

## BAB II

### LANDASAN TEORI

#### 2.1. Tinjauan Pustaka

##### A. Konsep Dasar Sistem Informasi

Menurut Rosa dan Shalahuddin (2013:2) sistem adalah kumpulan komponen yang saling berkaitan dan mempunyai satu tujuan yang ingin dicapai. Sedangkan menurut Hartono (2005:1) “sistem adalah suatu jaringan kerja dan prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu”.

Sistem mempunyai karakteristik atau sifat-sifat tertentu seperti elemen-elemen. Menurut Mulyanto (2009:2), dalam bukunya Sistem Informasi Konsep dan Aplikasi:

1. Mempunyai Komponen Sistem (*Components Sistem*)

Suatu sistem tidak berada dalam lingkungan yang kosong, tetapi sebuah sistem berada dan berfungsi di dalam lingkungan yang berisi sistem lainnya. Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Apabila suatu sistem merupakan salah satu dari komponen sistem lain yang lebih besar, maka akan disebut dengan subsistem, sedangkan sistem yang lebih besar tersebut adalah lingkungannya.

2. Mempunyai Batasan Sistem (*Boundary*)

Batas sistem merupakan pembatas atau pemisah antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

3. Mempunyai Lingkungan (*Environment*)

Lingkungan luar adalah apa pun di luar batas dari sistem yang dapat mempengaruhi operasi sistem, baik pengaruh yang menguntungkan ataupun yang merugikan. Pengaruh yang menguntungkan ini tentunya harus dijaga sehingga akan mendukung kelangsungan operasi sebuah sistem. Sedangkan lingkungan yang merugikan harus ditahan dan dikendalikan agar tidak mengganggu kelangsungan sebuah sistem.

4. Mempunyai Penghubung (*Interface*) Antar Komponen

Penghubung (*Interface*) merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Penghubung inilah yang akan menjadi media yang digunakan data dari masukan (*input*) hingga keluaran (*output*). Dengan adanya penghubung, suatu subsistem dapat berinteraksi dan berintegrasi dengan subsistem yang lain membentuk satu kesatuan.

5. Mempunyai Masukan (*Input*)

Masukan atau *input* merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*), yaitu bahan yang dimasukkan agar sistem tersebut dapat beroperasi dan masukan sinyal (*signal input*), yaitu masukan yang diproses untuk mendapatkan keluaran.

6. Mempunyai Pengolahan (*Processing*)

Pengolahan (*process*) merupakan bagian yang melakukan perubahan dari masukan untuk menjadi keluaran yang diinginkan.

7. Mempunyai Sasaran (*Objective*) dan Tujuan

Suatu sistem pasti memiliki sasaran (*objective*) atau tujuan (*goal*). Apabila sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Tujuan inilah yang mengarahkan suatu sistem. Tanpa adanya tujuan, sistem menjadi tidak terarah dan terkendali.

8. Mempunyai Keluaran (*Output*)

Keluaran (*output*) merupakan hasil dari pemrosesan. Keluaran dapat berupa informasi sebagai masukan pada sistem lain atau hanya sebagai sisa pembuangan.

9. Mempunyai Umpan Balik (*Feed Back*)

Umpan balik diperlukan oleh bagian kendali (*Control*) sistem untuk mengecek terjadinya penyimpangan proses dalam sistem dan mengembalikannya ke dalam kondisi normal

Menurut Mulyanto (2009:8) dalam bukunya yang berjudul Sistem Informasi Konsep dan Aplikasi .Sistem dapat di klasifikasikan berbagai sudut pandang, diantaranya adalah sebagai berikut:

1. Sistem abstrak (*abstract system*) dan sistem fisik (*physical system*).

Sistem abstrak (*abstract system*) adalah sistem yang berupa pemikiran atau gagasan yang tidak tampak secara fisik. Sedangkan sistem fisik (*physical system*) adalah sistem yang ada secara fisik dan dapat dilihat dengan mata.

2. Sistem alamiah (*natural system*) dan sistem buatan manusia (*human made system*).

Sistem alamiah adalah sistem yang keberadaannya terjadi karena proses alam, bukan buatan manusia. Sedangkan sistem buatan manusia (*human*

*made systems*) adalah sistem yang terjadi melalui rancangan atau campur tangan manusia.

3. Sistem tertentu (*deterministic system*) dan sistem tidak tentu (*probabilistic system*)

Sistem tertentu (*deterministic systems*) yaitu sistem yang operasinya dapat diprediksi secara cepat dan interaksi diantara bagian-bagiannya dapat dideteksi dengan pasti. Sedangkan sistem tidak tentu (*probabilistic systems*) yaitu sistem yang hasilnya tidak dapat diprediksi karena mengandung unsur probabilitas.

