		<i>Control Class</i>	Menghubungkan <i>boundary</i> dengan Tabel.
6.		<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan.

## 3) *Class Diagram*

*Class diagram* merupakan diagram yang digunakan untuk menampilkan beberapa kelas yang ada pada sebuah sistem yang dibangun, *class diagram* menunjukkan bagaimana class dalam sistem yang akan dibangun saling berkolaborasi untuk mencapai suatu tujuan

[16]. *Class diagram* digambarkan dengan bentuk kotak. Simbol-simbol *class diagram* dapat dilihat pada Tabel 2.4.

**Tabel 2.4** Simbol *Class Diagram*

No	Simbol	Nama	Keterangan
1.		Kelas	Kelas pada struktur sistem.
2.		Asosiasi/ <i>Association</i>	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
3.		Asosiasi berarah / <i>Direct Association</i>	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
4.		Generalisasi	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus).
5.		Kebergantungan/ <i>Dependency</i>	Relasi antarkelas dengan makna kebergantungan antarkelas.
6.		Agregasi/ <i>Aggregation</i>	Relasi antarkelas dengan makna semua-bagian ( <i>whole-part</i> ).

### 2.2.8 Basis Data

Basis data merupakan kumpulan data yang saling terintegrasi serta berelasi secara *logic* serta dirancang untuk melengkapi informasi yang dibutuhkan [17].

#### a. *Database Management System (DBMS)*

*Database Management System (DBMS)* perangkat lunak komputer yang berinteraksi dengan Pengguna, aplikasi lain, dan basis data untuk mengambil dan menganalisis data. DBMS memungkinkan definisi,

pembuatan, *query*, *update*, dan administrasi basis data [18]. DBMS menyediakan fasilitas antara lain :

1) **Data Definition Language (DDL)**

DDL memperbolehkan pengguna untuk mendeskripsikan *database*, misalnya merinci tipe dan batasan data yang akan disimpan dalam *database*. Terdapat tiga perintah penting dalam DDL, yaitu :

- a) *CREATE* adalah perintah yang digunakan untuk membuat, termasuk didalamnya membuat *database* baru, tabel baru, *view* baru, dan kolom baru. Contoh : *CREATE DATABASE* nama *database*.
- b) *ALTER* berfungsi untuk mengubah struktur tabel yang telah dibuat. Mencakup didalamnya mengubah nama tabel, menambah kolom, mengubah kolom, menghapus kolom, dan memberikan atribut pada kolom. Contoh : *Alter Table* nama tabel ADD nama kolom datatype.
- c) *DROP* perintah *DROP* berfungsi untuk menghapus *database* atau tabel. Contoh : *DROP DATABASE* nama *database*.

2) **Data Manipulation Language (DML)**

DML memperbolehkan pengguna untuk memanipulasi data dalam *database* yang telah dibuat. Terdapat 4 (empat) perintah penting dalam DML, yaitu :

- a) *INSERT* perintah ini digunakan untuk memasukkan data baru ke dalam sebuah tabel. Perintah ini tentu saja bisa dijalankan ketika *database* dan tabel sudah dibuat. Contoh : *INSERT INTO* nama tabel *VALUES* (data1, data2, dst..)
- b) *SELECT* perintah ini digunakan untuk mengambil dan menampilkan data dari tabel atau bahkan dari beberapa tabel dengan penggunaan relasi. Contoh : *SELECT* nama\_kolom1, nama\_kolom2 *FROM* nama tabel.
- c) *UPDATE* perintah ini digunakan untuk memperbaharui data pada sebuah tabel. Contoh : *UPDATE* nama\_tabel *SET* kolom1=data1, kolom2=data2,.. *WHERE* kolom=data.
- d) *DELETE* perintah ini digunakan untuk menghapus data dari sebuah tabel. Contoh : *DELETE FROM* nama tabel *WHERE* kolom=data.



### 2.2.9 MySQL

MySQL merupakan *database* server yang terenal dan banyak digunakan untuk membangun sebuah aplikasi web yang databasenya digunakan sebagai sumber pengolahan data [19].

### 2.2.10 Hypertext Preprocessor (PHP)

*Hypertext Preprocessor* (PHP) merupakan bahasa pemrograman yang digunakan untuk membuat web yang sifatnya *server-side scripting*. PHP memungkinkan untuk membuat sebuah halaman yang bersifat dinamis, dan untuk menghasilkan halaman *web* yang dinamis tersebut PHP dapat memanfaatkan *database* [20].






### 2.2.11 Codeigniter

*Codeigniter* adalah sebuah *framework* bahasa pemrograman PHP yang bersifat *open source* dan menggunakan model *move, view, controller* (MVC) yang berguna untuk memudahkan pengembang atau programmer dalam membangun sebuah sistem tanpa harus membuatnya dari awal [21].

