

BAB II

LANDASAN TEORI

2.1. Penelitian Terkait

Menurut Oktavia (2014:2) mengungkapkan bahwa diperlukan sebuah sistem yang dapat digunakan sebagai penunjang kegiatan manajemen aset, sehingga seluruh pengelolaan aset seperti pengalokasian aset, pergantian aset serta perbaikan aset dapat dimonitor dengan mudah dan tertata dengan baik. Dengan demikian, kegiatan operasional dapat berlangsung secara optimal.

Menurut Utomo (2010:2) mengungkapkan bahwa dengan adanya Sistem Informasi Manajemen Aset yang terkomputerisasi, dapat memberikan kemudahan dalam beberapa hal, seperti : pencatatan data, pendataulangan serta penghitungan jumlah dan nilai barang untuk keperluan manajemen aset selanjutnya serta pembuatan laporan.

2.2. Konsep Dasar Sistem Informasi

2.2.1. Sistem

1. Pengertian Sistem

Pada dasarnya sistem adalah suatu kerangka dari prosedur-prosedur yang saling berhubungan, yang disusun sesuai dengan skema yang menyeluruh untuk melaksanakan kegiatan atau fungsi utama dari perusahaan yang dihasilkan oleh suatu proses tertentu yang bertujuan untuk menyediakan informasi untuk

membantu mengambil keputusan manajemen operasi perusahaan dari hari ke hari serta menyediakan informasi yang layak untuk pihak di luar perusahaan.

Pengertian sistem yang dikemukakan oleh para ahli adalah sebagai berikut :

Menurut Sutarman (2009:5) dalam bukunya yang berjudul Pengantar Teknologi Informasi, “Sistem adalah kumpulan elemen yang saling berinteraksi dalam suatu kesatuan untuk menjalankan proses pencapaian suatu tujuan utama”.

Menurut Jogiyanto (2009:34) dalam bukunya yang berjudul Analisis dan Desain Sistem Informasi, “Sistem dapat di definisikan dengan pendekatan prosedur dan dengan pendekatan komponen”.

Berdasarkan beberapa pendapat yang dikemukakan diatas dapat ditarik kesimpulan bahwa sistem adalah kumpulan bagian-bagian atau sub sistem-sub sistem yang disatukan dan dirancang untuk mencapai suatu tujuan.

2. Karakteristik Sistem

Sesuatu dapat dikatakan sistem jika didalamnya terdapat ciri-xiri atau karakteristik sebuah sistem. Menurut Mulyanto (2009:2) dalam bukunya Sistem Informasi Konsep dan Aplikasi :

a. Mempunyai Komponen Sistem (*Component System*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan.

b. Batasan Sistem (*Boundary*)

Batasan sistem (*boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem lainnya atau dengan lingkungan luarnya.

c. Lingkungan Luar Sistem (*Environments*)

Apa pun diluar batas dari sistem yang dapat mempengaruhi operasi sistem, baik pengaruh yang menguntungkan ataupun yang merugikan.

d. Penghubung Antar Komponen (*Interface*)

Media penghubung antara sub sistem dengan sub sistem yang lainnya.

e. Masukan (*Input*)

Energi yang dimasukkan ke dalam sistem.

f. Pengolahan (*Processing*)

Bagian yang melakukan perubahan dari masukan untuk menjadi keluaran yang diinginkan.

g. Sasaran dan Tujuan (*Objective*)

Target atau sasaran akhir yang ingin dicapai oleh suatu sistem.

h. Keluaran (*Output*)

Hasil dari pemrosesan. Keluaran dapat berupa informasi sebagai masukan pada sistem lain atau hanya sebagai sisa pembuangan.

i. 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.

2.2.2. Informasi

Informasi merupakan hal yang sangat penting bagi perusahaan dalam pengambilan keputusan sehari-hari. Beberapa ahli mendefinisikan informasi sebagai berikut :

Menurut Mulyanto (2009:12) dalam bukunya yang berjudul Sistem Informasi Konsep dan Aplikasi, “Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya, sedangkan data merupakan sumber informasi yang menggambarkan suatu kejadian yang nyata”.

Menurut Jogiyanto (2009:8) dalam bukunya yang berjudul Analisis dan Desain Sistem Informasi, “Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya”.

2.2.3. Sistem Informasi

1. Pengertian Sistem Informasi

Menyangkut pemahaman tentang pengertian sistem informasi, Mulyanto (2009:29) dalam bukunya yang berjudul Sistem Informasi Konsep dan Aplikasi mengutipkan beberapa pendapat para ahli, diantaranya :

Menurut Alter, sistem informasi adalah “Kombinasi antar prosedur kerja, informasi, orang dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi.

Menurut Bodnar dan Hopwood, sistem informasi adalah “Kumpulan perangkat keras dan perangkat lunak yang dirancang untuk mentransformasikan data ke dalam bentuk informasi yang berguna”.

Menurut Gelinas, Oram dan Wiggins, sistem informasi adalah “Suatu sistem buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun, menyimpan dan mengelola data serta menyediakan informasi keluaran kepada pemakai”.

Menurut Turban, McLean dan Waterbe, sistem informasi adalah “Sistem yang mengumpulkan, memproses, menyimpan, menganalisis dan menyebarkan informasi untuk tujuan spesifik”.

