

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Dalam penulisan tugas akhir ini diperlukan teori-teori yang relevan atau sejalan untuk mendukung kemudahan dalam mempelajari serta merancang sistem yang diharapkan dapat berjalan maksimal, mangkus dan sangkil. Dengan demikian, akan sangat membantu untuk mempermudah proses berjalannya sistem.

2.1.1. Sistem

Yakub (2012:1) berpendapat bahwa “sistem adalah suatu jaringan kerja dari prosedur-prosedur yang berhubungan, terkumpul bersama-sama untuk melakukan suatu kegiatan atau tujuan tertentu”.

Menurut Sutabri (2012:10) berpendapat bahwa sistem adalah “suatu sistem yang dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain dan terpadu”.

Secara umum suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen yang terorganisir dan saling berinteraksi satu sama lain.

Menurut Sutabri (2012:20) mengungkapkan bahwa “suatu sistem mempunyai karakteristik atau sifat-sifat tertentu “, sebagai berikut :

a. Komponen Sistem

Suatu sistem yang terdiri dari sejumlah komponen atau elemen yang saling berinteraksi, yang artinya komponen atau elemen saling bekerjasama membentuk satu kesatuan. Komponen atau elemen sistem dapat berupa sub-sistem atau bagian-bagian sistem. Setiap sub-sistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi dan mempunyai proses secara keseluruhan.

b. Batasan Sistem

Adalah daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luar. Batasan sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batasan sistem tersebut memungkinkan suatu sistem di pandang sebagai satu kesatuan, dan batas suatu sistem juga menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

c. Lingkungan Luar Sistem

Adalah sistem yang merupakan semua diluar batas dari sistem yang mempengaruhi operasi sistem.

d. Penghubung Sistem

Merupakan suatu media penghubung antar sub-sistem dengan sub-sistem lainnya. Keluaran (*output*) dari sub-sistem akan menjadi masukan (*input*) untuk sub-sistem yang lainnya melalui penghubung.

e. Masukan Sistem (*Input*)

Masukan suatu sistem adalah energi yang di masukan kedalam sistem. Masukan tersebut dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang di

gunakan agar sistem tersebut dapat beroperasi. Sedangkan signal input adalah energi yang di proses untuk mendapatkan keluaran.

f. Keluaran Sistem

Keluaran sistem adalah hasil dari energi yang di olah dan di klasifikasikan menjadi keluaran yang berguna dari sisa pembuangan. Keluaran juga dapat dijadikan masukan untuk sub-sistem yang lainnya.

g. Pengolahan Sistem

Pengolahan sistem dapat mempunyai suatu bagian pengolahan yang akan merubah masukan menjadi keluaran. Suatu sub sistem proses pengolahan data barang akan menjadi laporan – laporan berupa tagihan dan laporan yang di butuhkan oleh manajemen.

h. Sasaran Sistem

Suatu sistem yang mempunyai maksud tertentu, ada yang bermaksud untuk mencapai tujuan. Goal biasanya di hubungkan dengan ruang lingkup yang lebih sempit. Sasaran suatu sistem sangat menentukan masukan yang di butuhkan sistem dan keluaran yang di hasilkan sistem. Suatu sistem dikatakan berhasil jika mengenai sasaran atau tujuan yang di harapkan.

Menurut Yakub (2012:4), menyatakan bahwa “Sistem dapat dikalsifikasikan dari beberapa sudut pandang”. Adapun penjelasan lebih detail mengenai klasifikasi sistem, yaitu:

a. Sistem Abstrak dan Sistem Fisik

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak nampak secara fisik. Sedangkan sistem fisik merupakan sistem yang ada secara fisik.

b. Sistem Alamiah dan Sistem Buatan Manusia

Sistem alamiah adalah sistem yang terbuat melalui proses alam, tidak di buat manusia. Sistem buatan manusia adalah sistem yang di rancang atau sengaja di buat oleh manusia.

c. Sistem Terbuka dan Sistem Tertutup

Sistem terbuka adalah sistem yang berhubungan dan di pengaruhi oleh lingkungan luarnya. Sedangkan tertutup merupakan sistem yang tidak berhubungan dan tidak di pengaruhi oleh lingkungan luarnya.

