

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian yang pertama dengan judul “Sistem Informasi Penjualan Kambing dan Aqiah Berbasis *E-Commerce* Pada UKM Hasanah”. Sistem ini digunakan untuk penjualan dan pemesanan aqiah yang dapat dilihat pada halaman web hasanah. Sistem ini mengelola data pesanan yang masuk, dan juga memudahkan pelanggan dalam melakukan transaksi pemesanan, serta dapat membantu pemilik dalam mengelola pemesanan secara systematis dan terkomputerisasi. Sistem ini dapat diakses oleh Admin, dan kosumen[2].

Penelitian yang kedua dengan judul “Perancangan Aplikasi Penjualan Hewan Ternak Qurban Dan Aqiqah Dengan Metode Unified Modeling Language(UML)”. Sistem ini digunakan untuk membantu konsumen dan penjual hewan Qurban dan Aqiqah dalam melakukan proses transaksi jual beli dengan mudah dan juga agar memudahkan keduanya melakukan proses[3].

Penelitian yang ketiga dengan judul “Perancangan Aplikasi Penjualan Ternak Berbasis Android (Studi Kasus: Pada Kelompok Ternak Wonosari Pacet Mojokerto)”. Aplikasi ini dibuat sebagai pengelola transaksi antara pembeli dan penjual, dan juga untuk membantu kelompok ternak Wonosari untuk menjual produk perternakan secara menyeluruh diberbagai wilayah[4].

Pada penelitian ini, penulis bermaksud membangun “Sistem Informasi Penjualan Kambing Pada Perternakan Pak Tiwan Berbasis Web”. Sistem ini diharapkan dapat membantu pihak Pak Iwan dalam memberika informasi kepada konsumen yang membutuhkan hewan kambing, sistem juga membantu konsumen dan pihak Pak Iwan dalam melakukan proses transaksi jual beli dengan mudah. Pembayaran pada sistem ini menggunakan konsep pembayaran DP.

2.2 Landasan Teori

Dalam penelitian ini perlu adanya teori-teori yang mendasar untuk menunjang proses penelitian ini, adapun teori-teori tersebut adalah:

2.2.1 Sistem Informasi

Sistem informasi adalah suatu cara tertentu untuk beroperasi dengan cara yang sukses dan untuk organisasi bisnis dengan cara yang menguntungkan. Pada dasarnya sistem informasi merupakan suatu sistem yang dibuat oleh manusia yang terdiri dari beberapa komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi. Sistem informasi di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan^[5].

Sistem informasi merupakan sebuah serangkaian prosedur dimana data dikumpulkan, diproses menjadi informasi dan didistribusikan kepada para pemakai. Sistem informasi menerima output, atau yang biasa disebut sebagai transaksi, yang kemudian dikonversi melalui berbagai proses menjadi output informasi yang akan didistribusikan kepada para pemakai informasi. Dapat disimpulkan bahwa sistem informasi itu suatu alat yang membantu dalam menyediakan informasi bagi penerimanya dan untuk membantu dalam pengambilan keputusan bagi manajemen didalam operasi perusahaan sehari-hari dan informasi yang layak untuk pihak luar perusahaan^[4]. Tujuan sistem informasia antara lain:

1. Menyediakan informasi tertentu dalam pengambilan keputusan manajemen.
2. Membantu petugas dalam melaksanakan operasional perusahaan dari hari ke hari.
3. Membantu organisasi dalam menemukan berbagai informasi tentang organisasi secara cepat menggunakan teknologi mutakhir.

2.2.2 Penjualan

Penjualan ialah aktivitas bisnis dalam menjual produk atau jasa, aktivitas penjualan ini sangat penting bagi perusahaan, terutama untuk meraih keuntungan. Sedangkan secara umum kegiatan jual beli

dijalankan oleh dua belah pihak atau lebih dengan alat pembayaran yang sah.

2.2.3 Rekayasa Perangkat Lunak

Rekayasa Perangkat Lunak (*Software Engineering*) merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin. Berikut adalah metode dan *tools* yang digunakan dalam proses sistem informasi penjualan kambing Pak Tiwan berbasis web:

A. Metode Waterfall

Metode *Waterfall* adalah suatu proses pengembangan perangkat lunak berurutan, di mana kemajuan dipandang sebagai terus mengalir ke bawah (seperti air terjun) melewati fase-fase perencanaan, pemodelan, implementasi (konstruksi), pengujian, dan pemeliharaan. Model air terjun (*waterfall*) menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut. Berikut tahapan-tahapan dari Model *Waterfall*:

1. Analisis kebutuhan perangkat lunak
Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*.
2. Desain
Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka dan prosedur pengodean.
3. Pembuatan kode program
Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program sesuai dengan desain yang telah dibuat pada tahap desain.
4. Pengujian
Pengujian fokus pada perangkat lunak secara segi *logic* dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan[6].
5. Pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Pada tahap ini dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

B. Metode Pengujian Sistem

Pengujian sistem adalah pengujian proram perangkat lunak yang lengkap dan terintegrasi. Salah satu jenis pengujian sistem yaitu Black-Box Testing yang biasa dikenal dengan sebutan pengujian fungsional yang merupakan metode pengujian software tanpa mengetahui struktur internal kode atau program. Dalam pengujian ini, tester menyadari apa yang harus dilakukan oleh program akan tetapi tidak memiliki pegetahuan bagaimana melakukannya. Metode pengujian ini memiliki beberapa kelebihan yaitu efisien untuk segmen kode besar. Akan tetapi memiliki kelemahan yaitu cakupan terbatas karena hanya sebagian kecil dari skenario [7].

