

BAB II

LANDASAN TEORI

2.1. Tinjauan Pusaka

A. Pengertian Sistem Informasi

Menurut Sucipto (2010:1-2). Menjelaskan bahwa “Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu.”

Informasi berarti data yang telah dibentuk dalam suatu format yang mempunyai arti dan berguna bagi manusia. Sebaliknya data merupakan sekumpulan baris fakta yang mewakili peristiwa yang terjadi pada organisasi atau lingkungan fisik sebelum diolah dalam suatu format yang dapat dipahami dan digunakan orang. Sistem informasi juga bermanfaat untuk lingkungan eksternal. (Sucipto, 2010: 222-223).

Terdapat berbagai macam pengertian sistem informasi menurut beberapa ahli, diantaranya sebagai berikut :

Menurut Mulyanto (2009:29) memberi pengertian bahwa “Sistem informasi merupakan suatu komponen yang terdiri dari manusia, teknologi informasi, dan prosedur kerja yang memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk mencapai suatu tujuan.”

Menurut Sutabri (2012:46), menyatakan bahwa:

“Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan.”

Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan (*Building Block*), yang terdiri dari blok masukan, blok model, blok keluaran, blok teknologi, blok basis data, dan blok kendali. Sebagai suatu sistem, keenam blok bangunan tersebut masing-masing saling berinteraksi satu dengan yang lain membentuk suatu kesatuan untuk mencapai sasaran . Blok bangunan itu menurut Sutabri (2012:47) terdiri dari:

a. Blok Masukan (*Input Block*)

Input mewakili data yang masuk ke dalam sistem informasi. Input disini termasuk metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

b. Blok Model (*Model Block*)

Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara tertentu untuk menghasilkan keluaran yang diinginkan.

c. Blok Keluaran (*Output Block*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

d. Blok Teknologi (*Technology Block*)

Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan. Blok teknologi terdiri dari teknisi (*Brainware*), perangkat lunak (*Software*) dan perangkat keras (*Hardware*).

e. Blok Basis Data (*Database Block*)

Basis data (*Database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Basis data diakses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut dengan DBMS (*Database Management Systems*).

f. Blok Kendali (*Controls Block*)

Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahankesalahan dapat langsung cepat diatasi.

Sistem informasi dapat dibentuk sesuai kebutuhan organisasi masing-masing. Oleh karena itu, untuk dapat menerapkan sistem yang efektif dan efisien diperlukan perencanaan, pelaksanaan, pengaturan, dan evaluasi sesuai keinginan masing-masing organisasi. Sehingga sistem informasi dapat diklasifikasikan sebagai berikut :

a. Sistem informasi berdasarkan level organisasi

Dikelompokkan menjadi level operasional hingga level manajerial.

b. Sistem informasi berdasarkan aktifitas manajemen

Dikelompokkan menjadi sistem informasi perbankan, sistem informasi akademik, sistem informasi kesehatan, sistem informasi asuransi dll.

c. Sistem informasi berdasarkan fungsionalitas bisnis

Dikelompokkan menjadi sistem informasi akuntansi, sistem informasi keuangan, sistem informasi manufaktur, sistem informasi pemasaran dan sistem informasi sumber daya manusia.

Tujuan dari sistem informasi adalah menghasilkan informasi dari bentuk data yang diolah menjadi bentuk yang berguna bagi para pemakainya. (Jogiyanto, 2010:13).

Juga dijelaskan bahwa tujuan sistem informasi terdiri dari Kegunaan (*Usefulness*), Ekonomi (*Economic*), Keandalan (*Realibility*), Pelayanan Langgan (*Customer Service*), Kesederhanaan (*Simplicity*), dan Fleksibilitas (*Fleksibility*).

a. Kegunaan (*Usefulness*)

Sistem harus menghasilkan informasi yang akurat, tepat waktu, dan relevan untuk pengambilan keputusan manajemen dan personil operasi di dalam organisasi.

b. Ekonomi (*Economic*)

Semua bagian komponen sistem termasuk laporan, pengendalian, mesin harus menyumbang suatu nilai manfaat setidaknya sebesar biaya yang dibutuhkan.

c. Keandalan (*Realibility*)

Keluaran sistem harus mempunyai tingkatan ketelitian yang tinggi dan sistem itu sendiri harus mampu beroperasi secara efektif bahkan pada waktu komponen manusia tidak hadir atau saat komponen mesin tidak beroperasi secara temporer.

