

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian lain telah dilakukan oleh salah satu mahasiswa bernama Paseng⁴ dengan judul “Sistem Informasi Pemesanan Tiket Kapal Laut Berbasis *Web* (Studi Kasus: PT Paradugma Baru Luwuk Banggai, Sulawesi Tengah)”. Tujuan dari penelitian ini untuk merancang dan membangun sistem informasi pemesanan tiket berbasis *web* di PT. Paradigma Baru Luwuk sehingga dapat membantu dalam hal pemesanan tiket secara tepat dan optimal. Bahasa pemrograman yang digunakan dalam sistem tersebut adalah *Php* dan *DBMS*.

Dari penelitian lain yang dilakukan oleh Bakhtiar⁵ dengan judul “Sistem Informasi Pemesanan Tiket Berbasis *Web* pada Perum Damri Palembang Menggunakan Metode *Waterfall*”. Sistem ini bertujuan untuk membangun aplikasi sistem informasi pemesanan tiket berbasis *web* menggunakan script *PHP* dan *MySQL*, untuk membantu Perum Damri Cabang Palembang menyebarkan informasi, layanan transportasi kepada masyarakat yang lebih cepat, akurat, dan efisien dengan memanfaatkan teknologi internet.

Penelitian lain dengan judul “Sistem Informasi Administrasi Pengelolaan Pembayaran Karcis Penyeberangan Berbasis *Website* (Studi Kasus : Tambatan Preca Kelurahan Donan Kabupaten Cilacap)” yang dilakukan oleh Krismanto.⁶ Sistem ini di buat untuk membantu kasir dalam menjalankan tugasnya yang masih mengalami kesulitan dalam mencatat pengolaan data penumpang, rekap penghasilan, dan pengolahan dana operasional. Selain itu kasir juga kesulitan menghitung pembagian penghasilan yang harus dibayarkan kepada anggota paguyuban.

Penelitian lain dengan judul “Sistem Pemesanan Pesawat Berbasis *Web*” yang dilakukan oleh Ibrahim.⁷ Sistem ini dibuat untuk memberikan informasi tentang penerbangan dan pemesanan tiket kepada konsumen. Obyek yang digunakan dalam permasalahan ini adalah PT. Lion Air. Selain sarana informasi pemesanan, sistem tersebut juga merupakan salah satu media penyedia informasi global mengenai perusahaan yang bersangkutan kepada khalayak ramai.

Kemudian penelitian lain dengan judul “Sistem Pemesanan Menu Makanan Dan Minuman Rumah Makan Berbasis *Website*” yang dilakukan oleh Hidayah.⁸ Dengan sistem ini pegawai juga merasa dimudahkan dalam pengecekan pemesanan pelanggan maupun perekapan pesanan. Pemesanan menu oleh pelanggan juga memanfaatkan sistem QR Code. Penggunaan QR Code ini bertujuan untuk menggantikan inputan manual nomor meja dan nama pelanggan. QR Code ini dilakukan dengan cara men-scan kode menggunakan smartphone masing-masing pelanggan.

Berdasarkan penelitian-penelitian yang telah ada sebelumnya, berbeda dengan penelitian lainnya pada penelitian ini akan dibuat sistem informasi berbasis *website* dan database sebagai penyimpanan data-data para nelayan dan berbagai objek wisata tersebut. Sehingga dapat mengenalkan objek wisata apa saja dan cara menuju tempat wisata yang ada di daerah sekitar Pantai Teluk Penyus Cilacap dengan berbasis Website yang bisa di akses kapan dan dimana saja sehingga bisa mempermudah para nelayan maupun wisatawan yang datang untuk berwisata di daerah sekitar Pantai Teluk Penyus Cilacap.

2.2 Landasan Teori

2.2.1 Sistem

Sistem adalah kumpulan/group dari sub sistem/bagian/komponen apapun baik fisik ataupun non fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan tertentu.⁹

2.2.2 Sistem Pemesanan

Sistem Pemesanan Jasa adalah sistem dalam suatu perusahaan jasa yang pada kegiatannya menawarkan sesuatu barang yang belum jadi atau barang yang tidak berwujud. Setelah melalui proses produksi barang yang dipesan baru akan terlihat hasilnya, hasil dari proses produksi sesuai dengan proses pemesanan yang dilakukan oleh konsumen.