### 2.2.12 Entity Relationship Diagram (ERD)




*Entity Relationship Diagram* (ERD) merupakan suatu bentuk hubungan atau rancangan kegiatan pada sebuah sistem mempunyai fungsi dan berkaitan langsung di dalam proses tersebut. ERD merupakan suatu bentuk pemodelan dari basis data relasional yang dibangun berdasarkan presespsi dalam dunia nyata terdiri dari objek-objek yang saling berhubungan satu dengan lainnya. Suatu objek disebut *entity* sedangkan hubungan yang dimiliki disebut dengan *relationship*. *Entity* sifatnya unik serta memiliki atribut yang menjadi pembeda dengan *entity* lainnya. Simbol-simbol ERD dapat dilihat pada Tabel 2.5.

**Tabel 2.5** Simbol *Entity Relationship Diagram* (ERD)

No.	Nama	Simbol	Keterangan Fungsi
1.	Entitas		Persegi panjang menyatakan himpunan entitas adalah orang, kejadian, atau berada.
2.	Atribut		Atribut merupakan informasi yang diambil tentang sebuah entitas.
3.	Relasi		Belah ketupat menyatakan himpunan relasi merupakan hubungan antar entitas.
4.	Link		Garis sebagai penghubung antara himpunan, relasi, dan himpunan entitas dengan atributnya
5.	Associative Entity		Entitas yang digunakan pada relasi <i>many-to-many</i> .

Kardinalitas merupakan derajat relasi yang dimiliki oleh ERD. Kardinalitas menjelaskan tentang jumlah batasan keterhubungan *entity* satu dengan lainnya. Macam-macam kardinalitas terdapat pada Tabel 2.6.

**Tabel 2.6** Kardinalitas *Entity Relationship Diagram* (ERD)

No.	Simbol	Nama	Keterangan
1.		Relasi Satu ke Satu ( <i>One to One</i> )	Relasi yang menunjukkan bahwa setiap himpunan entitas berhubungan dengan tepat satu himpunan entitas lainnya
2.		Relasi Satu ke Banyak ( <i>One to Many</i> )	Relasi yang menunjukkan bahwa hubungan antara entitas pertama dengan entitas kedua adalah satu banding banyak, begitu pula sebaliknya
3.		Relasi Banyak ke Banyak	Relasi yang menunjukkan bahwa setiap himpunan entitas boleh berhubungan dengan banyak

No.	Simbol	Nama	Keterangan
		( <i>Many to Many</i> )	himpunan entitas lainnya dan sebaiknya

### 2.2.13 Flowchart

*Flowchart* (diagram alir) merupakan sebuah diagram yang digambarkan menggunakan simbol untuk menyatakan sebuah algoritma atau langkah-langkah instruksi secara berurutan dalam sebuah sistem. Untuk menghubungkan proses satu ke proses selanjutnya digambarkan dengan garis penghubung. Dengan adanya *flowchart* alur sebuah proses dapat digambarkan menjadi lebih jelas dan rinci [22]. Dalam pembuatan program, langkah awal yang dilakukan adalah menyusun *flowchart* untuk mengetahui tahapan alur kegiatan menjadi lebih jelas. Apabila *flowchart* sudah tersusun maka, programmer akan menerjemahkan ke dalam bentuk program menggunakan bahasa pemrograman. Simbol yang digunakan dalam *flowchart* dikelompokkan menjadi 3 bagian, yaitu :

**a. *Flow direction symbols***

Digunakan untuk menghubungkan simbol satu dengan yang lain (*connecting line*).



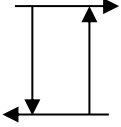
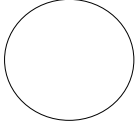
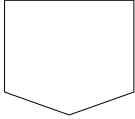
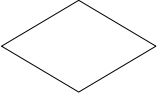
**b. *Processing symbols***



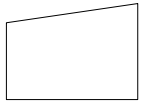

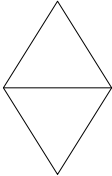
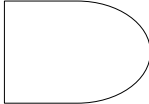

Menunjukkan jenis operasi pengolahan dalam suatu proses atau prosedur.