Menurut Wilkinson, sistem informasi adalah “Kerangka kerja yang mengkoordinasikan sumber daya (manusia, komputer) untuk mengubah masukan (*input*) menjadi keluaran (*output*) guna mencapai sasaran-sasaran perusahaan”.

Dari beberapa definisi diatas, dapat disimpulkan bahwa sistem informasi adalah suatu sistem yang terdiri dari kumpulan komponen sistem, yaitu perangkat lunak (*software*), perangkat keras (*hardware*) dan manusia (*brainware*) yang memproses informasi menjadi sebuah keluaran (*output*) yang berguna untuk mencapai suatu tujuan tertentu dalam suatu organisasi.

2. Komponen Sistem Informasi

Menurut Sutabri (2012:39) dalam bukunya yang berjudul “Analisis Sistem Informasi” Sistem informasi juga memiliki enam buah komponen, yaitu komponen masukan (*input*) , komponen model, komponen keluaran (*output*), komponen teknologi, komponen basis data dan komponen kontrol atau pengendalian.

Berikut ini adalah penjelasan dari komponen sistem informasi, yaitu :

a. Komponen Masukan (*Input*)

Komponen masukan merupakan data yang masuk ke dalam sistem informasi. Komponen ini perlu ada karena merupakan bahan dasar dan pengolahan informasi.

b. Komponen Model

Informasi yang dihasilkan oleh sistem informasi berasal dari data yang diambil dari basis data yang diolah dalam suatu model-model tertentu.

c. Komponen Keluaran (*Output*)

Produk dari sistem informasi adalah keluaran berupa informasi yang berguna bagi para pemakainya. Keluaran merupakan komponen yang harus ada di sistem informasi.

d. Komponen Teknologi

Teknologi merupakan komponen yang penting di sistem informasi. Tanpa adanya teknologi yang mendukung, maka sistem informasi tidak dapat menghasilkan informasi tepat pada waktunya.

e. Komponen Basis Data (*Database*)

Basis data (*database*) adalah kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Dari definisi ini, terdapat tiga hal yang berhubungan dengan basis data, yaitu data itu sendiri, simpanan permanen (*storage*) dan perangkat lunak yang disediakan untuk memanipulasi basis data .

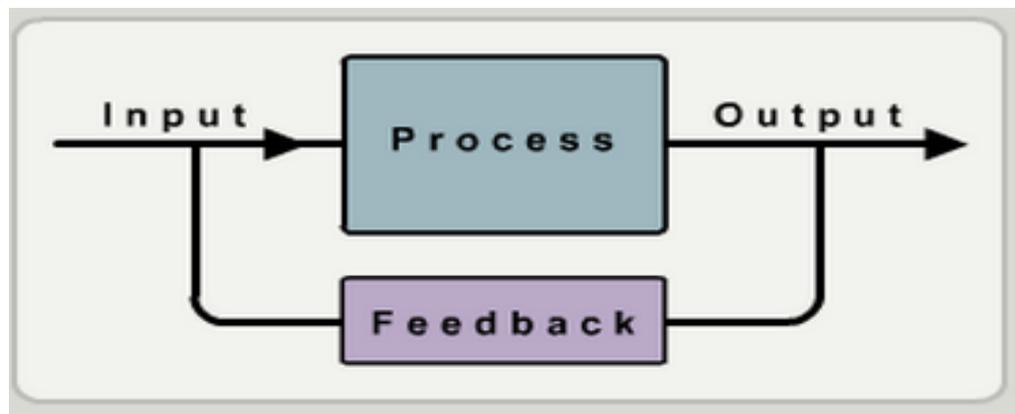
f. Komponen Kontrol atau Pengendalian

Komponen kontrol juga merupakan komponen yang penting dan harus ada di sistem informasi. Komponen kontrol ini digunakan untuk menjamin bahwa informasi yang dihasilkan oleh sistem informasi merupakan informasi yang akurat.

3. Aktifitas Dasar Sistem Informasi

Dalam konsep dasar sistem informasi, aktifitas dasar sistem informasi menurut Laudon dan Laudon (2010,p46-47) adalah sebagai berikut :

- a. Masukan (*Input*), melibatkan pengumpulan data mentah dari dalam organisasi atau lingkungan eksternal untuk pengolahan dalam suatu informasi.
- b. Proses (*Process*), melibatkan proses mengkonversi *input* mentah ke bentuk yang lebih bermakna.
- c. Keluaran (*Output*), mentransfer proses informasi kepada orang yang akan menggunakannya atau kepada aktivitas yang akan digunakan.
- d. Umpan balik (*Feedback*), *output* yang dikembalikan ke anggota organisasi yang sesuai untuk kemudian membantu mengevaluasi atau mengkoreksi tahap *input*.



Sumber : Laudon dan Laudon (2010, p46-47)

Gambar II. 1
Kegiatan Sistem Informasi

2.3. Konsep Dasar Pemrograman

2.3.1. Pengertian Pemrograman

Menurut Binanto (2009:1) , “Program adalah himpunan atau kumpulan instruksi yang dibuat oleh *programmer* atau suatu bagian *executable* dari suatu perangkat lunak (*software*)”.