2.1.2. Basis Data

Menurut Fathansyah (2015:23) berpendapat bahwa basis data adalah “himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah”.

Sebuah sistem basis data dapat memiliki beberapa basis data. Setiap basis data dapat berisi sejumlah objek basis data (seperti tabel , indeks , dan lain-lain) di samping berisi data, setiap basis data juga menyimpan definisi struktur (baik untuk basis data maupun objek-objeknya secara rinci).

Basis Data terdiri atas 2 kata , yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang / berkumpul. Sedangkan data adalah *representasi* fakta dunia nyata mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang diwujudkan dalam bentuk angka, huruf, simbol, teks gambar, bunyi atau kombinasinya.

Sebagai satu kesatuan istilah, basis data (*database*) sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti :

1. Himpunan kelompok data (arsip) yang saling berhubungan yang di organisasi sedemikian rupa agar kelak dapat di manfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (*redudansi*) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan *file* atau tabel atau arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Basis data dan lemari arsip sesungguhnya memiliki prinsip kerja dan tujuan yang sama. Prinsip utamanya adalah pengaturan data atau arsip. Dan tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data atau arsip. Perbedaannya hanya terletak pada media penyimpanan yang di gunakan. Jika lemari arsip menggunakan lemari dari besi atau kayu sebagai media penyimpanan, maka basis data menggunakan media penyimpanan elektronik seperti cakram magnetis. Hal ini merupakan konsekuensi yang logis, karena lemari arsip langsung dikelola oleh manusia, sementara basis data dikelola melalui perantara mesin pintar elektronik (yang kita kenal sebagai komputer). Perbedaan media ini yang selanjutnya melahirkan perbedaan-perbedaan lain yang menyangkut jumlah dan jenis metode yang dapat digunakan dalam upaya penyimpanan.

Satu hal yang juga harus di perhatikan, bahwa basis data bukan hanya sekedar penyimpanan data secara elektronik (dengan bantuan komputer) artinya ,

tidak semua bentuk penyimpanan data secara elektronik bisa disebut basis data. Kita dapat menyimpan dokumen berisi data dalam *file teks* (dengan program pengolah kata), *file spread sheet*, dan lainnya, tetapi tidak bisa disebut sebagai basis data. Hal ini karena di dalamnya tidak ada pemilahan dan pengelompokan data sesuai jenis data. Kelak ketika *file – file* tersebut sudah cukup banyak, maka situasi ini tentu akan menyulitkan pencarian data tertentu. Yang sangat ditonjolkan dalam basis data adalah pengaturan, pemilahan, pengelompokan, pengorganisasian data yang akan kita simpan sesuai fungsi/ jenisnya. Pemilahan, pengelompokan, pengorganisasian ini dapat berbentuk sejumlah tabel terpisah atau dalam bentuk pendefinisian kolom – kolom (*field*) data dalam setiap tabel.

1. Pengertian MySQL (*My Structure Query Language*)

Menurut Aditya (2011:61), mengungkapkan bahwa “MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang *multithread*, *multiuser*, dengan sekitar 6 juta instalasi diseluruh dunia”. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. Tidak seperti Apache yang merupakan *software* yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia yaitu MySQL AB. MySQL AB memegang penuh hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan

MySQL AB adalah David Axmark, Allan Larsson, dan Michael “Monty” Widenius. Adapun kelebihan-kelebihan dari MySQL yaitu :

1. *Source* MySQL dapat diperoleh dengan mudah dan gratis.
2. Sintaksnya lebih mudah dipahami dan tidak rumit.
3. Pengaksesan basis data dapat dilakukan dengan mudah.
4. MySQL merupakan program yang *multithreaded*, sehingga dapat dipasang pada server yang memiliki multi CPU.
5. Didukung program-program umum seperti C, C++, Java, Perl, PHP, Python, dsb.
6. Bekerja pada berbagai *platform*. (tersedia berbagai versi untuk berbagai sistem operasi).
7. Memiliki jenis kolom yang cukup banyak sehingga memudahkan konfigurasi sistem *database*.
8. Mendukung ODBC untuk sistem operasi Windows.
9. Mendukung record yang memiliki kolom dengan panjang tetap atau panjang bervariasi.