d. Pelayanan Langgan (*Customer Service*)

Sistem harus memberikan pelayanan dengan baik atau ramah kepada para pelanggan. Sehingga sistem tersebut dapat diminati oleh para pelanggannya.

e. Kesederhanaan (*Simplicity*)

Sistem harus cukup sederhana sehingga terstruktur dan operasinya dapat dengan mudah dimengerti dan prosedurnya mudah diikuti.

f. *Fleksibilitas (Flexibility)*

Sistem harus cukup fleksibel untuk menangani perubahan-perubahan yang terjadi, kepentingannya cukup beralasan dalam kondisi dimana sistem beroperasi atau dalam kebutuhan yang diwajibkan oleh organisasi.

B. Model Pengembangan Sistem

Langkah penyelesaian masalah skripsi ini sesuai dengan tahapan pengembangan perangkat lunak menggunakan model proses atau paradigma *waterfall*.

Waterfall model merupakan salah satu model proses perangkat lunak yang mengambil kegiatan proses dasar seperti spesifikasi, pengembangan, validasi, dan evolusi, dan merepresentasikannya sebagai fase-fase proses yang berbeda seperti analisis dan definisi persyaratan, perancangan perangkat lunak, implementasi, pengujian unit, integrasi sistem, pengujian sistem, operasi dan pemeliharaan (Sommerville, 2011:28).

Menurut Sommerville (2011:31) Tahap-tahap utama dari model *waterfall* adalah sebagai berikut:

1. Analisis dan Definisi Persyaratan

Proses mengumpulkan informasi kebutuhan sistem/perangkat lunak melalui konsultasi dengan user system. Proses ini mendefinisikan secara rinci mengenai fungsi-fungsi, batasan dan tujuan dari perangkat lunak sebagai spesifikasi sistem yang akan dibuat.

2. Perancangan Sistem dan Perangkat Lunak

Proses perancangan sistem ini difokuskan pada empat atribut, yaitu struktur data, arsitektur perangkat lunak, representasi antarmuka, dan detail (algoritma) prosedural. Yang dimaksud struktur data adalah representasi dari hubungan logis antara elemen-elemen data individual.

3. Implementasi dan Pengujian Unit

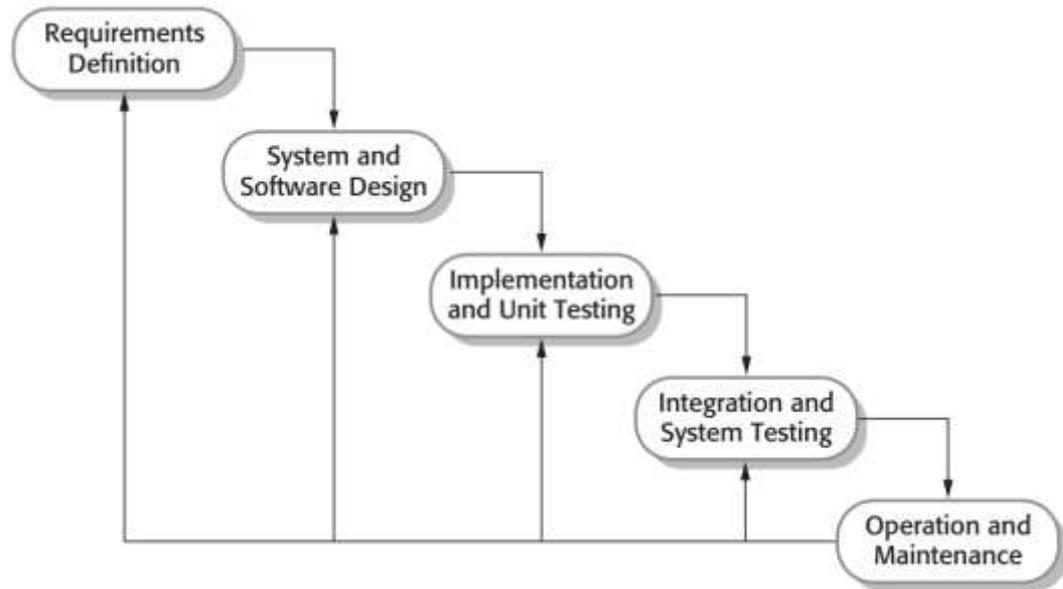
Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Kemudian pengujian unit melibatkan verifikasi bahwa setiap unit program telah memenuhi spesifikasinya.