Aktivitas membuat program disebut sebagai pemrograman. Jadi pemrograman merupakan suatu kumpulan urutan perintah ke komputer untuk mengerjakan sesuatu.

2.3.2. Bahasa Pemrograman

Bahasa pemrograman merupakan suatu himpunan dari aturan sintaks dan emantic yang dipakai untuk mendefinisikan program komputer.

Secara umum ada beberapa fungsi utama dari sebuah bahasa pemrograman, diantaranya adalah sebagai berikut :

1. Sebagai media komunikasi antara operator dengan mesin.
2. Sebagai media mengoperasikan sebuah mesin.
3. Sebagai media bagi programmer dalam mengembangkan sebuah aplikasi.
4. Memberikan perintah kepada mesin dan komputer.

2.4. *Software* dan Bahasa Pemrograman

Untuk mengembangkan proses perancangan aplikasi, penulis menggunakan *software* dan bahasa pemrograman sebagai berikut :

2.4.1. Bahasa Pemrograman Visual Basic

Microsoft Visual Basic merupakan sebuah bahasa pemrograman yang menawarkan *Intregated Development Environment* (IDE) visual untuk membuat perangkat lunak berbasis sistem operasi Microsoft Windows dengan menggunakan model pemrograman (COM).

Visual Basic merupakan turunan bahasa pemrograman BASIC dan menawarkan pengembangan perangkat lunak komputer berbasis grafik dengan cepat.

2.4.2. Visual Basic .Net Versi 2008 (VB 9.0)

Microsoft Visual Basic .Net adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak diatas sistem .NET Framework, dengan menggunakan bahasa BASIC.

Visual Basic .Net 2008 merupakan versi Visual Basic .Net yang dirilis oleh Microsoft pada tanggal 19 November 2007.

2.4.3. Crystal Report

Menurut Kuniyo dan Kusri (2007:264) dalam bukunya yang berjudul *Membangun Sistem Informasi Akuntansi dengan Visual Basic & SQL Server*, menjelaskan bahwa “Crystal report merupakan program yang dapat digunakan untuk membuat, menganalisis dan menterjemahkan informasi yang terkandung dalam *database* atau program ke dalam berbagai jenis laporan yang sangat flexible”.

Menurut Supardi (2012:9) dalam bukunya yang berjudul *Aplikasi Database Penjualan dengan VB6 & MySQL* menjelaskan bahwa “Crystal report merupakan sebuah alat bantu (*tool*) untuk membuat laporan”.

Berdasarkan definisi di atas, maka dapat disimpulkan bahwa crystal report merupakan program yang digunakan untuk membuat laporan yang terdapat di dalam *database*.

2.4.4. MySQL

Menurut Baskoro (2012:362) mengemukakan bahwa “MySQL adalah suatu perangkat lunak *database* relasi (*Relation Database Management System / RDMS*) yang bersifat *multithread* dan *multi-user*”. MySQL adalah *open source software* yang dibuat oleh sebuah perusahaan Swedia yaitu MySQL AB. Kelebihan MySQL diantaranya adalah sebagai berikut :

1. MySQL merupakan sebuah *database* yang mampu menyimpan data berkapasitas sangat besar hingga berukuran *gigabyte* sekalipun.
2. MySQL didukung oleh *server* ODBC, yang artinya *database* MySQL dapat diakses menggunakan aplikasi apa saja termasuk berupa visual seperti Delphi dan Visual Basic.
3. MySQL adalah *database* yang menggunakan enkripsi *password*.
4. MySQL merupakan *server* *database* *multi user*, artinya *database* ini dapat digunakan oleh banyak orang.
5. MySQL dapat menciptakan lebih dari 16 kunci per tabel dan satu kunci memungkinkan belasan *fields*.

2.4.5.SQLite

Menurut Kreibich (2010, p1), SQLite adalah paket perangkat lunak *public-domain* yang menyediakan *relational database management system* atau RDBMS. Sistem penghubung *database* digunakan untuk menyimpan catatan *user-defined* di tabel yang besar. Selain untuk penyimpanan dan manajemen data, mesin *database* dapat memproses perintah query yang kompleks yang mengkombinasikan data dari beberapa tabel untuk menghasilkan laporan dan ringkasan data.

Beberapa fitur yang terdapat pada SQLite:

1. Tidak Memerlukan Server (*Serverless*)

SQLite tidak membutuhkan proses server atau sistem terpisah untuk mengoperasikan. SQLite *library* mengakses penyimpanan file secara langsung.

2. Tidak Ada Pengaturan (*Zero Configuration*)

Tidak ada server artinya tidak ada pengaturan. Membuat sebuah *database* SQLite semudah membuka sebuah file.

3. Dapat di Akses Berbagai Platform (*Cross-Platform*)

Seluruh *database* berada dalam dalam file *cross platform* tunggal, tidak membutuhkan administrasi.

4. Menampung Seluruh *Database* Dalam Satu *Library* (*Self-Contained*)

Sebuah *library* tunggal berisi seluruh sistem *database*, yang terintegrasi langsung ke aplikasi host.