2.1.3. Metode Pengembangan Perangkat Lunak

Metode yang digunakan pada pengembangan perangkat lunak ini menggunakan model waterfall. Menurut Sukanto dan M. Salahuddin (2013:28) model waterfall sering juga disebut model sekuensial linier (sequential linier) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, dan pengujian.

1. Analisi Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk pengumpulan 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

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

3. Pengujian

Pengujian focus 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.

4. Pendukung (support) atau pemeliharaan (maintenance)

Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang suda ada, tapi tidak untum membuat perangkat lunak baru.

2.2. Peralatan Pendukung (*tools system*)

Data adalah aliran fakta-fakta mentah yang menunjukkan peristiwa yang terjadi dalam organisasi dan lingkungan fisik sebelum diorganisir menjadi suatu bentuk yang bisa di pahami dan digunakan. *Tools system* atau peralatan pendukung yang penyusun gunakan dalam pembuatan analisis sistem ini meliputi Diagram Alir Data dan kamus data (*Data Dictionary*) dan peralatan lainnya yang membantu penyusun dalam pembuatan tugas akhir ini.

2.2.1 Diagram Alir Data (*Data Flow Diagram*)

Menurut Kenneth E Kendall dan Julie E Kendall (2010:263) mengungkapkan bahwa Diagram Alir Data (*Data Flow Diagram*) adalah “grafik yang menggambarkan pandangan mengenai masukan, proses dan keluaran system yang berhubungan dengan masukan, proses dan keluaran serta merepresentasikan dan menganalisis prosedur-prosedur mendetail dalam sistem yang lebih besar”.

Ada beberapa simbol yang digunakan dalam Diagram Alir Data. Simbol-simbol yang akan digunakan antara lain :

1. Kesatuan Luar (*Eksternal Entity*)

Kesatuan luar atau batasan sistem adalah merupakan kesatuan sistem yang berupa orang atau organisasi atau sistem lainnya yang berada dilingkungan luar sistem, yang akan memberikan input atau menerima output dari sistem.

2. Proses (*process*)

Adalah kegiatan yang di perlukan oleh manusia, mesin atau komputer dari hasil satu arus data yang termasuk ke dalam proses untuk menghasilkan arus data keluar dari proses.

- a. Identifikasi adalah berupa angka yang menunjukkan nomor acuan dari proses dan tulis pada bagian atas nama simbol proses.
- b. Nama proses menunjukkan apa yang dikerjakan oleh proses tersebut dan harus dilihat secara jelas dan lengkap untuk menggambarkan kegiatan proses
- c. Proses menunjukkan siapa atau dimana proses dilakukan.

3. Simpanan Data (*Data Store*)

Simpanan data merupakan simpanan data yang berupa :

- a. Suatu file atau database sistem komputer.
- b. Suatu arsip atau catatan normal.
- c. Suatu tabel acuan manual.
- d. Suatu kotak tempat data dimeja seseorang.
- e. Suatu agenda atau buku.

Nama dari *data store* menunjukkan nama *file* nya, di dalam menggambarkan simpanan data *DFD* diperhatikan beberapa hal sebagai berikut :

- a. Hanya proses saja yang berhubungan dengan simpanan data, karena yang menggunakan atau merubah data adalah suatu proses.
- b. Arus data yang menuju kesimpulan data suatu proses menunjukkan proses update terhadap data yang tersimpan pada simpanan data yang dapat berupa proses penambahan atau penyimpanan dokumen baru, merubah data dari suatu record atau disuatu dokumen yang ada disamping data.
- c. Arus yang berasal dari simpanan data ke suatu proses menunjukkan bahwa suatu proses tersebut menggunakan data yang ada di simpanan data.

d. Untuk suatu proses yang dilakukan kedua-duanya yaitu menggunakan data *update* simpanan data dipilih suatu penggambaran untuk memperjelas yaitu sebagai berikut :

- 1) Menggunakan sebuah garis dengan panah kedua arah yang berlawanan dari data simpanan.
- 2) Menggunakan arus data terpisah.

