BAB II
TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian tentang sistem informasi sebelumnya telah dilakukan dengan judul "Sistem Informasi Penjualan Berbasis Web pada PT. Cahaya Sejahtera Sentosa Blitar" dimana kegiatan jual beli masih dilakukan secara konvensional juga pemasaran dan promosi yang masih terbatas, untuk itu diperlukan sebuah sistem informasi yang dapat memperluas pemasaran dan mempermudah transaksi penjualan sehingga proses penjualan menjadi lebih hemat, cepat, dan mudah. Sistem informasi tersebut dikembangkan dengan Bahasa pemrograman PHP[3].

Penelitian sebelumnya dengan judul "Sistem Informasi Produksi Pakaian Berbasis Desktop". Sistem informasi ini digunakan untuk mendukung proses produksi, manajemen persediaan, dan pembelian dalam perusahaan juga meminimalisir kehilangan data dan kesalahan manusia (*Human error*) yang sering terjadi karena pemrosesan dan penyimpanan transaksi data yang dilakukan secara konvensional. Sistem tersebut dikembangkan dengan metode *Waterfall* dan metode pengujian *Blackbox*[4].

Penelitian sebelumnya berjudul "Sistem Informasi E-Commerce Berbasis Web Pada Toko Indonesia Okuba Jepang" yang dibuat untuk menyelesaikan permasalahan sistem jual beli di Toko Indonesia Okuba Jepang yang belum efektif dimana customer tidak mengetahui stok apa saja yang tersedia di Toko Indonesia Okuba Jepang karena kurangnya informasi. Sistem ini dikembangkan dengan metode *Waterfall* menggunakan Bahasa pemrograman PHP dan *database MySQL*[5].

Penelitian sebelumnya dengan judul "Perancangan Sistem Informasi Berbasis Web Pemesanan Barang pada Rumah Cetak Merdeka (RCM) *Digital Printing* Padang" ditujukan untuk memanajemen pelayanan agar lebih akurat, efektif dan efisien pada Rumah Cetak Merdeka (RCM) *digital printing* Padang karena karyawan masih kewalahan melayani permintaan pemesanan produk maupun manajemen data pemesanan dengan jumlah karyawan yang bertugas sangat terbatas. Sistem ini dikembangkan dengan Bahasa pemrograman PHP, *framework CodeIgniter*, Apache untuk *web server*, dan *MySQL* sebagai *database*[6].

Pada penelitian ini penulis bermaksud merancang sistem informasi penjualan Kelompok Wanita Tani Sekar Asri dengan menggunakan metode *waterfall, framework CodeIgniter* (CI), Bahasa pemrograman PHP, dan *database MySQL*. Yang berbeda dari penelitian sebelumnya pada sistem ini juga akan disediakan fitur update stok rutin setiap seminggu sekali sehingga customer bisa mengetahui hasil panen tanpa perlu mengecek website setiap hari. Sistem informasi ini diharapkan dapat digunakan baik oleh customer maupun pengurus Kelompok Wanita Tani Sekar Asri.

2.2 Landasan Teori

Dalam penelitian ini diperlukan adanya teoti-teori yang mendasar untuk menunjang proses penelitian ini. Teori-teori yang digunakan dalam penelitian ini adalah

2.2.1 Sistem

Sistem merupakan suatu unsur atau elemen yang saling berkaitan satu sama lain dan saling mempengaruhi dalam melakukan kegiatan Bersama untuk mencapai suatu tujuan[7].

2.2.2 Informasi

Menurut Anggraeni dan Irivani (2017:13) “Informasi adalah sekumpulan data atau fakta yang diorganisasi atau diolah dengan cara tertentu sehingga memiliki arti bagi si penerima”[8].

Selanjutnya menurut Trimahardika dan Sutinah (2017:250), “Informasi merupakan suatu data yang telah diolah, diklasifikasikan dan diinterpretasikan serta digunakan untuk proses pengambilan keputusan[9].

Berdasarkan definisi diatas, disimpulkan bahwa Informasi merupakan hal mendasar yang sangat diperlukan oleh suatu kegiatan dalam pengambilan keputusan untuk menghindari terjadinya sebuah kesalahan.