2.2.3 Basis Data

Basis Data adalah kumpulan data yang terorganisir, yang umumnya disimpan dan diakses secara elektronik dari suatu sistem komputer. Pada saat pangkalan data menjadi semakin kompleks, maka

pangkalan data dikembangkan menggunakan teknik perancangan dan pemodelan secara formal.

2.2.4 Website

Website merupakan kumpulan halaman-halaman yang berasal dari berkas yang berisi Bahasa pemrograman yang berhubungan dan digunakan untuk menyajikan informasi, gambar, video, audio yang bersifat dinamis maupun statis.

2.2.5 Rekayasa Perangkat Lunak

A. Metode Pengembangan Sistem

Metode ini dilakukan dengan pendekatan yang sistematis, mulai dari tahap kebutuhan sistem lalu menuju ke tahap Analisa Kebutuhan, Desain Sistem, Penulisan Kode Program, Pengujian Sistem dan Penerapan program dan Pemeliharaan. Langkah demi langkah yang dilalui harus diselesaikan satu per satu (tidak dapat meloncat ke tahap berikutnya) dan berjalan secara berurutan, oleh karena itu di sebut waterfall (Air Terjun). Berikut pengertian langkah-langkah tersebut.

1. Analisa Kebutuhan : Sebelum melakukan pengembangan perangkat lunak, seorang pengembang harus mengetahui dan memahami bagaimana informasi kebutuhan pengguna terhadap sebuah perangkat lunak. Metode pengumpulan informasi ini dapat diperoleh dengan berbagai macam cara diantaranya, diskusi, observasi, survei, wawancara, dan sebagainya. Informasi yang diperoleh kemudian diolah dan dianalisa sehingga didapatkan data atau informasi yang lengkap mengenai spesifikasi kebutuhan pengguna akan perangkat lunak yang akan dikembangkan.

2. Desain Sistem : Informasi mengenai spesifikasi kebutuhan dari tahap Requirement Analysis selanjutnya di analisa pada tahap ini untuk kemudian diimplementasikan pada desain pengembangan. Perancangan desain dilakukan dengan tujuan membantu memberikan gambaran lengkap mengenai apa yang harus dikerjakan. Tahap ini juga akan membantu pengembang untuk menyiapkan kebutuhan hardware dalam pembuatan arsitektur sistem perangkat lunak yang akan dibuat secara keseluruhan.

3. Penulisan Kode Program : Tahap implementation and unit testing merupakan tahap pemrograman. Pembuatan perangkat lunak dibagi menjadi modul-modul kecil yang nantinya akan digabungkan dalam tahap berikutnya. Disamping itu, pada fase ini juga dilakukan pengujian dan pemeriksaan terhadap fungsionalitas modul yang sudah dibuat, apakah sudah memenuhi kriteria yang diinginkan atau belum.

4. Pengujian Program : Setelah seluruh unit atau modul yang dikembangkan dan diuji di tahap implementasi selanjutnya diintegrasikan dalam sistem secara keseluruhan. Setelah proses integrasi selesai, selanjutnya dilakukan pemeriksaan dan pengujian sistem secara keseluruhan untuk mengidentifikasi kemungkinan adanya kegagalan dan kesalahan sistem.

5. Penerapan dan Pemeliharaan Program : Pada tahap terakhir dalam Metode Waterfall, perangkat lunak yang sudah jadi dioperasikan pengguna dan dilakukan pemeliharaan. Pemeliharaan memungkinkan pengembang untuk melakukan perbaikan atas kesalahan yang tidak terdeteksi pada tahap-tahap sebelumnya. Pemeliharaan meliputi perbaikan kesalahan, perabikan implementasi unit sistem, dan peningkatan dan penyesuaian sistem sesuai dengan kebutuhan.

B. Metode Pengujian Sistem

1. *Black-Box Testing*

Lebih sering dikenal dengan sebutan pengujian fungsional merupakan metode pengujian perangkat lunak yang digunakan untuk menguji perangkat lunak tanpa mengetahui struktur internal kode atau program.

Dalam pengujian ini, tester menyadari apa yang harus dilakukan oleh program tetapi tidak memiliki pengetahuan tentang bagaimana melakukannya. Pada Black-Box Testing ini dilakukan pengujian yang didasarkan pada detail aplikasi seperti tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi, dan kesesuaian alur fungsi dengan bisnis proses yang diinginkan oleh customer. Black-box testing ini lebih menguji ke tampilan luar (Interface) dari suatu aplikasi agar mudah digunakan oleh pengguna. Pengujian ini tidak melihat dan menguji source code program. Black-box testing bekerja dengan mengabaikan struktur kontrol sehingga perhatiannya hanya terfokus pada informasi domain.