4. Arus Data (*Data Flow*)

Menunjukkan dari data yang didapat berupa masukan untuk sistem atau hasil dari proses sistem arus data dapat berbentuk :

- a. Berkas atau data yang digunakan.
- b. Tampilan atau input untuk komputer.
- c. Berkas atau data yang digunakan.
- d. Masukan untuk komputer.
- e. Komunikasi ucapan.
- f. Surat – surat atau memo.
- g. Surat isian yang di catat pada buku agenda.
- h. Transmisi dari suatu komputer lain.

Arus data sebaiknya diberi nama jelas dan mempunyai arti tulisan diatas garis panahnya.

A. Aturan Main

Persyaratan yang harus dipenuhi dalam menggambarkan diagram alir data ada empat, yaitu :

1. Di dalam Diagram Alir Data (DAD) tidak boleh menghubungkan external entity dengan external entity secara langsung.

2. Di dalam Diagram Alir Data (DAD) tidak boleh menghubungkan data store dengan data store secara langsung.
3. Di dalam Diagram Alir Data (DAD) tidak boleh menghubungkan data store dengan external entity secara langsung atau sebaliknya.
4. Setiap proses harus ada data flow masuk dan juga data flow keluar.

B. Langkah-Langkah Mengembangkan DAD

1. Membuat sebuah daftar tentang kegiatan-kegiatan bisnis dan digunakan untuk menentukan berbagai macam:
 - 1) Entitas Eksternal
 - 2) Aliran Data
 - 3) Proses-proses
 - 4) Penyimpanan data
2. Menciptakan sebuah diagram yang menunjukkan entitas-entitas eksternal dan aliran-aliran data menuju sistem.
3. Menggambar diagram nol yang menunjukkan proses-proses dan penyimpanan data.
4. Menciptakan diagram anak untuk setiap proses dalam diagram 0
5. Mengecek kesalahan dan memastikan label-label yang ditetapkan untuk setiap proses dan aliran data.

C. Tahapan Proses Pembuatan DAD

1. Buat Diagram Konteks

Tingkatan tertinggi dalam diagram alir data dan hanya memuat satu proses, menunjukkan system secara keseluruhan diberi nomor nol dan tidak memuat penyimpanan data.

2. Buat Diagram Nol

Menggambarkan detail dari diagram konteks, masukan dan keluaran yang ditetapkan dalam diagram konteks tetap konstan dalam semua diagram sub

urutannya dan sudah menunjukkan bentuk penyimpanan

3. Buat Diagram Detail (Rinci)

Dikembangkan untuk menciptakan diagram anak yang lebih mendetail.

2.2.2 Kamus Data (*Data Dictionary*)

Menurut Sukanto (2011:73) kamus data adalah “katalog fakta tentang data dan kebutuhan informasi dari suatu sistem informasi yang mendefinisikan data yang mengalir pada sistem secara lengkap”. Fungsi dari *data dictionary* adalah untuk menjelaskan lebih detail tentang *DFD* yang mencakup proses, *data flow* dan *data store*.

Hal yang harus dimuat dalam kamus data adalah :

1. Arus Data

Arus data menunjukkan dari mana data mengalir dan kemana data menuju. Keterangan arus data ini perlu dicatat dikamus data agar memudahkan mencari arus data didalam *DFD*.

2. Nama Arus Data

Karena kamus data dibuat berdasarkan arus data yang mengalir di *DFD*, maka nama arus harus dicatat dikamus data, sehingga mereka yang membaca *DFD* dan memerlukan penjelasan lebih detail tentang suatu arus data tertentu di *DFD* dapat langsung mencarinya dengan mudah pada kamus data.

3. Tipe Data

Data yang mengalir biasanya dalam bentuk laporan serta dokumen hasil cetakan komputer. Bentuk data ini perlu di catat pada kamus data.

4. Struktur Data

Struktur data menunjukkan arus data yang di catat pada kamus data yang terdiri dari item-item atau elemen-elemen data.

5. Alias

Alias atau nama lain dari data yang harus ditulis. Alias perlu ditulis karena data yang sama mempunyai nama perbedaan orang atau *departement* yang satu dengan yang lainnya.