4. Sistem tertutup (*closed system*) dan sistem terbuka (*open system*).

Sistem tertutup (*closed systems*) yaitu sistem yang tidak berhubungan dengan lingkungan di luar sistem. Sistem ini tidak berinteraksi dan tidak dipengaruhi oleh lingkungan luar. Sistem ini juga bekerja secara otomatis tanpa adanya campur tangan dari pihak luar. Dalam kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah sistem yang relatif tertutup (*relative closed system*). Sistem relatif tertutup biasanya mempunyai masukan dan keluaran yang tertentu serta tidak terpengaruh oleh keadaan di luar sistem. Sedangkan sistem terbuka (*open system*) adalah sistem yang berhubungan dengan lingkungan luar dan dapat terpengaruh dengan keadaan lingkungan luar. Sistem terbuka menerima masukan dari subsistem lain dan menghasilkan keluaran untuk subsistem lain. Sistem ini mampu beradaptasi dan memiliki sistem pengendalian yang baik karena lingkungan luar yang bersifat merugikan dapat mengganggu jalannya proses di dalam sistem.

Sementara itu, menurut Sutabri (2012:29), Informasi adalah data yang telah diklarifikasi atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan yang berarti bagi penerimanya dan bermanfaat dalam mengambil keputusan saat ini atau mendatang. Kemudian Hartono (2005:8) mendefinisikan Informasi sebagai data yang diolah menjadi bentuk yang berarti bagi penerimanya.

Dari kedua definisi tersebut dapat disimpulkan bahwa informasi merupakan kumpulan atau himpunan data yang diolah menjadi bentuk yang lebih berguna bagi yang menerimanya. Dalam hal ini informasi yang dihasilkan memberikan pengetahuan penting bagi penerimanya untuk menunjang suatu keputusan. Data merupakan bentuk yang masih mentah yang belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut. Data diolah suatu model untuk dihasilkan menjadi informasi.

Menurut Sutabri (2012:46) sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, yang mendukung fungsi operasi organisasi bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang di perlukan

Menurut Sutabri (2012:41) suatu informasi dapat dikatakan memiliki manfaat dalam proses pengembangan keputusan apabila informasi tersebut mempunyai kualitas dan nilai. Kriteria kualitas informasi adalah:

1. Akurat, yang berarti informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan.

2. Tepat waktunya, yang berarti informasi yang sampai kepada penerima tidak boleh terlambat. Mahalnya nilai informasi saat ini adalah karena harus cepatnya informasi tersebut didapatkan, sehingga diperlukan teknologi mutakhir untuk mendapatkan, mengolah, dan mengirimkan.
3. Relevan, yang berarti informasi harus mempunyai manfaat bagi pihak yang menerimanya.

Menurut Sutabri (2012:40) Suatu informasi yang tidak berkualitas atau tidak dapat disebabkan oleh beberapa hal berikut ini:

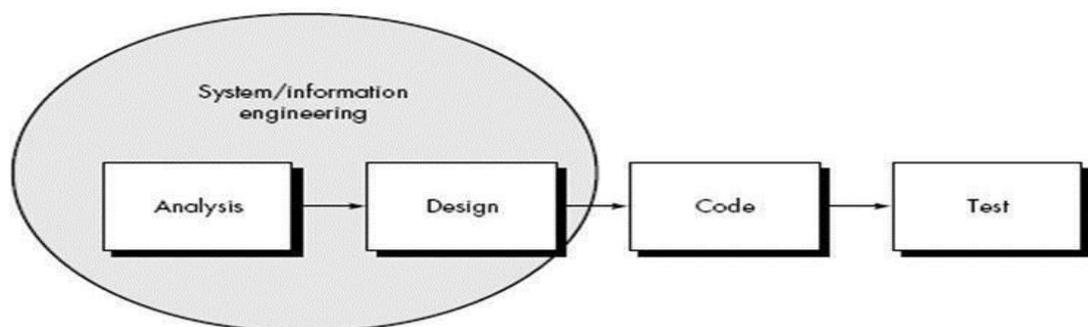
1. Metode pengukuran dan pengumpulan data yang salah.
2. Tidak mengikuti prosedur pengolahan data yang benar.
3. Data hilang atau tidak terolah.
4. Kesalahan mencatat atau mengoreksi data.
5. File historis atau induk yang salah (atau keliru memilih file historis).
6. Kesalahan dalam prosedur pengolahan (misal kesalahan program komputer).