5. Sedikit Memori (*Small Runtime Footprint*)

Membangun *database* lebih kecil dari kode *megabyte* dan membutuhkan beberapa megabyte memori. Dengan beberapa penyesuaian, baik ukuran *library* dan penggunaan memori bisa berkurang secara signifikan.

6. Transaksi (*Transactional*)

Transaksi memungkinkan akses yang aman dari beberapa proses atau rangkaian proses.

7. Fitur-Fitur yang Mendukung (*Full-Featured*)

SQLite mendukung sebagian besar dari fitur *query language* yang ditemukan di standard SQL92 (SQL2).

8. Terpercaya (*Highly Reliable*)

Tim pengembang SQLite mengambil kode pengujian dan verifikasi dengan sangat serius serta telah melewati proses *testing*.

2.5. Teori Pendukung

2.5.1. *Entity Relationship Diagram (ERD)*

Menurut Yakub (2008:25) mengemukakan bahwa “*Entity Relationship Diagram* merupakan suatu model jaringan yang menggunakan susunan data yang yang disimpan pada sistem secara abstrak”.

Dalam rekayasa perangkat lunak, sebuah *Entity Relationship Model (ERM)* merupakan abstrak dan konseptual representasi data. *Entity Relationship* adalah salah satu metode pemodelan basis data yang digunakan untuk menghasilkan skema konseptual untuk jenis atau model data emantic sistem. Dimana sistem seringkali memiliki basis data relasional dan ketentuannya bersifat *top-down*. Diagram untuk menggambarkan model *Entity Relationship* ini disebut *Entity Relationship Diagram, ER Diagram* atau *ERD*.

Notasi-notasi simbolik yang digunakan dalam *Entity Relationship Diagram* adalah sebagai berikut :

1. *Entitas*

Entitas adalah segala sesuatu yang dapat digambarkan oleh data. *Entitas* juga dapat diartikan sebagai individu yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain (Fathansyah, 1999). Ada dua macam *entitas*, yaitu *entitas* kuat dan *entitas* lemah. *Entitas* kuat merupakan *entitas* yang tidak memiliki ketergantungan dengan *entitas* lainnya. Contohnya *entitas* anggota. Sedangkan *entitas* lemah merupakan *entitas* yang kemunculannya tergantung pada keberadaan *entitas* lain dalam suatu relasi.

2. *Atribut*

Atribut merupakan pendeskripsian karakteristik dari *entitas*. *Atribut* digambarkan dalam bentuk lingkaran atau elips. *Atribut* yang menjadi kunci *entitas* atau *key* diberi garis bawah.

3. *Relasi* atau Hubungan

Relasi menunjukkan adanya hubungan diantara sejumlah *entitas* yang berasal dari himpunan *entitas* yang berbeda.

4. Penghubung atau Garis

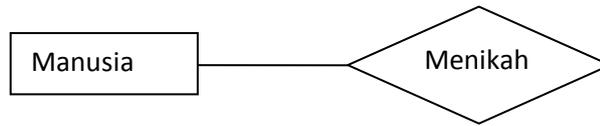
Penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atribut dinyatakan dalam bentuk garis.

Derajat *relationship* menjelaskan jumlah *entity* yang berpartisipasi dalam suatu *relationship*.

1. Derajat Satu (*Unary*)

Adalah satu buah *relationship* menghubungkan satu buah *entity*.

Contoh :



Sumber : Ladjamudin (2005:144)

Gambar II.2
Contoh Unary

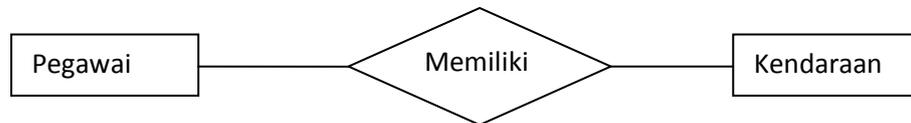
Keterangan :

Manusia menikah dengan manusia, *relationship* menikah hanya menghubungkan *entity* manusia.

2. Derajat Dua (*Binary*)

Adalah satu buah *relationship* yang menghubungkan dua buah *entity*.

Contoh :



Sumber : Ladjamudin (2005:145)

Gambar II.3
Contoh Binary

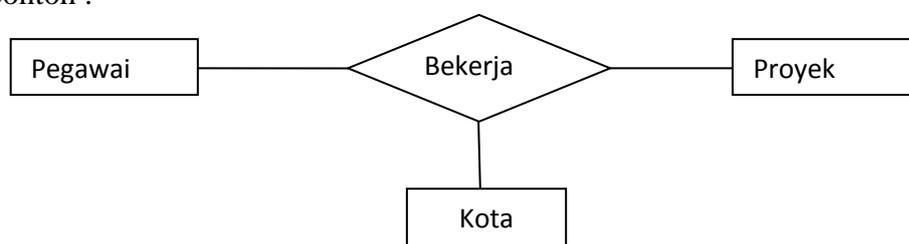
Keterangan :

Pegawai memiliki kendaraan, sebuah *relationship* memiliki hubungan *entity* pegawai dan *entity* kendaraan.

3. Derajat Tiga (*Ternary*)

Adalah satu buah *relationship* yang menghubungkan tiga buah *entity*.