2.2.6 Pemrograman Berbasis Orientasi

Pemrograman berorientasi objek adalah sebuah paradigma pemrograman yang berorientasikan kepada sebuah objek. Seluruh fungsi dan data di dalam paradigma ini dibalut dalam suatu objek-objek. Setiap objek dapat menerima pesan, mengirim pesan, memproses data dan ke objek lainnya secara terstruktur.

Berikut konsep dasar dalam mendalami Pemrograman Berbasis Objek :

1) Objek

Objek merupakan pembungkus data dan fungsi secara bersama menjadi suatu unit dalam sebuah program. Objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.

2) Kelas

Kelas merupakan purwarupa yang mendefinisikan data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu.

3) *Encapsulation* (Pembungkusan)

Encapsulation adalah mekanisme pemrograman yang membungkus kode dan data yang dimanipulasi dan menjaganya supaya terhindar dari interferensi dan penggunaan yang tidak perlu. Salah satu caranya dengan membentuk objek.

4) *Polymorphism* (Perbedaan Bentuk)

Polymorphisme artinya mempunyai banyak bentuk. Dua objek atau lebih dikatakan sebagai *polymorphic*, bila objek-objek itu mempunyai antar muka yang *identic* namun mempunyai perilaku perilaku yang berbeda.

5) Abstraksi

Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan.

6) *Inheritance* (Pewarisan)

Inheritance memungkinkan programmer meletakkan member yang sama dalam satu kelas dan kelas - kelas lain dapat mewarisi member tersebut. Kelas yang mengandung anggota yang sama dari beberapa kelas lain dinamakan *superclass* atau *parent class*. Kelas yang mewarisi

dinamakan *subclass* atau *child class*. *Inheritance* menghasilkan *class hierarchy*.

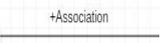
2.2.7 Unified Modeling Language (UML)

Unified Modelling Language (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak.¹⁰ UML digunakan untuk membuat model perangkat lunak, dimana yang dapat berkerja pada perangkat keras, sistem operasi, dan sistem jaringan. UML mendeskripsikan notasi khusus yang digunakan untuk menggambarkan jalanya sistem yang akan dikembangkan.

a) Use Case

Use case adalah pemodelan sistem informasi yang akan dibangun dengan mencantumkan kelakuan atau interaksi antara *system* dan *actor*. *Use case* digunakan untuk memberi informasi mengenai fungsi yang berada di dalam sistem dan siapa yang menggunakan fungsi-fungsi tersebut dapat dilihat pada Tabel 2.1.

Tabel 2.1 *Use Case Diagram*

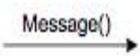
NO	Simbol	Fungsi
1		Segala sesuatu yang berinteraksi dengan sistem aplikasi komputer. Jadi <i>actor</i> ini bisa berupa orang, perangkat keras atau mungkin juga obyek lain dalam sistem yang sama.
2		Menspesifikasikan bahwa perilaku <i>use case</i> merupakan bagian dari <i>use case</i> lain.
3		Menggambarkan navigasi antar <i>class</i> , berupa banyak obyek lain yang berhubungan dengan satu obyek, dan apakah suatu <i>class</i> menjadi bagian dari <i>class</i> lainnya.

4		<i>System Boundary</i> yaitu batasan sebuah sistem.
5		Use case menjelaskan urutan kegiatan yang dilakukan aktor dan sistem untuk mencapai suatu tujuan tertentu. Walaupun menjelaskan kegiatan, use case hanya menjelaskan apa yang dilakukan oleh aktor dan sistem, bukan bagaimana aktor dan sistem melakukan kegiatan.

b) Sequence Diagram

Sequence Diagram merupakan *diagram* yang menjelaskan *behavior* (kelakuan) objek ke *use case*. Kelakuan objek nanti akan dijelaskan dengan mendeskripsikan jangka hidup objek dan pesan yang diterima dan dikirim dapat dilihat pada Tabel 2.2.