Menurut Sutabri (2012:40) Kesulitan menghadapi kesalahan dapat diatasi dengan:

1. Kontrol interen untuk menemukan masalah.
2. Pemeriksaan interen dan eksteren.
3. Penambahan “batas kepercayaan” kepada data.
4. Instruksi pemakai dalam prosedur pengukuran dan pengolahan agar para pemakai dapat menilai kesalahan – kesalahan yang mungkin terjadi.

Metode yang paling umum digunakan adalah dengan siklus hidup pengembangan sistem (*system development life cycle – SDLC*). Metode SDLC adalah proses pengembangan atau mengubah suatu sistem perangkat lunak dengan

menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya. Hal terpenting adalah mengenali tipe pelanggan (*customer*) dan memilih menggunakan model SDLC yang sesuai dengan karakter pelanggan (*customer*) dan sesuai karakter pengembang. Metode SDLC menggunakan pendekatan sistem yang disebut pendekatan air terjun (*waterfall approach*), yang menggunakan tahapan pengembangan sistem. Model ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ketahap *analysis*, *design*, *coding*, *testing*, dan *support atau maintenance*. Sesuai dengan namanya *waterfall* (air terjun) maka tahapan dalam model ini disusun bertingkat, setiap tahap dalam model ini dilakukan berurutan, seperti gambar dibawah berikut (Gambar II.1). Jadi jika langkah satu belum dikerjakan maka tidak akan bisa melakukan pengerjaan langkah berikutnya.



**Gambar II.1**

**Model Waterfall**

Sumber: Rosa dan Shalahuddin (2013:29)

Menurut Rosa dan Shalahuddin (2013:28), Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*), model air terjun menyediakan pendekatan alur

hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung (*support*). Secara garis besar metode *waterfall* mempunyai langkah-langkah sebagai berikut:

1. Analisa Kebutuhan (*Requirment Analysis*)

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*, Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2. Desain

Desain perangkat lunak adalah proses multi langkah yang focus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3. *Coding*

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. *Testing*

Pengujian fokus pada perangkat lunak secara dari segi logik 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.

#### 5. *Support atau Maintenance*

Pada tahapan ini tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah diberikan kepada *user*. Perubahan ini bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru.

Tahap pendukung atau *maintenance* dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada tanpa membuat perangkat lunak yang baru.

### **B. E-Commerce**

#### 1. Pengertian E-Commerce

Perdagangan Elektronik (E-Commerce = *electronic commerce*) adalah bagian dari *e-lifestyle* yang memungkinkan transaksi jual beli dilakukan secara online dari sudut tempat mana pun (Hidayat, 2008:5).

E-Commerce juga dapat diartikan sebagai suatu proses berbisnis dengan menggunakan teknologi elektronik yang menghubungkan antara perusahaan, konsumen dan masyarakat dalam bentuk transaksi elektronik dan pertukaran/penjualan barang, servis, dan informasi secara elektronik (Munawar, 2009:1).

Sedangkan pengertian E-Commerce (Perdagangan Elektronik) menurut Wong (2010:33) adalah pembelian, penjualan dan pemasaran barang serta jasa melalui sistem elektronik. Seperti televisi, radio dan jaringan komputer atau internet.

#### 2. Jenis-Jenis E-Commerce

Ada banyak cara untuk mengklasifikasikan transaksi E-Commerce. Salah satunya dengan melihat sifat peserta yang terlibat dalam transaksi e-Commerce. Berdasarkan sifat penggunaannya, E-Commerce dibagi menjadi 3 jenis (Laudon, 2007:45):

- a. E-Commerce bisnis ke konsumen (B2C) melibatkan penjualan produk dan layanan secara eceran kepada pembeli perorangan.
- b. E-Commerce bisnis ke bisnis (B2B) melibatkan penjualan produk dan layanan antar perusahaan.
- c. E-Commerce konsumen ke konsumen (C2C) melibatkan konsumen yang menjual secara langsung ke konsumen.

### 3. Komponen E-Commerce

E-Commerce memiliki beberapa komponen standar yang dimiliki dan tidak dimiliki transaksi bisnis yang dilakukan secara offline, yaitu (Hidayat, 2008:7):

1. Produk: Banyak jenis produk yang bisa dijual melalui internet seperti komputer, buku, musik, pakaian, mainan, dan lain-lain.
2. Tempat menjual produk (*a place to sell*): tempat menjual adalah internet yang berarti harus memiliki domain dan hosting.
3. Cara menerima pesanan: email, telpn, sms dan lain-lain.
4. Cara pembayaran: Cash, cek, bankdraft, kartu kredit, internet payment (misalnya paypal).