Contoh :



Sumber : Ladjamudin (2005:145)

Gambar II.4
Contoh Ternary

Keterangan :

Pegawai pada kota tertentu mempunyai suatu proyek. *Relationship* bekerja menghubungkan *entity* pegawai , proyek dan kota.

Cardinality rasio yaitu menjelaskan batasan pada jumlah *entity* yang berhubungan melalui suatu *relationship*. Ada empat jenis *cardinality* rasio, yaitu :

1. *One To One* (1:1)

Yaitu perbandingan antara *entity* pertama dengan *entity* kedua berbanding satu berbanding satu.

Contoh :



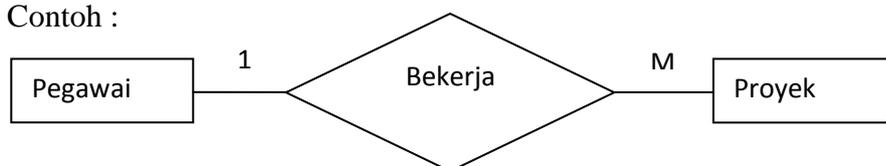
Sumber : Ladjamudin (2005:147)

Gambar II.5
Contoh One To One

2. *One To Many* (1:M)

Yaitu perbandingan antara *entity* pertama dengan *entity* kedua berbanding satu berbanding banyak.

Contoh :



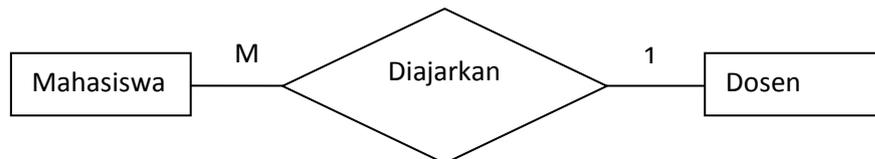
Sumber : Ladjamudin (2005 : 147)

Gambar II.6
Contoh One To Many

3. *Many To One (M:1)*

Yaitu perbandingan antara *entity* pertama dengan *entity* kedua berbanding banyak berbanding satu.

Contoh :



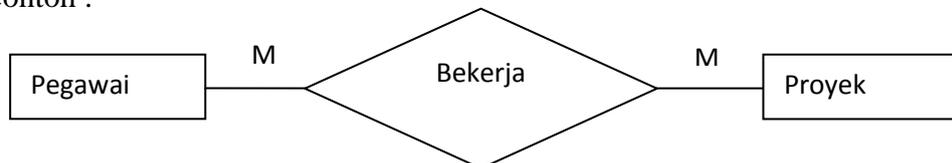
Sumber : Ladjamudin (2005 : 147)

Gambar II.7
Contoh *Many To One*

3. *Many To Many (M:M)*

Yaitu perbandingan antara *entity* pertama dengan *entity* kedua berbanding banyak berbanding banyak.

Contoh :



Sumber : Ladjamudin (2005 : 148)

Gambar II.8
Contoh *Many To Many*

Langkah-langkah untuk membuat *Entity Relationship Diagram (ERD)* adalah sebagai berikut :

1. Mengidentifikasi dan menetapkan seluruh himpunan entitas yang akan terlibat.
2. Menentukan atribut-atribut *key* dari masing-masing himpunan entitas.
3. Mengidentifikasi dan menetapkan seluruh himpunan relasi diantara himpunan entitas yang ada beserta *foreign key*-nya.

4. Menentukan derajat dan cardinality rasio relasi untuk setiap himpunan relasi.
5. Melengkapi himpunan relasi dengan atribut-atribut yang bukan kunci (*non-key*).

2.5.2. Logical Record Structure (LRS)

Logical Record Structure dibentuk dengan nomor dari tipe *record*. Beberapa tipe *record* digambarkan oleh kotak empat persegi panjang dan dengan nama yang unik. Beda *Logical Record Structure (LRS)* dengan diagram *Entity Relationship Diagram (ERD)* nama tipe *record* berada diluar kotak *field* tipe *record* ditempatkan. LRS 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.

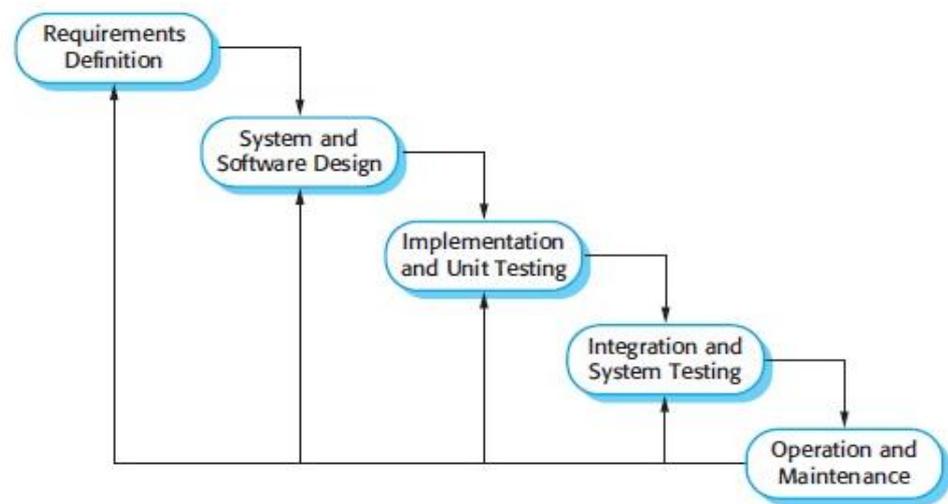
Menurut Ladjamudin (2006:210) “terdapat dua aturan dalam melakukan transformasi *Entity Relationship Diagram (ERD)* ke *Logical Record Structure (LRS)*”. Dua aturan tersebut yaitu :