Tabel 2.2 *Sequence Diagram*

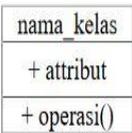
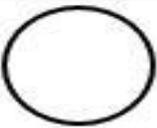
No	Simbol	Fungsi
1		<i>Lifeline</i> , antarmuka yang saling berinteraksi
2		<i>Actor</i> , Digunakan untuk menggambarkan user/Pengguna
3		<i>Message</i> , Spesifikasi dari komunikasi antar objek yang memuat informasi – informasi tentang aktifitas yang terjadi.
4		<i>Boundary Class</i> , Digunakan untuk menggambarkan sebuah <i>form</i> .
5		<i>Control Class</i> , Digunakan untuk menghubungkan <i>Boundary</i> dengan tabel

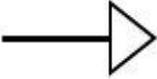
6		<i>Entity Class</i> , Menggambarkan hubungan kegiatan yang akan dilakukan.

c) Class Diagram

Salah satu jenis diagram yang paling berguna di *UML*, hal ini karena dapat dengan jelas memetakan struktur sistem tertentu dengan memodelkan kelas, atribut, operasi serta hubungan antar objek. *Class Diagram* menggambarkan serta deskripsi atau penggambaran dari class, atribut, dan objek disamping itu juga hubungan satu sama lain seperti pewarisan, *containmet*, asosiasi dan lainnya dapat dilihat pada Tabel 2.3.

Tabel 2.3 *Class Diagram*

Simbol	Nama	Keterangan
	Kelas	Kelas pada struktur sistem.
	<i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
	<i>Association</i>	Relasi antar <i>class</i> dengan arti umum, asosiasi biasanya juga disertai dengan <i>Multiplicity</i> .
	<i>Directed Association</i>	Relasi antarkelas dengan makna kelas yang atau digunakan oleh kelas yang lain,

		asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
	Generalisasi	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).
	<i>Dependency</i>	Relasi antarkelas dengan makna 'kebergantungan antarkelas'
	<i>Aggregation</i>	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>)

d) Entity Relationship Data

Komponen yang terdapat pada Entity Relationship Diagram yaitu sebagai berikut :¹¹

1. Entitas

Entitas merupakan suatu objek yang dapat dibedakan dari lainnya yang dapat diwujudkan dalam basis data. Objek dasar dapat berupa orang, benda, atau hal yang keterangannya perlu disimpan didalam basis data. Untuk menggambarkan sebuah entitas digunakan aturan sebagai berikut:

- Entitas dinyatakan dengan simbol persegi panjang.
- Nama entitas dituliskan didalam simbol persegi panjang.
- Nama entitas berupa kata benda, tunggal.
- Nama entitas sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

2. Atribut

Atribut merupakan keterangan-keterangan yang terkait pada sebuah entitas yang perlu disimpan dalam basis data. Atribut berfungsi sebagai penjelas pada sebuah entitas. Untuk menggambarkan atribut digunakan aturan sebagai berikut:

- Atribut digambarkan dengan simbol ellipsis.
- Nama atribut dituliskan didalam simbol ellipsis.
- Nama atribut merupakan kata benda, tunggal.
- Nama atribut sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

3. Relasi

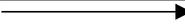
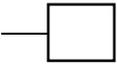
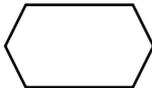
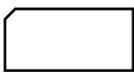
Relasi merupakan hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Aturan penggambaran relasi adalah sebagai berikut :

- Relasi dinyatakan dengan simbol belah ketupat.
- Nama relasi dituliskan didalam simbol belah ketupat.
- Nama relasi berupa kata kerja aktif.
- Nama relasi sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

2.2.8 Flowcharts

Flowchart atau bagan alir adalah penggambaran secara grafik dari langkah-langkah dan urutan-urutan prosedur dari suatu program. Tujuan *flowchart* merupakan untuk membantu menggambarkan suatu alur dalam menyelesaikan masalah dengan menggunakan simbol-simbol¹². Analisis Sistem menggunakan *flowchart program* untuk menggambarkan urutan tugas-tugas pekerjaan dalam suatu prosedur atau operasi. Berikut ini Simbol – Simbol di dalam *Flowchart* dapat dilihat pada Tabel 2.4.

Tabel 2. 4 *Flowchart*

NO	Simbol	Fungsi
1.		Merepresentasikan <i>input</i> data atau <i>Output</i> data yang diproses atau Informasi
2.		Merepresentasikan Operasi
3.		Keluar kea tau masuk dari bagian lain Flowchart khususnya halaman yang sama
4.		Merepresentasikan alur kerja
5.		Digunakan untuk komentar tambahan
6.		Keputusan dalam Program
7.	Rincian Operasi berada ditempat lain	
8.		Pemberian harga awal
9.		Awal/akhir <i>Flowchart</i>
10.		<i>Input/Output</i> yang menggunakan kartu berlubang