6. Volume

Volume yang perlu dicatat adalah volume rata-rata dari arus data. Volume rata-rata menunjukkan data yang mengalir dalam periode tertentu.

7. Periode

Periode ini menunjukkan kapan terjadinya arus data yang dicatat dikamus data karena dapat digunakan untuk mengidentifikasi kapan *input* data harus dimasukkan ke dalam sistem.

8. Penjelasan

Untuk lebih menjelaskan lagi tentang makna dari arus data yang di catat di kamus data, maka bagian penjelasan dapat di isi dengan keterangan-keterangan tentang arus data tersebut.

Kamus data juga mempunyai bentuk untuk mempersingkat arti makna dari simbol yang di jelaskan yang disebut dengan notasi. Dalam notasi kamus data

lebih mudah menjelaskan data di bandingkan dengan narasi. Notasi atau simbol yang di gunakan di bagi menjadi dua macam, yaitu :

1. Notasi Tipe Data

Digunakan untuk membuat spesifikasi *format input* maupun *output* suatu data. Notasi yang digunakan sebagai berikut

Tabel II.I Notasi Tipe Data

NOTASI	ARTI
X	Setiap Karakter
9	Angka Numerik
A	Karakter Alphabet
Z	Angka Nol Ditampilkan Sebagai Spasi Kosong
.	Sebagai Pemisah Ribuan
,	Sebagai Pemisah Pecahan
-	Sebagai Tanda Penghubung
/	Sebagai Tanda Pembagi

Sumber: Kendall (2010:344)

2. Notasi Struktur Data

Notasi ini digunakan untuk membuat spesifikasi elemen data. Notasi yang umum digunakan antara lain :

Tabel II.2 Notasi Struktur Data

NOTASI	ARTI
=	Terdiri Dari
+	<i>And</i> (dan)
()	Pilihan (Boleh Ya atau Tidak)
{ }	Iterasi (Penghubung Proses)
[]	Pilih Salah Satu Pilihan
I	Pemisah Pilihan Diantara Tanda []
*	Keterangan Atau Catatan
@	Petunjuk (<i>Key Field</i>)

Sumber: Kendall (2010:338)

2.2.2. *Entity Relationship Diagram*

Menurut Fathansyah (2012:79) menyatakan bahwa : Entity Relationship Diagram (ERD) merupakan model *entity relationship* yang berisi komponen-komponen himpunan entitas dan himpan relasi yang masing-masing diegkapi dengan *atribute-atribute* yang mempresentasikan seluruh fakta dari dunia nyata yang kita tinjau.

ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD. ERD memiliki beberapa aliran notasi seperti

notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow's Foot, dan notasi lain. Namun yang banyak digunakan adalah notasi dari Chen

Entity Relationship Diagram (ERD) terdiri dari beberapa komponen diantaranya sebagai berikut:

a. Entitas

Entitas merupakan individu yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Sederhananya entitas menunjuk pada individu suatu objek, sedang himpunan entitas menunjuk pada rumpun (family) dari individu tersebut.

b. Atribut

Setiap entitas pasti memiliki atribut yang mendeskripsikan karakteristik (Property) dari entitas tersebut. Penentuan atau pemilihan atribut-atribut yang relevan bagi sebuah entitas merupakan hal yang penting lainnya dalam pembentukan model data.

c. Relasi

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

d. Garis

Sebagai penghubung antara himpunan relasi dengan himpunan entitas dengan himpunan atributnya.

Entity Relationship Diagram (ERD) memiliki Kardinalitas Relasi diantaranya sebagai berikut:

a. Satu ke satu (*One to One*)

Yang berarti setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B =, dan begitu juga sebaliknya setiap entitas pada himpunan entitas B berhubungan dengan paling banyak dengan entitas A.

b. Satu ke banyak (*One to Many*)

Yang berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya, dimana setiap entitas pada himpunan B berhubungan dengan paling banyak satu entitas pada himpunan B.

c. Banyak ke banyak (*Many to Many*)

Yang berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, dan demikian juga sebaliknya, dimana setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A.