2.2.3 Sistem Informasi

Sistem Informasi adalah suatu sistem yang terdiri dari kumpulan komponen sistem, yaitu *software, hardware,* dan *brainware* yang memproses informasi menjadi sebuah *output* yang berguna untuk mencapai suatu tujuan tertentu dalam suatu organisasi[10].

2.2.4 Penjualan

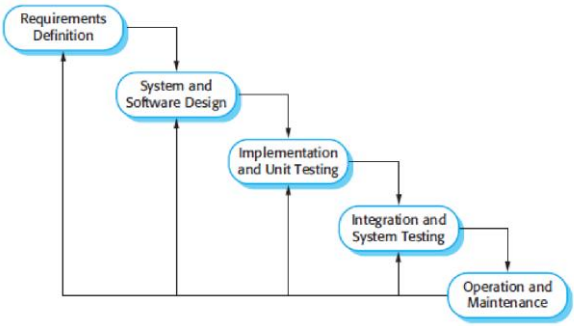
Penjualan adalah proses sosial manajerial dimana individu dan kelompok mendapatkan apa yang mereka butuhkan dan inginkan,

menciptakan, menawarkan, dan mempertukarkan produk yang bernilai dengan pihak lain[11].

2.2.5 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak (*software engineering*) merupakan pembangunan dengan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin. Perangkat lunak sering dibuat namun banyak yang pada akhirnya tidak memenuhi kebutuhan customer atau bahkan terjadi masalah *non*-teknis seperti keengganan pengguna perangkat lunak (*user*) untuk mengubah cara kerja dari konvensional ke otomatis, atau ketidakmampuan *user* dalam menggunakan computer. Oleh karena itu, rekayasa perangkat lunak dibutuhkan agar perangkat lunak yang dibuat tidak hanya menjadi perangkat lunak yang tidak terpakai[12].

2.2.6 Waterfall



Gambar 2. 1 Metode *Waterfall* oleh Sommerville

Menurut Sommerville (2011) *Waterfall* model adalah sebuah contoh dari proses perencanaan, dimana semua proses kegiatan harus terlebih dahulu direncanakan dan dijadwalkan sebelum dikerjakan. Penggunaan model *Waterfall* dalam pengembangan sistem diharapkan mampu memudahkan pembuatan sehingga pembangunan sistem bisa terstruktur[13].

Tahapan metode *Waterfall* menurut Sommerville sebagai berikut :

a. *Requirement Analysis and Definition*

Tahap ini lebih difokuskan kepada penetapan tujuan dari sistem yang ditetapkan oleh konsultasi. Kemudian ditetapkan juga secara detail spesifikasi sistem.

b. *System and Software Design*
Proses desain sistem yang membutuhkan perangkat keras atau perangkat lunak dengan membentuk sistem secara keseluruhan arsitektur. Desain perangkat lunak juga melibatkan identifikasi dan abstraksi sistem perangkat lunak.

c. *Implementation and Unit Testing*
Tahap ini mengimplementasikan hasil desain ke dalam kode atau bahasa pemrograman. Sedangkan pengujian melibatkan verifikasi yang memastikan setiap unit sistem memenuhi spesifikasinya.

d. *Integration and System Testing*
Tahap ini dilakukan pengujian program setelah kode dihasilkan sebagai sebuah sistem lengkap untuk memastikan bahwa perangkat lunak telah memenuhi persyaratan.

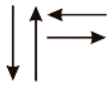
e. *Operation and Maintenance*
Tahap ini merupakan tahap terakhir dimana dilakukan pemeliharaan pada sistem dengan tujuan melakukan penyesuaian dan perbaikan.

2.2.7 Flowchart

Flowchart adalah bagan bagan yang mempunyai arus yang menggambarkan Langkah-langkah penyelesaian suatu masalah. *Flowchart* juga merupakan sebuah cara untuk menyajikan algoritma. Berikut beberapa symbol dalam *flowchart*[14], yang dimuat dalam tabel 2.1

Tabel 3. 1 Simbol-Simbol dalam *Flowchart*

No.	Simbol	Nama	Keterangan
1.		*Terminal*	Memulai & mengakhiri suatu program.
2.		*Input/ Output*	Memasukan data dan mennjukkan hasil dari suatu proses tanpa bergantung pada jenis peralatan.