4. Integrasi dan Pengujian Sistem

Unit program/program individual diintegrasikan menjadi sebuah kesatuan sistem dan kemudian dilakukan pengujian. Dengan kata lain, pengujian ini ditujukan untuk menguji keterhubungan dari tiap-tiap fungsi perangkat lunak untuk menjamin bahwa persyaratan sistem telah terpenuhi. Setelah pengujian sistem selesai dilakukan, perangkat lunak dikirim ke pelanggan/user.

5. Operasi dan Pemeliharaan

Tahap ini biasanya memerlukan waktu yang paling lama. Sistem diterapkan (*di-install*) dan dipakai. Pemeliharaan mencakup koreksi dari beberapa kesalahan yang tidak diketemukan pada tahapan-tahapan sebelumnya, perbaikan atas implementasi unit sistem dan pengembangan pelayanan sistem, sementara persyaratan-persyaratan baru menjadi sebuah penambahan.



Sumber: Sommerville (2011:30)

Gambar II.1 Model *Waterfall*

C. *E-Commerce*

Electronic Commerce (perniagaan electronic) merupakan bagian dari *Electronic Business* (bisnis yang dilakukan dengan menggunakan *electronic transmission*). Definisi global tentang *e-Commerce* adalah semua bentuk transaksi perdagangan barang atau jasa yang dilakukan secara elektronik.

Menurut Pujastuti (2014:1) *E-Commerce* adalah suatu jenis dari mekanisme bisnis secara elektronik yang memfokuskan diri pada transaksi bisnis berbasis individu dengan menggunakan internet (teknologi berbasis jaringan digital) sebagai medium pertukaran barang atau jasa baik antara dua buah institusi (*business to business*) dan konsumen langsung (*business to consumer*), melewati kendala ruang dan waktu yang selama ini merupakan hal-hal yang dominan. Dengan aplikasi *e-Commerce*, hubungan antar perusahaan dengan entitas eksternal lainnya (pemasok, distributor, rekanan, konsumen) dapat dilakukan secara lebih cepat, lebih intensif,

lebih murah daripada aplikasi prinsip manajemen secara konvensional (*door to door, one-to-one relationship*).

Salah satu media yang digunakan dalam *e-Commerce* adalah internet. Perkembangan media ini merupakan yang paling pesat dibanding dengan media lain dalam mendukung *e-Commerce*. Ada dua faktor pendukung yang menyebabkan internet berkembang lebih cepat dalam memediasi *e-Commerce*, yakni:

1. Internet memiliki jangkauan yang sangat luas, murah, cepat, dan mudah diakses oleh publik.
2. Internet menggunakan electronic data sebagai media penyampaian pesan atau data sehingga dapat dilakukan pengiriman dan penerimaan informasi secara mudah dan ringkas, baik dalam bentuk data elektronik analog maupun digital.

D. Pemograman Terstruktur

Menurut Triyanto (2013:1) menjelaskan bahwa “Pemrograman yaitu aktivitas membuat program, yaitu menyusun sejumlah perintah yang dikenal computer. Terstruktur dapat berarti terpola, bentuk yang mengikuti aturan tertentu, juga berarti sesuatu yang sistematis”

Selanjutnya Triyanto (2008:1) juga menjelaskan tentang sejarah, tujuan, kriteria dan struktur pemograman terstruktur sebagai berikut:

1. Sejarah Pemograman Terstruktur

Orang pertama yang mencetuskan ide pemrograman terstruktur adalah Profesor Edsger W. Dijkstra dari University of Eindhoven, Nederland. Ide

utamanya adalah bahwa statemen GOTO sebaiknya tidak digunakan di dalam pemrograman terstruktur, sebab bisa membuat program menjadi rumit.

Ide ini ditanggapi oleh HD Milis, yang beranggapan bahwa pemrograman terstruktur semestinya tidak hanya dihubungkan dengan tanpa penggunaan GOTO, tetapi yang lebih utama adalah struktur program itulah yang menentukan apakah suatu pemrograman terstruktur atau tidak.