5. Metode pengiriman: pengiriman bisa dilakukan melalui paket, salesman, atau didownload jika produk yang dijual memungkinkan untuk itu (misalnya software).
6. *Customer service*: email, formulir on-line, FAQ, telpon, chatting, dan lain-lain.

### C. Website

#### 1. *Internet*

Definisi *internet* menurut Darma dkk (2009:1) mengemukakan bahwa “secara harfiah, *internet* (kependekan dari ‘*interconnected-networking*’) ialah rangkaian komputer yang terhubung satu sama lain”.

Menurut Utomo dan Syafrudin (2008:9) mengemukakan bahwa “*Internet* (*inter-network*) merupakan jaringan yang menggabungkan beberapa komputer yang terhubung dalam sebuah *internet protocol* (*IP*) yang mencakup secara luas ke seluruh dunia”.

#### 2. *Website*

*Website* merupakan sebuah cara untuk menampilkan informasi di *internet*. Segala macam bentuk informasi yang tersedia seperti pendidikan, budaya, perdagangan, media sosial dan sebagainya, dapat ditampilkan oleh sebuah *website*.

Menurut Hidayat (2010:2) menyimpulkan bahwa:

*Website* atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis

maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman.

### 3. *Web Server*

Menurut Supardi (2010:2) mengemukakan bahwa “*Web server* merupakan perangkat lunak yang mengelola (mengatur) permintaan *user* dari *browser* dan hasilnya dikembalikan kembali ke *browser*. Contoh *web server* adalah *IIS (Internet Information Services)* produk *Microsoft Corp*”.

### 4. *Web Browser*

Menurut Oktavian (2010:13) mengemukakan bahwa “*Web browser* adalah program komputer yang digunakan untuk membaca *HTML*, kemudian menerjemahkan dan menampilkan hasilnya secara visual ke layar komputer”.

## **D. Bahasa Pemrograman**

### 1. *HTML (Hypertext Markup Language)*

Menurut Oktavian (2010:13) mengemukakan bahwa “*HTML* adalah suatu bahasa yang dikenali oleh *web browser* untuk menampilkan informasi dengan lebih menarik dibandingkan dengan tulisan teks biasa (*plain text*)”.

Sedangkan menurut Winarno dan Zaki (2011:1) mengemukakan bahwa “*HTML* merupakan singkatan dari *Hypertext Markup Language*, artinya bahasa ini adalah bahasa *markup* untuk memformat halaman *web*. Dengan kata lain, bahasa untuk mengatur bagaimana penampilan dan pemformatan konten di *web*”.

### 2. *PHP (PHP Hypertext Preprocessor)*

Menurut Kadir (2008:358) mengemukakan bahwa “*PHP* merupakan bahasa pemrograman skrip yang diletakan dalam *server* yang biasa digunakan

untuk membuat aplikasi *web* yang bersifat dinamis. *PHP* mendukung berbagai *database*, termasuk yang didukungnya adalah *MySQL*”.

*PHP* pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu *PHP* masih bernama *FI (Form Interpreted)*, yang wujudnya berupa sekumpulan *script* yang digunakan untuk mengolah data *form* dari *web* dinamis.

### 3. *CSS (Cascading Style Sheet)*

Menurut Riyanto (2009:236) mengemukakan bahwa “*Cascading Style Sheet (CSS)* merupakan salah satu bahasa pemrograman *web* untuk mengendalikan beberapa komponen dalam sebuah *web* sehingga akan lebih terstruktur dan seragam”.

*CSS* dapat mengendalikan gambar, warna bagian tubuh pada teks, warna tabel, ukuran *border*, warna *border*, warna *hyperlink*, warna *mouse over*, spasi antar paragraf, spasi antar teks, *margin* kiri, kanan, atas, bawah, dan parameter lainnya. *CSS* adalah bahasa *style sheet* yang digunakan untuk mengatur tampilan dokumen. Dengan adanya *CSS* memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda.

### 4. *JavaScript*

Menurut Flanagan (2011:1) menyimpulkan bahwa “*JavaScript* adalah bahasa pemrograman *web*. Mayoritas situs *web modern* telah menggunakan *JavaScript*, dan semua *web browser modern* di komputer, *game console*, *tablet*, ponsel pintar, dan termasuk *interpreter JavaScript*”.