3.		*Flow*	Menghubungkan antara symbol yang satu dengan yang lain, dengan kata lain menyatakan jalannya suatu proses (disebut juga sebagai *connecting line*).
4.		*Decision*	Memilih proses berdasar pada kondisi yang tersedia.
5.		*Document*	Merupakan simbol yang menggambarkan data yang berbentuk informasi.
6.		*Manual Operation*	Menunjukkan pengolahan yang tidak dilakukan oleh komputer.
7.		*Manual Input*	Memasukkan (*input*) data secara manual dengan keyboard.

2.2.8 UML

UML (*Unified Modeling Language*) merupakan salah satu standar Bahasa yang banyak digunakan di dunia industry untuk mendefinisikan *requirement*, membuat analisis dan desain serta menggambarkan arsitektur dalam pemrograman berorientasi objek[15].

1. Use Case Diagram

Use Case Diagram merupakan pemodelan untuk *behavior* sistem informasi yang akan dibuat. Secara garis besar *use case diagram* digunakan untuk mengetahui fungsi apa saja yang ada dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu[15]. Simbol-simbol yang ada pada *use case diagram* diantaranya sebagai berikut, pada tabel 2.2.

Tabel 3. 2 Simbol-simbol *Use Case Diagram*

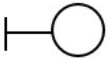
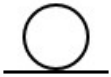
No.	Simbol	Nama	Keterangan

1.		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil terukur bagi *Actor*.
2.		*Actor*	Membuat himpunan peran yang *user* mainkan Ketika berinteraksi dengan *use case* menjadi lebih spesifik.
3.		*Association*	Menghubungkan objek satu dengan yang lainnya.
4.		*Include*	Menunjukkan *use case* sumber secara eksplisit dan spesifik.
5.		*Extend*	Menunjukkan secara spesifik bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
6.		*System*	Menunjukan dengan spesifik paket yang menampilkan sistem seara terbatas.

2. Sequence Diagram

Sequence Diagram digunakan untuk menggambarkan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan maupun diterima antar objek. Oleh karena itu untuk menggambarkan *sequence diagram* maka harus diketahui terlebih dahulu objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu[14]. Simbol-simbol pada *sequence diagram* diantaranya tertera pada Tabel 2.3.

Tabel 3. 3 Simbol-Simbol *Sequence Diagram*

No.	Simbol	Nama	Keterangan
1.		*LifeLine*	Objek entitas yang saling berinteraksi.
2.		*Actor*	Menggambarkan user yang menggunakan sistem.
3.		*Message*	Menggambarkan komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi.
4.		*Boundary*	Menggambarkan sebuah form.
5.		*Control Class*	Menghubungkan *boundary* dengan Tabel.
6.		*Entity Class*	Menggambarkan hubungan kegiatan yang akan dilakukan.

3. Class Diagram

Class diagram adalah jenis diagram struktur statis dalam UML yang menggambarkan struktur sistem dengan menunjukkan sistem *class*, atributnya, metode, dan hubungan antar objek. *Class diagram* disebut jenis diagram struktur karena menggambarkan apa yang harus ada dalam sistem yang dimodelkan dengan berbagai komponen. Berbagai komponen tersebut dapat mewakili *class* yang akan diprogram, objek utama, atau interaksi antara *class* dan objek[19]. Berikut adalah komponen-komponen yang menyusun *class diagram* :

1. Komponen atas

Bagian ini berisikan nama *class* yang selalu diperlukan baik itu dalam pengklasifikasi atau objek.

2. Komponen tengah

   Komponen ini berisikan atribut *class* yang digunakan untuk mendeskripsikan kualitas kelas. Ini hanya diperlukan saat mendeskripsikan instansi tertentu dari sebuah *class*.