Ide pemrograman terstruktur muncul karena jumlah baris program semakin lama semakin besar, tentu saja hal ini terjadi karena diinginkan aplikasi yang lengkap dan lebih berkualitas.

Dengan ide pemrograman terstruktur diharapkan dapat membantu manajemen source code (kode program) sehingga program mudah untuk dikelola bagi kepentingan selanjutnya.

2. Tujuan Pemograman Terstruktur

Tujuan utama pemrograman terstruktur adalah agar program-program besar menjadi lebih mudah ditelusuri alur logikanya, mudah untuk dimodifikasi (dikembangkan) dan mudah pula untuk ditemukan bagian yang salah ketika program sedang diuji.

3. Kriteria Pemograman Terstruktur

- a. Struktur programnya; jelas dan tegas
- b. Fasilitas penulisan kode program; jelas dan tegas
- c. Statemen untuk kebutuhan Selection dan Looping; lengkap
- d. Fasilitas menyatakan berbagai type data (struktur data); lengkap dan tegas
- e. Fasilitas pemberian komentar; lengkap

- f. Fasilitas instruksi yang tersedia (operasi arithmatik/matematik, string, ...); lengkap
 - g. Fasilitas modular (baik internal maupun eksternal); lengkap
 - h. Fasilitas debugging, mudah dan jelas
4. Struktur Pemograman Terstruktur

Dalam pemrograman terstruktur hanya dikenal 3 struktur :

a. Sekuensial

Yaitu program yang tidak memiliki lompatan. Baris program dijalankan secara normal (lurus) satu per-satu dari atas ke bawah.

b. *Selection*

Yaitu program yang memiliki pilihan apakah harus menjalankan baris program sesuai dengan urutannya atau melompati sejumlah baris program tersebut.

c. *Looping*

Yaitu program yang juga mengandung pilihan apakah akan mengulangi program yang sudah pernah dijalankan sebelumnya atau tidak.

E. *Unified Modeling Language (UML)*

Menurut Pressman (2009:841) UML adalah bahasa standar yang digunakan untuk menulis sebuah *blueprint* atau rancangan dari perangkat lunak. UML bisa digunakan untuk menggambarkan, menjelaskan, membangun dan mendokumentasikan *artifacts* (bagian-bagian) dari perangkat lunak.

Menurut Pressman (2009:842-846) UML terdiri dari berbagai tipe diagram, antara lain :

1. *Activity diagram*

Activity Diagram melengkapi *use case* dengan cara menyediakan gambaran grafis mengenai alur interaksi pada skenario tertentu. Mirip dengan *flowchart*, *activity diagram* menggunakan kotak tumpul untuk mengartikan sebuah fungsi (aktifitas) spesifik dari sistem, tanda panah sebagai alur kerja sistem, piramida sebagai percabangan, dan garis tebal untuk menunjukkan bahwa aktivitas paralel sedang berjalan.

2. *Use case diagram*

Use case adalah sebuah kata-kata naratif atau *template* yang menjelaskan fungsi atau fitur sebuah sistem dari sudut pandang pengguna. Sebuah *use case* ditulis oleh pengguna dan berfungsi sebagai dasar untuk membuat model yang lebih komperhensif atau menyeluruh.

3. *Deployment Diagram*

Sebuah *Deployment Diagram* dari UML berfokus pada struktur perangkat lunak dan berguna untuk menunjukkan distribusi fisik sistem perangkat lunak antara *hardware* hingga lingkungan pengguna.

4. *Component Diagram*

Component diagram menggambarkan struktur dan hubungan antar komponen perangkat lunak, termasuk ketergantungan (*dependency*) di antaranya.

Komponen perangkat lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang

lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

F. *Entity-relation Diagram (ERD)*

Menurut Connolly & Begg (2015:405) memberikan penjelasan bahwa “ERD adalah pendekatan *top – down* untuk mendesain basis data yang dimulai dengan melakukan identifikasi data penting yang biasa disebut entitas dan relasi antara data yang direpresentasikan dalam model”.

Sedangkan menurut Hoffer, dkk (2011:59) mengatakan bahwa:

ERD adalah representasi grafik dari data untuk organisasi atau untuk area bisnis, menggunakan entitas sebagai kategori data dan *relationships* untuk asosiasi antar entitas. Dari pengertian di atas dapat diambil kesimpulan bahwa ERD adalah model yang merepresentasikan data dalam entitas dan hubungan antar entitas secara jelas yang akan digunakan untuk membangun basis data.