2.2.4 Rekayasa Web

Rekayasa web ialah proses yang digunakan untuk menciptakan sebuah aplikasi web yang mempunyai kualitas tinggi. Rekayasa web mengadaptasi rekayasa perangkat lunak dalam konsep dasar yang menekankan aktifitas teknis dan manajemen. Akan tetapi adaotasi tidak secara utuh, dengan perubahan dan penyesuaian. Rekayasa web itu gabungan antara web publishing (suatu konsep yang berasal dari printed publishing) dan aktifitas rekayasa perangkat lunak. Dikatakan demikian desain sebuah aplikasi web menekankan pada desain grafis, desain informasi, serta desain sistem dan pemrograman^[8].

A. Tools/Alat Bantu Penelitian

1. Flowchart

Flowchart adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir (*flowchart*) digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi[9]. Ada beberapa jenis-jenis flochart diantaranya:

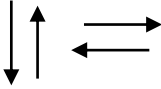



- A. Bagan alir sistem (*system flowchart*)
- B. Bagan alir dokumen (*document flowchart*)
- C. Bagan alir skematik (*schematic flowchart*)

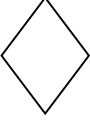


D. Bagan alir program (*program flowchart*)

E. Bagan alir proses (*process flowchart*)

Dibawah ini merupakan simbol-simbol dalam *flowchart* dapat dilihat pada tabel 2.1

Tabel 2. 1 Simbol Flowchart

 <p>Flow Symbol Direction</p>	<p>Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line.</p>
 <p>Terminator Symbol</p>	<p>Yaitu simbol untuk permulaan (<i>start</i>) atau akhir (<i>end</i>) dari suatu kegiatan.</p>
 <p>Processing Symbol</p>	<p>Yaitu simbol yang menunjukkan pengolahan yang dilakukan oleh komputer.</p>
 <p>Manual Operation Symbol</p>	<p>Yaitu simbol yang menunjukkan pengolahan data yang tidak dilakukan oleh komputer.</p>

 Decision Symbol	Yaitu simbol untuk pemilihan proses berdasarkan kondisi yang ada.
 Input-Output Symbol	Yaitu simbol yang menyatakan proses input dan output tanpa tergantung dari jenis peralatannya.
 Document Symbol	Yaitu simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

2. *Unified Modeling Language (UML)*

Unified Modeling Language (UML) adalah hasil kerja dari konsorium berbagai organisasi yang berhasil dijadikan sebagai standar baku dalam *Object Oriented Analysis* dan *Design (OOAD)*. Macam-macam dari *Unified Modeling Language (UML)* antara lain[9]: *use case diagram*, *sequence diagram* dan *class diagram*.





a. *Use case*







Use Case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use Case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut *scenario* sedangkan pengguna disebut *actor*. *Actor*

adalah sebuah peran yang biasa dimainkan oleh pengguna dalam interaksinya dengan sistem. Model *use case* adalah bagian dari model *requirement*. Definisi lain *use case* dibuat berdasarkan keperluan *actor*[10].

Berdasarkan definisi diatas maka dapat disimpulkan bahwa *Use Case* adalah konstruksi untuk mendeskripsikan bagaimana sistem akan terlihat dimata pengguna potensial yang terdiri dari sekumpulan *scenario* dan *actor*. Sedangkan *use case* diagram memfasilitasi komunikasi diantara analisis dan pengguna serta analisis dan klien[10]. *Use Case* dapat dilihat pada tabel 2.2.

Tabel 2. 2 Use Case Diagram

No.	Simbol	Nama Simbol dan Keterangan
1.		<i>Actor</i> , Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2.		<i>Dependency</i> , Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri(<i>independent</i>).
3.		<i>Generalization</i> , Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4.		<i>Include</i> , Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.



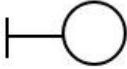



5.		<i>Extend</i> , Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6.		<i>Association</i> , yang menghubungkan antara objek satu dengan objek lainnya
7.		<i>System</i> , Menspesifikasikan paket yang menampilkan sistem secara terbatas
8.		<i>Use Case</i> , Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.
9.		<i>Collaboration</i> , Interaksi aturan-aturan dan elemen lain yang bekerjasama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10.		<i>Note</i> , Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

b. *Sequence Diagram*

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukkan sejumlah objek dan *message* yang diletakan antara obyek-obyek didalam *use case*.

Komponen utama *sequence diagram* terdiri atas objek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical*. *Sequence Diagram* menambahkan dimensi waktu pada interaksi diantara obyek[11]. Simbol-simbol yang dipakai dalam pembuatan *sequence diagram* dapat dilihat pada Tabel 2.3.