3. Komponen bawah

   Bagian ini adalah komponen *class diagram* yang menyertakan operasi *class* yang ditampilkan dalam format daftar. Sehingga, setiap operasi mengambil barisnya sendiri.

2.2.9 PHP

PHP (*Hypertext Preprocessor*) merupakan suatu Bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang bersifat *server-side* dan dapat ditambahkan kedalam HTML[16].

2.2.10 Basis Data

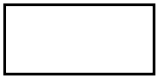
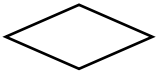
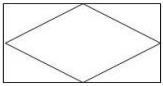
Basis data adalah kumpulan data yang disimpan secara sistematis dalam komputer dan dapat diolah atau dimanipulasi menggunakan perangkat lunak untuk menghasilkan informasi. Definisi basis data meliputi spesifikasiberupa tipe data, struktur, dan juga batasan-batasan data yang akan disimpan[17].

Basis data merupakan aspek yang sangat penting dalam sistem informasi dimana basis data merupakan gudang penyimpanan data yang akan diolah lebih lanjut. Basis data juga digunakan untuk menghindari duplikasi data, hubungan antar data yang tidak jelas, organisasi data, dan juga update data yang rumit[17].

2.2.11 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) [18] merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh System Analys dalam tahap analisis persyaratan proyek pengembangan system. Sementara seolah-olah teknik diagram atau alat peraga memberikan dasar untuk desain database relasional yang mendasari sistem informasi yang dikembangkan. ERD bersama-sama dengan detail pendukung merupakan model data yang pada gilirannya digunakan sebagai spesifikasi untuk database.

Tabel 3. 4 Simbol- Simbol ERD

No.	Simbol	Nama
1.		Entitas
2.		Atribut
3.		Hubungan
4.		Garis
5.		Entitas Asosiatif

1) Entitas
   Entity (entitas) yaitu suatu obyek yang dapat dibedakan dari lainnya yang dapat diwujudkan dalam basis data.
2) Hubungan (relasi/*relationship*)
   Suatu hubungan adalah hubungan antara dua jenis entitas dan direpresentasikan sebagai garis lurus yang menghubungkan dua entitas.
3) Atribut
   Atribut memberikan informasi lebih rinci tentang jenis entitas. Atribut memiliki struktur internal berupa tipe data.
   Jenis-jenis atribut:
   a. Atribut *key*
      Atribut *key* adalah satu atau gabungan dari beberapa atribut yang dapat membedakan semua baris data (*Row/Record*) dalam tabel secara unik. Dikatakan unik jika pada atribut yang dijadikan *key* tidak boleh ada baris data dengan nilai yang sama.
   b. Atribut *simple*

Atribut yang bernilai atomic, tidak dapat dipecah/ dipilah lagi.

c. Atribut *multivalue*
Nilai dari suatu attribute yang mempunyai lebih dari satu (*multivalue*) nilai dari atrribute yang bersangkutan.

d. Atribut *composite*
Atribut *composite* adalah suatu atribut yang terdiri dari beberapa atribut yang lebih kecil yang mempunyai arti tertentu yang masih bisah dipecah lagi atau mempunyai sub attribute.

e. Atribut *derivatif*
Atribut yang tidak harus disimpan dalam database Ex. Total. atau atribut yang dihasilkan dari atribut lain atau dari suatu *relationship*. Atribut ini dilambangkan dengan bentuk oval yang bergaris putus-putus.

Derjat relasi atau kardinalitas rasio :

1) *One to one* (1:1)
Setiap anggota entitas A hanya boleh berhubungan dengan satu anggota entitas B, begitu pula sebaliknya.
2) *One to many* (1:M/*Many*)
Setiap anggota entitas A dapat berhubungan dengan lebih dari satu anggota entitas B tetapi tidak sebaliknya. Setiap anggota entitas A dapat berhubungan dengan lebih dari satu anggota entitas B tetapi tidak sebaliknya.
3) *Many to many* (M:M)
Setiap entitas A dapat berhubungan dengan banyak entitas himpunan entitas B dan demikian pula sebaliknya.