1. Tipe Entitas (*Entity Type*)

Menurut pendapat Connolly & Begg (2015:406) menjelaskan bahwa “tipe entitas adalah sekumpulan objek dengan properti yang sama yang diidentifikasi dari organisasi atau perusahaan yang memiliki keberadaan yang bebas (*independent existence*).” Connolly & Begg (2015:406) juga menambahkan bahwa “Konsep dasar dari ER model adalah *entity type* atau tipe entitas, yang mewakili sebuah grup dari ‘objek’ yang ada pada ‘dunia nyata’ dengan beberapa properti yang sama.” Selain itu Connolly & Begg (2015:408) memberi penjelasan juga bahwa “*entity occurrence* adalah sebuah objek dari suatu tipe entitas yang dapat diidentifikasi secara unik.”

Cara memrepresentasi diagram dari tipe entitas yaitu setiap entitas dilambangkan dengan sebuah persegi panjang diberi nama dari entitas tersebut. Nama tipe entitas biasanya adalah kata benda tunggal. Huruf pertama dari setiap kata pada nama tipe entitas ditulis dengan huruf besar. (Connolly & Begg, 2015:408)

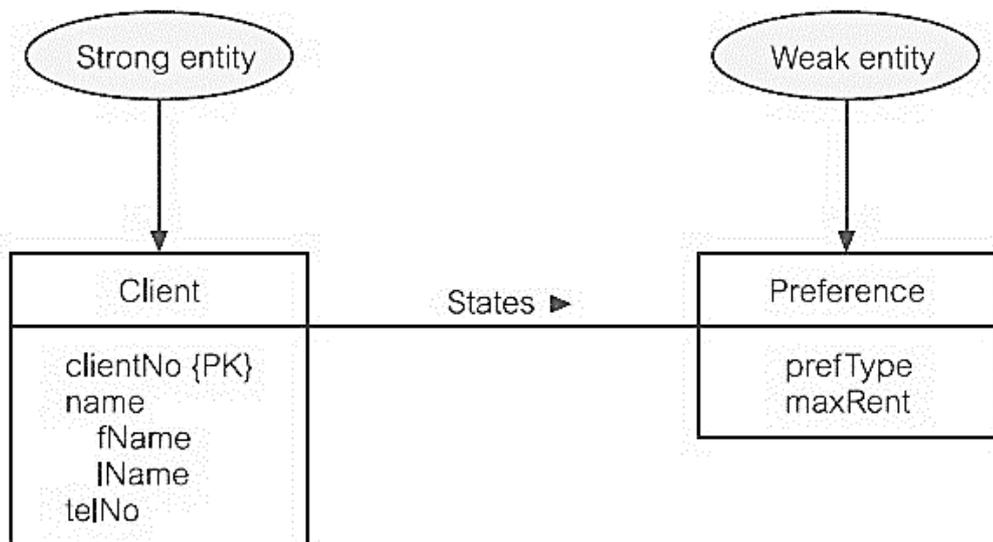
Menurut Connolly & Begg (2015:417), tipe entitas dibagi dua jenis yaitu sebagai berikut:

a. Entitas Kuat (*Strong Entity*)

Suatu tipe entitas yang dapat berdiri sendiri (*independent*) dan tidak bergantung pada keberadaan entitas lainnya.

b. Entitas Lemah (*Weak Entity*)

Suatu entitas tidak dapat berdiri sendiri (*dependent*) dan bergantung pada keberadaan entitas lainnya.



Sumber: Connolly & Begg (2015:418)

Gambar II.2 Contoh *strong entity* dan *weak entity*

2. Tipe Relasi (*Relationship Type*)

Suatu entitas akan semakin bermakna disaat kita menghubungkan satu entitas dengan entitas lainnya. Proses menghubungkan ini didasarkan pada kebutuhan data pada masing-masing entitas. Menurut Connolly & Begg (2015:408) “*relationship types* adalah asosisasi yang memiliki arti diantara tipe-tipe entitas. Sebuah *relationship type* adalah satu set asosisasi antara satu atau lebih dari tipe-tipe entitas yang terlibat”.