Tabel 2. 3 Simbol Squence

GAMBAR	NAMA	KETERANGAN
	<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sisitem.
	<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan
	<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari foem
	<i>Control Class</i>	Menggambarkan penghubung antara boundary dengan tabel
	<i>A focus of Control & A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya message
	<i>A message</i>	Menggambarkan Pengiriman Pesan

c. **Class Diagram**

Class Diagram membantu dalam visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak. *Class diagram* memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain (dalam *logical view*) dari suatu sistem. Selama proses analisis, *class diagram* memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama proses analisis, *class diagram* memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama tahap decian, *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat[11].

3. **Entity Relationship Diagram (ERD)**

Entity Relationship Diagram (ERD) merupakan suatu model jaringan yang menggunakan susunan data yang disimpan pada sistem secara abstrak. ERD menggambarkan hubungan antar satu entitas yang memiliki sejumlah atribut dengan entitas lain dalam suatu sistem yang tertegrasi ERD terdiri dari 3 komponen yaitu:

1. Entitas (*entity*)

Entitas (*entity*) merupakan objek-objek dasar yang terkait di dalam sistem. Objek dasar tersebut dapat berupa orang, benda atau hal lain yang keterangannya perlu disimpan di dalam basis data. Entitas dinyatakan dengan simbol persegi panjang.

2. Atribut (*attribute*)

Atribut (*attribute*) atau sering disebut dengan properti merupakan keterangan-keterangan yang terkait pada sebuah entitas yang perlu disimpan sebagai basis data. Atribut berfungsi sebagai penjelas sebuah entitas untuk menggambarkan atribut yang dilakukan. Atribut dinyatakan dengan simbol elipps.

3. Hubungan atau relasi (*relation*)

Hubungan atau relasi (*relation*) adalah kejadian atau transaksi yang terjadi diantara dua entitas yang keterangannya perlu disimpan dalam basis data. Relasi dinyatakan dalam simbol belah ketupat[10].

2.2.5 Basis Data

Basis data atau *Database* merupakan koleksi dari data yang terorganisasi dengan cara sedemikian rupa hingga data tersebut mudah disimpan dan dimanipulasi. Prinsip utama basis data adalah pengaturan data dengan tujuan utama untuk fleksibilitas dan kecepatan dalam pengambilan data kembali. Jadi, sistem basis data atau *database* adalah sistem yang terdiri dari kumpulan file atau tabel yang saling berhubungan dan memungkinkan beberapa pemakai mengakses dan memanipulasi[12].

MySQL merupakan sebuah program database server yang mampu untuk menerima serta mengirimkan data dengan cepat, multi *user* serta menggunakan perintah standar SQL (*Structured Query Language*). MySQL juga dapat didefinisikan sebagai database server, dan dapat juga berpesanan sebagai *client* yang sering disebut dengan database server dengan kemampuan dapat berjalan dengan baik di Operasi Sistem maupun dengan *Platform Windows*[11].

MySQL dikembangkan oleh sebuah perusahaan di Swedia yang bernama MySQL AB, yang pada saat itu bernama TeX Data Konsultan AB sekitar tahun 1994-1995. MySQL sudah ada sejak tahun 1979. Kepopuleran MySQL dimungkinkan karena mudah untuk digunakan, cepat secara kinerja *query*, dan juga mencukupi untuk kebutuhan basis data perusahaan skala menengah atau kecil. Sebuah basis data yang terdapat pada MySQL mengandung satu atau beberapa table yang terdiri dari sejumlah baris dan kolom[11].

MySQL merupakan turunan salah satu konsep utama dalam database yaitu SQL (*Structured Query Language*). *SQL* adalah suatu bahasa yang digunakan untuk mengakses data didalam sebuah database relasional. Dalam penggunaan *SQL* terdapat beberapa perintah yang berguna untuk mengakses data yang terdapat di dalam database^[11].

2.2.6 Website

Website merupakan istilah yang sudah tidak asing lagi, secara umum website dapat diartikan sebagai sebuah halaman yang tersedia dalam sebuah server yang dapat diakses menggunakan jaringan internet dimana didalamnya berisi bermacam-macam informasi dari suatu konten tertentu. Sebuah halaman web yang tampil

pada jejaring, umumnya dibuat melalui serangkaian plain text yang dikenal dengan istilah HTML(*Hyper Text Markup Language*)^[8].

Website terbagi menjadi 2 situs yaitu *Website* Statis dan *Website* Dinamis. Adapun *Website* Statis yang mempunyai halaman konten yang tidak berubah-ubah atau tetap. Sedangkan *Website* Dinamis itu website secara struktur diajukan untuk diupdate sesering mungkin. Adapun unsur-unsur dari website yaitu:

1. Domain: merupakan alamat yang unik di internet yang digunakan untuk mengidentifikasi sebuah website atau dengan kata lain alamat yang digunakan untuk menentukan sebuah website pada internet.
2. Hosting : memiliki peranan untuk menyimpan semua database(script, gambar, video, teks, dan lain sebagainya) yang diperlukan untuk membentuk suatu website^[10].