1. Setiap *entity* akan diubah ke bentuk sebuah kotak dengan nama *entity* berada di luar kotak dan atribut berada di dalam kotak.
2. Sebuah relasi kadang disatukan dalam sebuah kotak bersama *entity*, kadang dipisah dalam sebuah kotak tersendiri.

Aturan pokok di atas akan sangat dipengaruhi oleh elemen yang menjadi titik perhatian utama pada langkah transformasi yaitu *cardinality*.

2.5.3. Model *Waterfall*

Menurut Sommerville (2011, p30-31), tahapan utama dari *waterfall model* langsung mencerminkan aktifitas pengembangan dasar. Terdapat 5 tahapan pada *waterfall model*, yaitu *requirement analysis and definition*, *system and software design*, *implementation and unit testing*, *integration and system testing*, dan *operation and maintenance*.



Sumber : Sommerville (2011, p30)

Gambar II.9
Waterfall Model

Berikut adalah penjelasan dari tahapan-tahapan tersebut :

1. *Requirement Analysis and Definition*

Merupakan tahapan penetapan fitur, kendala dan tujuan sistem melalui konsultasi dengan pengguna sistem. Semua hal tersebut akan ditetapkan secara rinci dan berfungsi sebagai spesifikasi sistem.

2. *System and Software Design*

Dalam tahapan ini akan dibentuk suatu arsitektur sistem berdasarkan persyaratan yang telah ditetapkan. Dan juga mengidentifikasi dan

menggambarkan abstraksi dasar sistem perangkat lunak dan hubungan-hubungannya.

3. *Implementation and Unit Testing*

Dalam tahapan ini, hasil dari desain perangkat lunak akan direalisasikan sebagai satu set program atau unit program. Setiap unit akan diuji apakah sudah memenuhi spesifikasinya.

4. *Integration and System Testing*

Dalam tahapan ini, setiap unit program akan diintegrasikan satu sama lain dan diuji sebagai satu sistem yang utuh untuk memastikan sistem sudah memenuhi persyaratan yang ada. Setelah itu sistem akan dikirim ke pengguna sistem.

5. *Operation and Maintenance*

Dalam tahapan ini, sistem diinstal dan mulai digunakan. Selain itu juga memperbaiki *error* yang tidak ditemukan pada tahap pembuatan. Dalam tahap ini juga dilakukan pengembangan sistem seperti penambahan fitur dan fungsi baru.

2.5.4. Kode Batang (*Barcode*)

Kode batang (*barcode*) adalah sekumpulan kode yang berbentuk garis, dimana masing-masing ketebalan setiap garis berbeda sesuai dengan isi kodenya. Kode batang (*barcode*) pertama kalinya diperkenalkan dan dipatenkan di Amerika oleh Norman Joseph Woodland dan Bernard Silver, mahasiswa Drexel Institute of Technology pada akhir tahun 40-an.

1. Jenis-Jenis *Barcode*

Menurut Wahyono (2010:6), dalam bukunya yang berjudul “Membuat Sendiri Aplikasi dengan Memanfaatkan Barcode” mengungkapkan bahwa *barcode* memiliki banyak sekali standard dan tidak terpaku pada satu model tertentu. Tetapi meski demikian, secara umum *barcode* terbagi menjadi dua kelompok besar, yaitu *barcode* satu dimensi dan *barcode* dua dimensi.

a. *Barcode* Satu Dimensi (*1D – Linear Barcode*)

Barcode ini dinamakan satu dimensi atau ada juga yang menyebut *linear barcodes* karena kodenya hanya terdiri dari baris-baris. Berikut ini adalah beberapa jenis *barcode* satu dimensi :

1) Tipe code 39

Merupakan *barcode alphanumeric (full ASCII)* yang dapat mewakili abjad (A-Z) dan angka (0-9), serta beberapa karakter lain, seperti \$, /, +, %, titik dan spasi. Tipe code 39 memiliki jumlah digit maksimal 16. Kode seperti ini biasanya digunakan untuk *barcode* buku maupun *barcode* anggota perpustakaan.



Sumber : Wahyono (2010:8)

Gambar II.10
Contoh Barcode Tipe 39

2) Tipe Code 128

Barcode tipe *code* 128 juga tergolong di dalam *barcode alphanumeric* (*full ASCII*), tetapi memiliki kerapatan yang lebih tinggi dan juga panjang baris yang bervariasi. *Barcode* tipe *code* 128 biasanya digunakan untuk aplikasi, seperti pengaturan maskapai pelayaran dan pengelolaan gudang. Setiap karakter pada *code* 128 dikodekan oleh 3 bar dan 3 spasi dengan ketebalan masing-masing elemen 1 sampai 4 kali ketebalan minimum (*module*). Jumlah total *module* untuk bar selalu genap, sedangkan untuk spasi selalu ganjil. Selain itu *code* 128 memiliki 3 *start character* yang berbeda sehingga *code* 128 memiliki 3 sub set karakter yang bersesuaian dengan *start character*-nya.