Connoly & Begg (2015:410) menjelaskan cara memrepresentasi diagram dari tipe relasi yaitu dengan sebuah garis yang menghubungkan entitas-entitas yang saling berhubungan. Garis tersebut diberi nama sesuai dengan nama hubungannya dan diberi tanda panah satu arah disamping nama hubungannya. Biasanya sebuah relasi dinamakan dengan menggunakan kata kerja seperti mengatur atau dengan sebuah frame singkat yang meliputi sebuah kata kerja, seperti DisewaOleh, sedangkan tanda panah ditempatkan di bawah nama relasi yang mengindikasikan arah bagi pembaca untuk mengartikan nama dari suatu relasi. Huruf pertama Pada suatu relasi ditulis dengan huruf besar.

3. Atribut (*Attribute*)

Jika menurut Connoly & Begg (2015:413) “*attribute* adalah *property* dari suatu entitas ataupun tipe relasi. Setiap entitas pasti memiliki penjelasan tentang entitas itu sendiri, penjelasan-penjelasan itu sering disebut sebagai *attribute* yang menyertai suatu entitas.”

Connoly & Begg (2015:413-414) mengklasifikasikan atribut menjadi *simple* atau *composite*, *single-valued* atau *multi-valued* atau *derived* dengan penjelasan sebagai berikut:

a. *Simple attribute.*

Atribut yang terdiri dari komponen tunggal dengan keberadaan yang berdiri sendiri atau independen

b. *Composite attribute*

Atribut yang terdiri dari beberapa komponen, masing-masing dengan keberadaan independen

c. *Single-valued attribute*

Attribute yang hanya memiliki sebuah nilai untuk setiap *accurance* dari sebuah entitas.

d. *Multi-valued attribute*

Attribute yang memiliki nilai untuk setiap *accurance* dari etitas.

e. *Derived Attribute*

Attribute yang nilainya diperoleh dari pengolahan atau diturunkan dari attribute lain yang berhubungan.

4. *Attribute Domain*

Attribute domain adalah suatu set nilai yang diijinkan untuk satu atau lebih atribut. Setiap atribut memiliki nilai yang disebut domain. Connoly & Begg (2015:413-414) mengklasifikasikan atribut menjadi *simple* atau *composite*, *single-valued* atau *multi-valued* atau *derived* dengan penjelasan sebagai berikut:

- a. *Simple attribute* : atribut yang terdiri dari komponen tunggal dengan keberadaan yang independen. *Simple attribute* tidak bisa dibagi ke dalam komponen yang lebih kecil.
- b. *Composite attribute* : atribut yang terdiri dari beberapa komponen yang masing – masing dengan keberadaan independen. Beberapa atribut dapat dibagi ke dalam komponen yang lebih kecil.
- c. *Single-Valued Attribute* : atribut yang memiliki nilai tunggal untuk setiap kemunculan tipe entitas.
- d. *Multi-valued attribute* : atribut yang memiliki beberapa nilai untuk setiap kemunculan tipe entitas.

- e. *Derived Attribute* : atribut yang merepresentasikan nilai turunan dari nilai atribut yang terkait

5. *Relational Keys*

Menurut Connolly & Begg (2015:158-159), *relational keys* dibagi menjadi beberapa jenis, yaitu:

a. *Superkey*

Merupakan atribut atau set atribut yang secara unik mengidentifikasi sebuah tupel dalam relasi

b. *Candidate key*

adalah *superkey* dalam relasi. *Candidate Key* (K) bagi sebuah relasi memiliki dua *property*, yaitu Keunikan yang mana dalam setiap *tuple* dari R, nilai K secara unik mengidentifikasikan *tuple* tersebut dan *Irreducibility* yang berarti tidak ada *subset* yang sesuai dari K yang memiliki keunikan sifat, ketika sebuah *key* terdiri dari lebih dari satu *attribute* yang disebut sebagai *composite key*.

c. *Primary Key*

Adalah *candidate key* yang terpilih untuk mengidentifikasi *tuple* secara unik dalam sebuah relasi. Sementara *candidate key* yang tidak terpilih disebut *alternate key*.

d. *Foreign key*

Adalah sebuah atau sekelompok *attribute* dalam relasi yang dibandingkan dengan *candidate key* pada beberapa relasi.