2.2.3. Logical Record Structure

Menurut Anggit Dwi (2012:22) mengungkapkan bahwa “LRS terdiri dari *link-link* diantara tipe *record*. *Link* ini menunjukkan arah dari satu tipe *record* lainnya “. Sebuah model sistem yang digambarkan dengan sebuah diagram-ER akan mengikuti pola / aturan pemodelan tertentu dalam kaitannya dengan konversi ke LRS, maka perubahan yang terjadi adalah mengikuti aturan - aturan berikut ini : Setiap entitas akan diubah kebentuk kotak, Sebuah atribut relasi disatukan dalam sebuah kotak bersama entitas jika hubungan yang terjadi pada diagram-ER 1:M (relasi bersatu dengan *cardinality* M) atau tingkat hubungan 1:1 (relasi bersatu dengan *cardinality* yang paling membutuhkan referensi),

sebuah relasi dipisah dalam sebuah kotak tersendiri (menjadi entitas baru) jika tingkat hubungannya M:M (*many to many*) dan memiliki *foreign key* sebagai *primary key* yang diambil dari kedua entitas yang sebelumnya saling berhubungan”.

2.2.4. Struktur Kode

Fathansyah (2015:34) mengemukakan bahwa “struktur kode Bertujuan untuk mengklasifikasikan data, memasukkan data kedalam komputer dan untuk mengambil bermacam-macam informasi yang berhubungan dengan data tersebut. Kode data di bentuk dari kumpulan angka, huruf dan karakter-karakter khusus”.

Dalam merancang kode yang baik ada beberapa hal yang harus diperhatikan, yaitu sebagai berikut :

1. Harus mudah di ingat

Agar kode mudah diingat, maka dapat dilakukan dengan cara menghubungkan kode tersebut dengan obyek yang di wakili dengan kodenya.

2. Harus unik

Kode harus unik untuk masing – masing item yang di wakili. Unik berarti tidak ada kode yang kembar.

3. Harus fleksibel

Kode harus fleksibel sehingga memungkinkan perubahan – perubahan atau penambahan item baru tetap dapat diwakili oleh kode.

4. Harus efisien

Kode harus sependek mungkin, sehingga mudah diingat dan juga akan

efisien bila di rekam atau di simpan di dalam komputer.

5. Harus konsisten

Kode harus konsisten dengan kode yang telah di gunakan.

6. Harus distandarisasi

Kode harus di standarisasi untuk seluruh tingkatan dan departement dalam organisasi. Kode yang tidak standart akan mengakibatkan kebingungan, salah pengertian dan cenderung dapat terjadi kesalahan pemakai begitu juga dengan yang menggunakan kode tersebut.

7. Harus menghindari spasi

Spasi dalam kode sebaiknya dihindari, karena dapat menyebabkan kesalahan dalam menggunakannya. Karakter – karakter yang hampir serupa bentuk dan bunyi pengucapannya sebaiknya tidak di gunakan dalam kode.

Ada beberapa macam tipe kode yang dapat di gunakan dalam sistem informasi , antara lain :

a. Kode Mnemonik (*Mnemonik Code*)

Bertujuan supaya mudah diingat, di buat dengan dasar singkatan atau mengambil sebagian karakter dari item yang akan di wakili dengan kode ini.

b. Kode Urut (*Sequential Code*)

Disebut juga dengan kode seri, merupakan kode yang nilainya urut antara satu kode dengan kode berikutnya.

c. Kode Blok (*Block Code*)

Mengkalsifikasikan item ke dalam kelompok blok tertentu yang mencerminkan satu klasifikasi tertentu atas dasar pemakaian maksimum yang di harapkan.

d. Kode Grup (*Group Code*)

Kode yang berdasarkan field – field dan tiap – tiap field kode mempunyai arti tertentu.

e. Kode Desimal (*Decimal Code*)

Mengklasifikasikan kode atas dasar 10 unit angka desimal di mulai dari angka 0 sampai dengan angka 9 atau dari angka 00 sampai dengan angka 99 tergantung dari banyaknya kelompok.

2.2.5. Website

Di dalam sebuah *website*, ada beberapa komponen pendukung agar sebuah *website* dapat di akses oleh pemakai, diantaranya :

1. *Internet*

Menurut Febrian (2008:27) menyimpulkan bahwa “*internet* yang kita kenal pertama kali dikembangkan tahun 1969 dengan nama ARPAnet (*USDefense Advanced Reserch Projects Agency*) oleh Departemen Pertahanan Amerika Serikat”.