Beberapa kelebihan dan kekurangan dari penggunaan *JavaScript* menurut Wahana Komputer (2010:5), sebagai berikut:

#### a. Kelebihan *JavaScript*

- 1) Lebih praktis dan mudah sebab bahasa pemrograman *JavaScript* memiliki sedikit sintaks.
- 2) Koneksi cepat sebab peletakan program terdapat di sisi *client*, berukuran *file* sangat kecil, dan dapat langsung dijalankan di *browser*

b. Kekurangan *JavaScript*

- 1) Pengelolaan objek dalam *JavaScript* sangat terbatas
- 2) Penggunaan *script* dapat di-*copy* langsung melalui sebuah *web browser*, sehingga setiap orang dapat menggunakan program *JavaScript* yang telah dibuat.

## 5. *JQuery*

Menurut Wicaksono (2011:1) mengemukakan bahwa “*JQuery* adalah *library* atau kumpulan kode *Javascript* siap pakai. Keunggulan menggunakan *JQuery* apabila dibandingkan dengan *JavaScript* standar, yaitu menyederhanakan kode *Javascript* dengan cara memanggil fungsi-fungsi yang disediakan oleh *JQuery*”.

## E. Basis Data

### 1. Pengertian Basis Data

Menurut Kustiyahningsih (2010:145) menyimpulkan bahwa “Dalam arti umum basis data adalah sekumpulan data yang diproses dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan tepat, yang dapat digambarkan sebagai aktivitas dari satu atau lebih organisasi yang berelasi”.

### 2. *MySQL*

Menurut Kadir (2008:2) mengemukakan bahwa “*MySQL* merupakan *software* yang tergolong sebagai *DBMS (Database Management System)* yang

bersifat *Open Source*. *Open source* menyatakan bahwa *software* ini dilengkapi dengan *source code* (Kode yang dipakai untuk membuat *MySQL*)”.

Beberapa kelebihan *MySQL* menurut Anhar (2010:22), antara lain:

- a. *MySQL* dapat berjalan dengan stabil pada berbagai sistem operasi, seperti *Windows, Linux, FreeBSD, Mac Os X Server, Solaris* dan masih banyak lagi.
- b. Bersifat *Open Source*, *MySQL* didistribusikan secara *open source* (gratis), di bawah lisensi *GNU General Public License (GPL)*.
- c. Bersifat *Multiuser*, *MySQL* dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah.
- d. *MySQL* memiliki kecepatan yang baik dalam menangani *query* (perintah *SQL*). Dengan kata lain, dapat memperoleh lebih banyak *SQL* per satuan waktu.
- e. Dari segi *security* atau keamanan data, *MySQL* memiliki beberapa lapisan sekuriti, seperti *level subnet mask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta *password* yang terenkripsi.

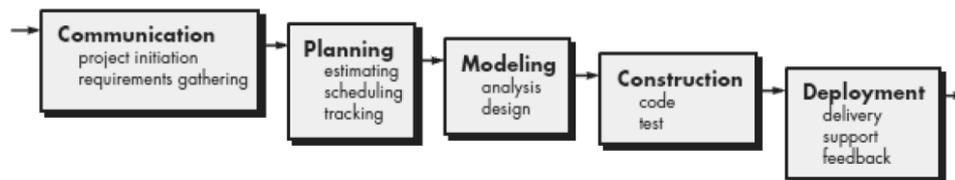
### 3. *PhpMyAdmin*

Menurut Firdaus (2007:15) mengemukakan bahwa “*PhpMyAdmin* adalah suatu program *open source* yang berbasis *web* yang dibuat menggunakan aplikasi *PHP*. Program ini digunakan untuk mengakses database *MySQL*. Program ini mempermudah dan mempersingkat kerja penggunanya”.

## **F. Model Pengembangan Perangkat Lunak**

Menurut Pressman (2010:39) mengemukakan bahwa “model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*”.

Fase-fase dalam model *waterfall* menurut Pressman (2010:39), sebagai berikut:



Sumber : Pressman (2010:39)

**Gambar II.2**

### **Konsep *Waterfall Model***

#### 1. *Communication*

Langkah ini merupakan analisis terhadap kebutuhan perangkat lunak, dan tahap untuk mengadakan pengumpulan data dengan melakukan pertemuan *customer*, maupun mengumpulkan data-data tambahan baik yang ada di jurnal, artikel, maupun dari *internet*.

#### 2. *Planning*

Proses *planning* merupakan lanjutan dari proses *communication* (*analysis requirement*) tahapan ini akan menghasilkan dokumen *user requirement* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan *software*, termasuk rencana yang dilakukan.

#### 3. *Modeling*

Proses *modeling* ini akan menerjemahkan syarat kebutuhan ke sebuah perancangan *software* yang diperkirakan sebelum *coding* dibuat. Proses ini

berfokus pada rancangan struktur data, arsitektur *software*, representasi *interface*, dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut *software requirement*.