Sumber : Wahyono (2010:8)

Gambar II.11
Contoh Barcode Tipe Code 128

3) Code 25

Merupakan kode *barcode* yang hanya untuk angka (0-9), maksimum 32 digit. Jadi *barcode* ini berbentuk numerik dan memiliki panjang baris yang bervariasi. *Barcode* yang juga disebut sebagai *interleaved 2 of 5* bisa digunakan untuk aplikasi dalam dunia industri dan laboratorium.



Sumber : Wahyono (2010:9)

Gambar II.12
Contoh Barcode Code 25

4) EAN 13

Simbologi *barcode* model ini dikeluarkan EAN untuk identitas suatu produk. Standarisasi EAN menggunakan 3 digit pertama adalah kode untuk negara asal produk, 4 digit berikutnya adalah *manufacture number*, 5 digit berikutnya adalah *product number* dan 1 digit terakhir adalah *check digit* atau angka untuk melakukan test validasi *barcode*. Kode EAN juga sering digunakan di Indonesia untuk identifikasi produk nasional. Indonesia sendiri mempunyai kode 899 untuk 3 digit pertama.



Sumber : Wahyono (2010:10)

Gambar II.13
Contoh Barcode Jenis EAN 13

5) UPC (*Universal Product Code*)

Barcode UPC ini hanya terdiri dari angka (0-9) namun *barcode* harus mempunyai panjang tepat 11 atau 12 digit. Kurang atau lebih dari angka itu, tidak bisa digunakan. Jadi *barcode* ini berbentuk numerik

dan memiliki panjang baris yang tetap. UPC biasanya digunakan untuk pelabelan pada produk-produk kecil atau eceran. Simbol UPC dibuat untuk kemudahan pemeriksaan keaslian suatu produk dan bilangan UPC harus diregistrasikan atau terdaftar di *Uniform Code Council*.



Sumber : Wahyono (2010:10)

Gambar II.14
Contoh Barcode Jenis UPC

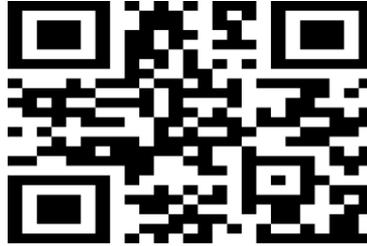
b. Barcode Dua Dimensi (2D – Matrix Barcode)

Barcode jenis ini disebut sebagai *barcode* dua dimensi karena tidak hanya terbuat dari garis-garis saja, tetapi lebih mendekati bentuk gambar tertentu. Dengan menggunakan dua dimensi, informasi atau data yang besar dapat disimpan di dalam suatu ruang yang lebih kecil. *Barcode* model ini dikembangkan lebih dari sepuluh tahun lalu. Berikut adalah beberapa jenis *barcode* dua dimensi :

1) QR Code

QR merupakan kependekan dari *Quick Response*, sebuah harapan dari pembuatnya bahwa *code* ini akan cepat di-*decode*. Tidak seperti *barcode* yang satu sisinya saja yang mengandung data, *QR code* mempunyai dua sisi yang berisi data. Hal tersebut membuat *QR code*

dapat lebih banyak memuat informasi dibandingkan *barcode* satu dimensi.



Sumber : Wahyono (2010:11)

Gambar II.15
Contoh Barcode 2 Dimensi QR Code

2) Kode PDF417

Model ini merupakan pengembangan dari *barcode* satu dimensi *linear*, tepatnya disebut sebagai *stacked linear barcode*. PDF sendiri diartikan sebagai *Portable Data File*. Model ini memiliki kemampuan untuk menyimpan lebih dari 2000 karakter di dalam ruang berukuran 4 inch persegi.

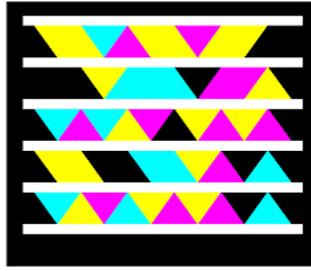


Sumber : Wahyono (2010:14)

Gambar II.16
Contoh Barcode 2 Dimensi PDF417

3) HCCB (*High Capacity Color Barcode*)

Salah satu contoh barcode dua dimensi adalah High Capacity Color Barcode (HCCB) yang di-develop oleh Microsoft dengan lisensi dari ISAN International Agency. Microsoft mengimplementasikan HCCB menggunakan 4 warna pada ukuran 5 x 10 grid.