Menurut Sibero (2013:10) menerangkan bahwa “*internet (Interconnected Network)* adalah jaringan komputer yang menghubungkan antar jaringan secara global, internet daapat juga disebut jaringan dalam suatu jaringan yang luas”.

2. *Web Browser*

Menurut Arief (2011:19) menyimpulkan bahwa “*Web Browser* merupakan program yang berfungsi untuk menampilkan dokumen-dokumen *web* dalam format *HTML*”.

Menurut Sibero (2013:12) berpendapat bahwa “*Web Browser* adalah aplikasi perangkat lunak yang di gunakan untuk mengambil dan menyajikan sumber informasi *web*. Sumber informasi *web* diidentifikasi dengan *Uniform Resource Identifier* (URI) yang terdiri dari halaman *web*, video, gambar ataupun konten lainnya”. Dari pendapat tadi, dapat disimpulkan bahwa *webbrowser* adalah pogram aplikasi yang di gunakan untuk menampilkan informasi berupa *video*, gambar ataupun konten lainnya yang di baca dari format *HTML*.

3. *Web Server*

Menurut Sibero (2013:11) mengungkapkan bahwa “*Web Server* adalah sebuah komputer yang terdiri dari perangkat keras dan perangkat lunak. Secara bentuk dan fisik dan serta kerjanya, perangkat keras *webserver* tidak berbeda dengan komputer rumah atau PC, yang membedakan adalah kapasitas dan kapabilitasnya”.

2.2.6. Struktur Navigasi

Menurut Syukriah (2008:3) mengatakan “Struktur Navigasi adalah struktur atau alur dari suatu program”. Sebelum membuat suatu aplikasi maupun *website* sebaiknya kita menentukan struktur navigasi terlebih dahulu. Ada empat macam bentuk dasar dari struktur navigasi yang biasa di gunakan dalam pembuatan aplikasi menurut Syukriah (2008:4), yaitu :

1. Struktur Navigasi Linier

Struktur navigasi linier hanya mempunyai satu rangkaian cerita yang berurut, yang menampilkan satu demi satu tampilan layar secara berurut menurut urutannya. Tampilan yang dapat ditampilkan pada struktur jenis ini

adalah satu halaman sebelumnya atau satu halaman sesudahnya, tidak dapat dua halaman sebelumnya atau dua halaman sesudahnya.

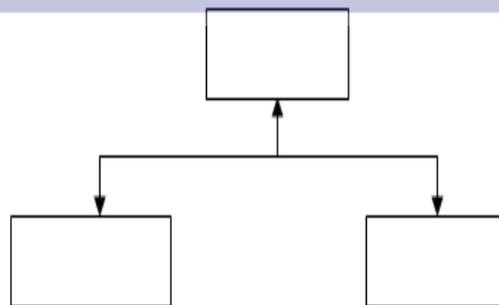


Sumber : Syukriah (2008:4)

Gambar II.1. Struktur Navigasi Linier

2. Struktur Navigasi Hirarki

Struktur navigasi hirarki biasa disebut struktur bercabang, merupakan suatu struktur yang mengandalkan percabangan untuk menampilkan data berdasarkan kriteria tertentu. Tampilan pada menu pertama akan disebut sebagai *master page* (halaman utama pertama). Halaman utama ini mempunyai halaman percabangan yang disebut *Slave Page* (halaman pendukung). Jika salah satu halaman pendukung dipilih atau di aktifkan, maka tampilan tersebut akan bernama *MasterPage* (halaman utama kedua), dan seterusnya.