#### 4. *Construction*

*Construction* merupakan proses membuat kode. *Coding* atau pengkodean merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh komputer. *Programmer* akan menerjemahkan transaksi yang diminta oleh *user*. Tahapan ini merupakan tahapan secara nyata dalam mengerjakan suatu *software*, artinya penggunaan komputer dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang telah dibuat. Tujuan testing adalah menemukan kesalahan-kesalahan terhadap sistem tersebut untuk kemudian bisa diperbaiki.

#### 5. *Deployment*

Tahapan ini dikatakan final dalam pembuatan sebuah *software* atau sistem. Setelah melakukan analisis, desain dan pengkodean maka sistem yang sudah jadi akan digunakan oleh *user*. Kemudian *software* yang telah dibuat harus dilakukan pemeliharaan secara berkala.

### **G. Unified Modelling Language(UML)**

“UML adalah salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.” menurut Rosa dan Shalahuddin (2013:133).

Menurut Rosa dan Shalahuddin (2013:140), UML terdiri dari tiga belas macam diagram yang dikelompokkan dalam tiga kategori, yaitu:

1. *Structure Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interactions Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

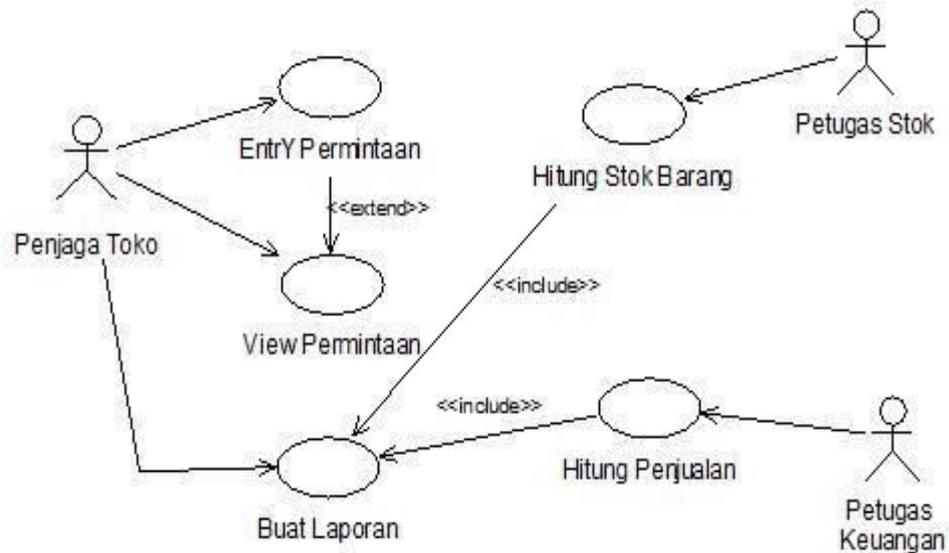
Berikut penjelasan empat diagram dari tiga belas macam diagram UML yang akan diuraikan:

1. *Use Case Diagram*

*Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih *actor* dengan sistem informasi yang akan dibuat serta menggambarkan fungsionalitas yang diterapkan dari sebuah sistem. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada *use case* adalah nama didefinisikan sesimple mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut *actor* dan *use case*.

- a. *Actor* merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari *actor* adalah gambar orang, tapi *actor* belum tentu merupakan orang.

- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau *actor*.



**Gambar II.3**

**Contoh Gambar Use Case**

Sumber: Sukamto dan Shalahuddin (2013:140)

2. *Activity Diagram*

Diagram aktivitas atau *activity diagram* menggambarkan *work flow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa apa yang dilakukan *actor*, jadi aktivitas yang dapat dilakukan oleh sistem. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :

- Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- Urutan atau pengelompokan tampilan dari sistem dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.

- c. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

### 3. *Component Diagram*

Diagram komponen atau *component diagram* dibuat untuk menunjukkan organisasi dan ketergantungan di antara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada di dalam sistem. Diagram komponen juga dapat digunakan untuk memodelkan hal-hal berikut :

- a. *Source code* program perangkat lunak.
- b. Komponen *executable* yang dilepas ke *user*.
- c. Basis data secara fisik.

### 4. *Deployment Diagram*

Diagram *deployment* atau *deployment diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat digunakan untuk memodelkan hal-hal berikut :

- a. Sistem tambahan (*embedded sistem*) yang menggambarkan rancangan *device, node, hardware*.
- b. Sistem *client* atau *server*.
- c. Sistem terdistribusi murni.
- d. Rekayasa ulang aplikasi.