Sumber : Wahyono (2010:12)

Gambar II.17
Contoh Barcode 2 Dimensi HBBC

4) Data Matrik

Kode data matrik merupakan *barcode* matrik dua dimensi yang terdiri dari sel hitam dan putih atau modul titik-titik di sekitar *rectangular pattern*. Menggunakan ukuran data sampai pada 2 *kilobytes*. Panjang *encoded* data tergantung pada *symbol dimensional* yang digunakan. Sebuah data matrik simbol dapat menyimpan sampai dengan 2.335 karakter *alphanumeric*. Salah satu contoh penggunaan data matrix ini yang sering dijumpai seperti barcode pada mini PCI card dengan ukuran symbol yang bervariasi mulai dari 8 x 8 sampai 144 x 144.



Sumber : Wahyono (2010:13)

Gambar II.18
Contoh Barcode 2 Dimensi Data Matrix

2. Keuntungan Penggunaan *Barcode*

Menurut Yuana (2010;67) dalam bukunya *Trik dan Ide Brilian Master PHP* menerangkan bahwa keuntungan penggunaan sistem *barcode* diantaranya adalah sebagai berikut :

- a. Proses *input* data lebih cepat, karena *barcode scanner* dapat membaca atau merekam data lebih cepat dibandingkan dengan melakukan *input* data secara manual.
- b. Proses *input* data lebih tepat, karena teknologi *barcode* mempunyai ketepatan yang tinggi dalam pencarian data.
- c. Penelusuran informasi data lebih akurat, karena teknologi *barcode* mempunyai akurasi dan ketelitian yang sangat tinggi.
- d. Peningkatan kerja manajemen, karena dengan penelusuran data yang lebih cepat, tepat dan akurat maka pengambilan keputusan oleh manajemen akan jauh lebih baik dan lebih tepat, yang nantinya akan sangat berpengaruh dalam menentukan kebijakan perusahaan.

2.6. Konsep Dasar Manajemen Aset

2.6.1. Pengertian Manajemen

Menurut Marwansyah (2009:1), “Manajemen adalah proses pencapaian tujuan organisasi melalui pelaksanaan fungsi-fungsi perencanaan, pengorganisasian, pengarahan dan pengawasan”.

2.6.2. Pengertian Aset

Menurut Sugiama (2013:15), mengungkapkan bahwa definisi aset adalah barang (*thing*) atau sesuatu barang (*anything*) yang dimiliki oleh seseorang atau sebuah organisasi baik swasta maupun pemerintah yang memiliki nilai ekonomi, nilai komersial dan nilai tukar.

Pada umumnya jenis aset dapat dibagi menjadi tiga, yaitu sebagai berikut :

1. Aset Lancar (*Current Assets*)

Adalah bentuk aset yang dalam waktu singkat dapat diubah menjadi uang kas. Aset lancar meliputi kas (*cash*), investasi jangka pendek (*temporary investment*), wesel tagih (*notes receivable*), piutang dagang (*accounts receivable*), penghasilan yang masih akan diterima (*accrued receivable*), persediaan (*inventories*) dan biaya yang dibayar di muka (*prepaid expenses*).

2. Aset Tetap (*Fixed Assets*)

Adalah harta kekayaan milik perusahaan yang dapat diukur dengan jelas dan bersifat permanen. Aset tetap dibeli dengan tujuan dipakai sendiri oleh perusahaan dan tidak dijual kembali. Contohnya adalah tanah (*land*), bangunan (*building*), mesin-mesin (*machinery*), perabot dan peralatan kantor (*office furniture and fixtures*), perabot dan peralatan toko (*store furniture and fixtures*) dan alat pengangkutan (*delivery equipment*).

3. Aset Tidak Berwujud (*Intangible Assets*)

Adalah semua aset yang tidak dapat disimpan dalam bentuk persediaan dan dipegang bentuknya tetapi dapat dirasakan. Aset tidak berwujud ini merupakan hak milik perusahaan dan kepemilikannya dilindungi oleh undang-undang.

Contohnya adalah hak cipta (*copyrights*), hak sewa atau hak kontrak (*leasehold*), hak monopoli (*franchises*) dan biaya organisasi (*organization costs*).

2.6.3. Manajemen Aset

1. Pengertian

Menurut Sugiana (2013:15), mengungkapkan bahwa manajemen aset adalah ilmu dan seni untuk memandu pengelolaan kekayaan yang mencakup proses merencanakan kebutuhan aset, mendapatkan, menginventarisasi, melakukan legal audit, menilai, mengoperasikan, membaharukan atau menghapuskan hingga mengalihkan aset secara efektif dan efisien.

2. Tujuan Manajemen Aset

Secara umum tujuan manajemen aset adalah untuk pengambilan keputusan yang tepat agar aset yang dikelola berfungsi secara efektif dan efisien.

Efektif dalam pengelolaan aset berarti aset yang dikelola dapat mencapai tujuan yang diharapkan organisasi yang bersangkutan, misal mencapai kinerja tertinggi dalam pelayanan pelanggan.

Adapun efisien berarti menggunakan sumber daya serendah mungkin untuk mendapat hasil (output) yang tinggi.

Tujuan manajemen aset yang lebih rinci diantaranya adalah sebagai berikut :

- a. Meminimalisasi biaya selama umur aset bersangkutan (*to minimize the whole life cost of assets*).
- b. Menghasilkan laba maksimum (*profit maximum*).

- c. Mencapai penggunaan serta pemanfaatan aset secara optimum
(optimizing the utilization of assets).