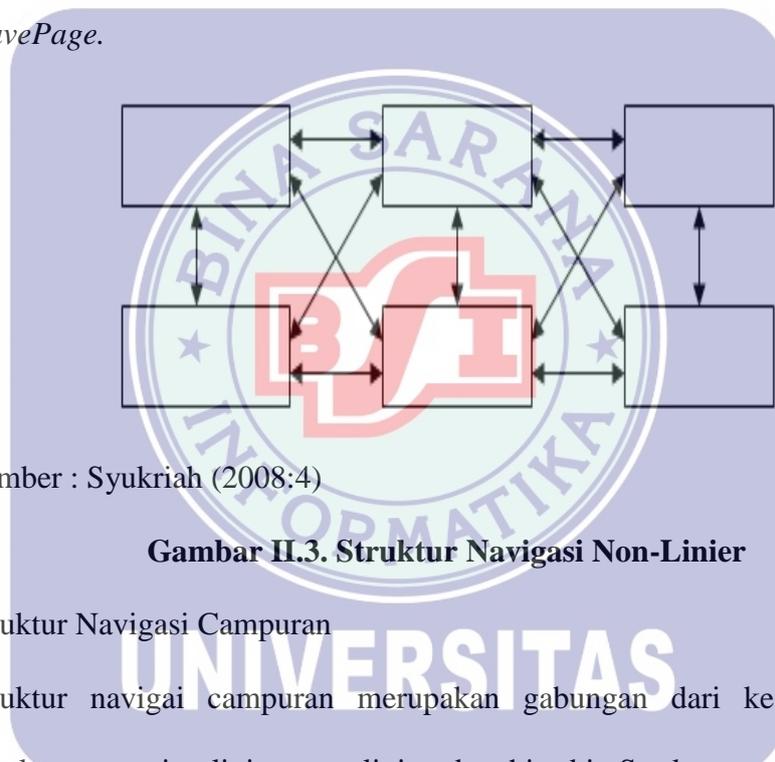


Sumber : Syukriah (2008:5)

Gambar II.2. Struktur Navigasi Hirarki

3. Struktur Navigasi Non-Linier

Struktur navigasi non-linier atau struktur tidak berurut merupakan pengembangan dari struktur navigasi linier. Pada struktur ini di perkenankan membuat navigasi bercabang. Percabangan yang di buat di struktur non-linier ini berbeda dengan percabangan pada struktur hirarki, karena pada percabangan non linier ini walaupun terdapat percabangan, tetapi tiap-tiap tampilan mempunyai kedudukan yang sama yaitu tidak ada *Master Page* dan *SlavePage*.

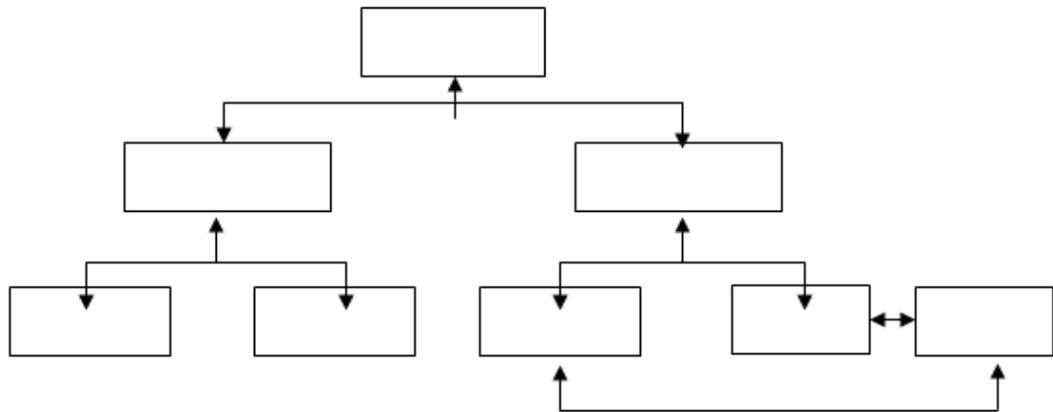


Sumber : Syukriah (2008:4)

Gambar II.3. Struktur Navigasi Non-Linier

4. Struktur Navigasi Campuran

Struktur navigasi campuran merupakan gabungan dari ke tiga struktur sebelumnya yaitu linier, non-linier dan hirarki. Struktur navigasi ini juga biasa disebut dengan struktur navigasi bebas. Struktur navigasi ini banyak digunakan dalam pembuatan *website* karena struktur ini dapat digunakan dalam pembuatan *website* sehingga dapat memberikan interaksi yang lebih tinggi.



Sumber: Syukriah (2008:5)