## **H. *Entity Relationship Diagram(ERD)***

Bin Ladjamudin (2013:142), ERD adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam system secara abstrak. Menurut

Rosa dan Shalahuddin(2013:50) *ERD* adalah suatu pemodelan awal basis data. Komponen-komponen yang terdapat didalam *Entity Realtionship* Model:

1. *Entity* adalah suatu kumpulan objek atau sesuatu yang dapat dibedakan atau dapat didefinisikan secara unik.kumpulan entitas yang sejenis disebut *entity set*, di bawah ini adalah jenis-jenis entity dan contohnya:
  - a. *Entity* yang besifat fisik, yaitu *entity* yang dapat dilihat, contohnya rumah, kendaraan, mahapeserta didik, dosen dan lainnya.
  - b. *Entity* yang bersifat konsep atau logika, yaitu *entity* yang tidak dapat dilihat, contohnya pekerjaan, perusahaan, rencana, mata kuliah dan lainnya.Simbol yang digunakan untuk *entity* adalah persegi panjang.
2. *Relationship* adalah hubungan yang terjadi antara satu entititas atau lebih *entity*, *relationship* tidak mempunyai keberadaan fisik, kecuali yang mewarisi hubungan antara *entity* tersebut dan *relationship set* adalah kumpulan *relationship* yang sejenis, contoh simbol yang digunakan adalah bentuk belah ketupat, *diamond* atau *rectangle*.
3. *Attribute* adalah karakteristik dari *entity* atau *relationship* yang menyediakan penjelasan detail tentang atau *relationship* tersebut dan *attributevalue* adalah suatu data *actual* atau informasi yang disimpan di suatu *attribute* di dalam suatu *entity* atau *relationship*. Terdapat dua jenis atribut yaitu:
  - a. *Identifer (key)* untuk menentukan suatu *entity* secara unik
  - b. *Descriptor (ninkey attribute)* untuk menentukan karakteristik dari suatu *entity* yang tidak unik.
4. *Indicator Type* terdapat dua jenis yaitu:

- a. *Indicator Type Associative Object* berfungsi sebagai suatu objek dan suatu *relationship*.
  - b. *Indicator Type Subpertype* terdiri dari suatu obyek dan satu sub kategori atau lebih yang dihubungkan dengan satu *relationship* yang tidak bernama.
5. *Cardinality Ratio* atau *Mapping Cardinality* adalah menjelaskan hubungan batasan jumlah keterlambatan satu *entity* dengan *entity* lainnya atau banyaknya *entity* yang bersesuaian dengan *entity* yang lain melalui *relationship*. Jenis *Cardinality Ratio*:
1. *One to One* (1:1) adalah hubungan satu *entity* dengan satu *entity*.
  2. *One to Many* (1:M) adalah hubungan suatu *entity* dengan banyak *entity* atau banyak *entity* dengan satu *entity*.
  3. *Many to Many* (M:M) adalah hubungan banyak *entity* dengan banyak *entity*.
6. Derajat *Relationship* menyatakan jumlah *entity* yang berpartisipasi di dalam suatu *relationship*, terdapat tiga jenis yaitu:
1. *Unary degree* (derajat satu) adalah derajat yang memiliki satu *relationship* untuk dua buah *entity*.
  2. *Binary degree* (derajat dua) adalah derajat yang memiliki satu *relationship* untuk dua buah *entity*.
  3. *Tenary degree* (derajat tiga) adalah derajat yang memiliki satu *relationship* untuk tiga atau lebih *entity*.

7. *Participation Constraint* menjelaskan apakah keberadaan suatu *entity* tergantung pada hubungannya dengan *entity* lain. Terdapat dua macam *participation constraints* yaitu:

a. Total *participation*, yaitu keberadaan suatu *entity* tergantung pada hubungannya dengan *entity* lain dalam E\_R digambarkan dua garis penghubung antar *entity* dan *relationship*.

b. *Partial participation*, yaitu keberadaan suatu *entity* tidak tergantung pada hubungan dengan *entity* lain dalam E\_R digambarkan dengan satu garis penghubung.

### I. *Logical Record Structure*

Menurut Iskandar dan Rangkuti (2008:126) “*Logical Record Structure* terdiri dari *link-link* diantara *tipe record*. *Link* ini menunjukkan arah dari satu *tipe record* lainnya. Banyak *link* dari LRS yang diberi tanda *field-field* yang kelihatan pada kedua *link tipe record*. Penggambaran LRS mulai dengan menggunakan model yang dimengerti”.

*Logical Record Structure* dibentuk dengan nomor dari *tipe record* dan beberapa *tipe record* digambarkan oleh kotak empat persegi panjang dan dengan nama yang unik. Perbedaan LRS dengan diagram *entity relationship diagram* nama *tipe record* berada diluar kotak *field tipe record* ditempatkan.