**c. *Input / Output symbols***

Menunjukkan jenis peralatan yang digunakan sebagai media input atau output. *Flowchart* disusun dengan simbol-simbol yang berfungsi untuk membantu dalam menggambarkan proses di dalam sebuah program. Simbol-simbol yang digunakan antara lain :

**Tabel 2.7** Simbol *Flowchart*

No.	Simbol	Nama	Keterangan
1.		<i>Terminal</i>	Memulai dan mengakhiri suatu program
2.		<i>Input/Output</i>	Memasukan data maupun menunjukkan hasil dari suatu <i>process</i> tanpa tergantung dengan jenis peralatannya
3.		<i>Flow</i>	Menghubungkan antara simbol satu dengan simbol yang lain atau menyatakan jalannya arus dalam suatu proses. Simbol arus ini sering disebut juga dengan <i>connecting line</i> .
4.		<i>On-Page Connector Symbol</i>	Simbol penghubung yang digunakan untuk keluar-masuk atau penyambungan proses dalam halaman yang sama.
5.		<i>Off-Page Connector Symbol</i>	Simbol penghubung yang digunakan untuk keluar-masuk atau penyambungan proses dalam halaman atau lembar yang berbeda.
6.		<i>Decision</i>	Memilih proses berdasarkan kondisi yang ada.

No.	Simbol	Nama	Keterangan
7.		<i>Document</i>	Merupakan simbol untuk data yang terbentuk informasi.
8.		<i>Manual Operation</i>	Menunjukkan pengolahan yang tidak dilakukan oleh komputer/pc.
9.		<i>Manual Input</i>	Memasukan data secara manual <i>on-line</i> keyboard.
10.		<i>Predefined Process Symbol</i>	Berfungsi untuk mempersiapkan penyimpanan yang akan atau sedang digunakan sebagai tempat pengolahan di dalam <i>storage</i> .
11.		<i>Sorting Office Symbol</i>	Berfungsi untuk menunjukan proses pengurutan data di luar proses komputer.
12.		<i>Delay Symbol</i>	Berfungsi untuk tempat menunggu material <i>trolley</i> di proses lebih lanjut.
13.		<i>Display Symbol</i>	Berfungsi untuk menyatakan peralatan output yang digunakan yaitu layar, plotter, printer.

### 2.2.14 Pengujian *Usability*

*Usability* adalah atribut yang memiliki kualitas dan menjelaskan kualitas serta menilai bagaimana kemudahan antarmuka pengguna untuk digunakan. *Usability* sendiri memiliki komponen-komponen yang digunakan untuk mengukur pendapat dari responden. Berikut merupakan komponen-komponen dalam pengujian *usability*:

a. ***Learnability***

*Learnability* merupakan pengukuran tingkat kemudahan pengguna dari penggunaan fungsi serta fitur yang ada pada sistem atau aplikasi.

b. ***Efficiency***

*Efficiency* merupakan kecepatan pengguna dalam mengerjakan perintah pada sistem atau aplikasi.

c. ***Memorability***

*Memorability* merupakan kemampuan pengguna dalam menjaga pengetahuannya setelah jangka waktu tertentu.

d. ***Errors***

*Errors* menjelaskan tentang cara memperbaiki *error* serta jumlah *error* yang dilakukan pengguna.

e. ***Satisfaction***

*Satisfaction* merupakan tingkat kepuasan pengguna setelah pengguna menggunakan sistem atau aplikasi.

Setiap pertanyaan di atas menggunakan skala likert yang digunakan untuk mengukur pendapat atau sikap dari perorangan atau sekelompok terhadap fenomena sosial tertentu. Setiap pertanyaan memiliki interval jawaban 1-4 [23]. Adapaun jawaban dari setiap pertanyaan yang digunakan dapat dinilai dengan skor yang dapat dilihat pada Tabel 2.8.

**Tabel 2.8** Skala *Likert*

Jawaban	Skor
Sangat tidak setuju	1
Kurang Setuju	2
Setuju	3
Sangat Setuju	4

Hasil dari kuesioner dari responden, kemudian dilakukan perhitungan dengan menggunakan rumus sebagai berikut:

$$U = \frac{np}{nm} \times 100\%$$

$U$  = Nilai hasil *usability*

$np$  = Nilai perolehan, jumlah seluruh hasil kuesioner/jawaban dari responden

$nm$  = Nilai maksimal, jumlah maksimal yang dapat diperoleh (jumlah responden x nilai maksimal skala likert yang dipakai x jumlah pertanyaan)

Kemudian hasil dari pengujian *usability* dicocokkan dengan kategori interpretasi peringkat sistem. Berikut merupakan kategori interpretasi peringkat sistem yang dapat dilihat pada Tabel 2.9.

**Tabel 2.9** Kategori Interpretasi Peringkat Sistem

<b>Presentasi Hasil Uji <i>Usability</i></b>	<b>Predikat</b>
81 – 100 %	Sangat Baik
61 – 80 %	Baik
41 – 60 %	Cukup Baik
21 – 40 %	Kurang Baik
0 – 20 %	Tidak Baik

*~ Halaman ini sengaja dikosongkan*