6. *Structural Constraints*

Berdasarkan Connoly & Begg (2015:419), batasan yang akan dibahas ini merupakan menempatkan tipe-tipe entitas yang berpartisipasi dalam sebuah relasi. Tipe utama dari batasan pada sebuah relasi disebut dengan *multiplicity*. *Multiplicity* merupakan jumlah kejadian yang mungkin dari suatu entitas yang berhubungan dengan kejadian tunggal dari entitas lain yang terkait melalui relasi tertentu. *Multiplicity* dibagi menjadi tiga jenis, yaitu :

- a. *One-to-one (1:1) relationship*
- b. *One-to-many (1:*) relationship*
- c. *Many-to-many (*:*) relationship*

G. Logical Record Structure (LRS)

Menurut Kroenke (2006:76), *Logical Record Structure (LRS)* adalah representasi dari struktur record – record pada tabel-table yang terbentuk dari hasil relasi antar himpunan entitas. Beberapa tipe *record* digambarkan oleh kotak persegi panjang dan dengan nama yang unik. LRS terdiri dari hubungan diantara tipe *record*. Link ini menunjukkan arah dari data tipe *record* lainnya. Penggambaran LRS mulai dengan menggunakan model yang dimengerti. Dua metode yang digunakan, dimulai dengan hubungan kedua model yang dapat dikonversikan ke LRS. Metode lain yang dimulai dengan ERD dan langsung dikonversikan ke LRS.

H. Pengujian Black Box

Proses pengujian *black box* adalah pengujian yang dilakukan dengan cara menguji aplikasi dengan memasukan data kedalam form-form yang telah disediakan. Tahap ini adalah kelanjutan dari tahap implementasi. Pengujian yang dilakukan berfokus pada persyaratan fungsional perangkat lunak. Pengujian ini

memungkinkan perekayasa sistem mendapatkan serangkaian kondisi input yang sepenuhnya semua persyaratan fungsional untuk semua program.

2.2. Penelitian Terkait

Dalam penelitian terkait ini penulis menggunakan dua referensi jurnal yang berhubungan dengan sistem manajemen aset untuk mendukung penulisan skripsi ini.

Menurut Oprastini (2010:1) *E-Commerce* mempunyai kelebihan dalam bisnis yaitu untuk memperpendek jarak tempuh, perluasan pasar, perluasan jaringan mitra bisnis, mengurangi biaya-biaya perusahaan, meningkatkan kinerja perusahaan cara efektif dan efisien dan meningkatkan *customer royalty*. Dengan melihat kelebihan *e-commerce*, maka dapat memberikan solusi bagi perusahaan khususnya penjualan dengan cara menampilkan produk pada web Pengrajin Besi Desa Pasir Wetan. *Software* yang digunakan untuk mendukung hal tersebut adalah *dreamweaver*, PHP, dan MySQL. Sistem informasi ini dibangun dengan bahasa pemrograman PHP dan untuk *database* menggunakan MySQL, sedangkan untuk desainnya menggunakan *dreamweaver*. Metode pengumpulan data menggunakan observasi, wawancara, dan studi pustaka. Pengembangan sistem menggunakan *waterfall*.

Menurut Juansah & Sudrajat (2012:1) Perkembangan teknologi dan informasi yang semakin pesat membuat perusahaan dituntut untuk dapat mengikuti perkembangan yang ada. Seiring dengan perkembangan teknologi dan informasi, persaingan dalam dunia bisnis menuntut perusahaan untuk dapat terus meningkatkan pelayanan proses informasi dan penjualan produk-produk yang dihasilkan dari perusahaan. Penerapan *e-commerce* tentang Penjualan Jok Kursi Rotan di PD. LANGGAN WAJAR CUSHION guna meningkatkan omset pendapatan penjualan Jok Kursi Rotan merupakan bentuk aplikasi penjualan *Online*. Aplikasi penjualan ini diharapkan mampu memberikan solusi penjualan yang masih menggunakan cara manual.

Metode pengembangan *e-commerce* ini menggunakan *Linear Sequential Model*. Metode ini disebut juga metode *waterfall*. Metode ini mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematis dan sekuensial.

E-commerce tentang penjualan Jok Kursi Rotan diharapkan dapat memberikan kemudahan bagi para pelanggan yang ingin melakukan transaksi pembelian pada PD. Langgan Wajar Cushion sehingga dapat mempermudah dalam proses transaksi pembelian.