Dua metode yang dapat digunakan, dimulai dengan hubungan kedua model yang dapat dikonversikan ke LRS. Metode yang lain dimulai dengan *Entity Relationship diagram* dan langsung dikonversikan ke LRS.

1. Konversi ERD ke LRS, Diagram *entity relationship diagram* harus diubah ke bentuk LRS (struktur record secara logik). Dari bentuk LRS inilah yang nantinya dapat ditransformasikan ke bentuk relasi (tabel).

Konversi ERD ke LRS Sebuah model sistem yang digambarkan dengan sebuah ERD akan mengikuti pola permodelan tertentu. Dalam kaitannya dengan konversi ke LRS, untuk perubahan yang terjadi adalah mengikuti aturan-aturan berikut:

- a. Setiap entitas diubah kebentuk kotak dengan nama entitas, berada diluar kotak dan atribut berada didalam kotak.
  - b. Sebuah relationship kadang disatukan, dalam sebuah kotak bersama entitas, kadang sebuah kotak bersama-sama dengan entitas, kadang disatukan dalam sebuah kotak tersendiri.
3. Konversi LRS ke relasi (tabel) relasi atau tabel adalah bentuk pernyataan data secara grafis 2 (dua) dimensi, yang terdiri dari kolom dan baris. Relasi adalah bentuk visual dari sebuah file, dan tiap tuple dalam sebuah field, atau yang dalam bentuk lingkaran Diagram entity relationship dikenal dengan sebutan atribut. Konversi dari logical record structure. dilakukan dengan cara:
- a. Nama logical record structure menjadi nama relasi.
  - b. Tiap atribut menjadi sebuah kolom didalam relasi.

#### **J. *Black Box Testing***

*Black Box Testing* adalah tipe testing yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya. Sehingga para tester memandang perangkat lunak seperti layaknya sebuah “kotak hitam” yang penting dilihat isinya, tapi cukup dikenai proses testing di bagian luar. Jenis testing ini hanya memandang perangkat lunak dari sisi spesifikasi dan kebutuhan yang telah didefinisikan pada saat awal perancangan, menurut Rizky (2011:264).

Jenis *black box testing*, perangkat lunak tersebut akan dieksekusi

kemudian berusaha dites apakah telah memenuhi kebutuhan pengguna yang didefinisikan pada saat awal tanpa harus membongkar listing programnya.

## 2.2. Penelitian Terkait

Dalam tinjauan jurnal ini penulis menggunakan dua referensi jurnal yang berhubungan dengan sistem informasi penjualan barang untuk mendukung isi dari penulisan skripsi ini.

Menurut Hakim, dkk (2014:79), “Adapun kesimpulan dari penulisan skripsi ini adalah Desain Sistem Informasi penjualan berbasis web ini dibuat agar dapat memudahkan konsumen. Dimana pihak konsumen dapat kemudahan melihat detail informasi semua produk dengan mengakses *website* tersebut dan kemudahan bagi pihak penjual yaitu dapat meningkatkan efisiensi dan efektifitas proses manajemen penjualan. Penerapan dari proses penjualan barang mulai dari pemesanan sampai konfirmasi pembayaran dapat dilakukan secara *online* dimana pihak konsumen tidak perlu datang ke gerai untuk mendapatkan produk yang diinginkan cukup dengan mengakses *website* tersebut melalui media internet”.

Menurut Kosasi (2014:232), “Sistem informasi penjualan berbasis web memiliki fitur navigasi yang memberikan kemudahan bagi pengunjung, dalam hal ini baik konsumen ataupun pelanggan saat mengunjungi halaman situs web. Memiliki kontribusi menyelesaikan persoalan untuk memberikan jaminan layanan informasi yang lebih dekat kepada konsumen dan beralih ke proses digitisasi penjualan. Menampilkan pesan-pesan tertentu dalam mengarahkan pengunjung, calon pembeli dapat melakukan proses pemesanan barang secara online kapanpun dan dimanapun serta mendapatkan informasi produk secara up to date. Kemampuan aplikasi digitisasi penjualan menawarkan banyak peluang baru terutama kesempatan memperluas pangsa pasar dengan biaya operasional yang murah karena semua transaksi dapat berlangsung dengan tidak bergantung kepada waktu dan tempat transaksi bisnis. Untuk realisasi penerapannya membutuhkan kesiapan sumberdaya manusia dan ketersediaan infrastruktur teknologi informasi dalam kelancaran transaksi dan menjalin interaksi yang personal dengan konsumen (pelanggan) perusahaan “.