

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Sistem ada yang menggunakan pendekatan pada komponen-komponen yang membangunnya, ada juga sistem yang menggunakan pendekatan pada prosedur-prosedurnya. Hal tersebut tergantung dari sudut pandang pengguna sistem dan kinerja dari sistem itu sendiri. maka dari itu diperlukan pemahaman yang lebih lanjut yang menguraikan tentang sistem dan prosedur/komponen yang terkait menggunakan konsep dasar sistem yang menjelaskan tentang pengertian sistem, karakteristik sistem, klasifikasi sistem, informasi, dan sistem informasi.

2.1.1. Pengertian Sistem

Jaringan kerja dari prosedur/komponen menekankan urutan operasi-operasi yang tertampung di dalam sistem. Sistem melibatkan beberapa orang pelaku di dalam satu atau lebih departemen yang diterapkan untuk menjamin penanganan yang lebih baik untuk mengatasi transaksi-transaksi bisnis.

Menurut Hutahaean (2015:2) mengemukakan bahwa “sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran yang tertentu”.

Mulyani (2016:2) mengatakan bahwa “sistem bisa diartikan sebagai sekumpulan sub sistem, komponen ataupun *element* yang saling bekerjasama dengan tujuan yang sama untuk menghasilkan *output* yang sudah ditentukan sebelumnya”.

Dari kedua pandangan para ahli di atas, dapat disimpulkan bahwa sistem merupakan sekumpulan elemen, komponen atau prosedur yang saling bekerja sama untuk mencapai tujuan/sasaran tertentu yang menghasilkan keluaran yang sudah ditentukan sebelumnya.

2.1.2. Karakteristik Sistem

Sistem itu dikatakan sebagai sistem yang baik apabila memiliki karakteristik-karakteristik tertentu, adapun karakteristik tersebut (Hutahaean, 2015:3), yaitu:

1. Komponen

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

2. Batasan sistem

Daerah yang membatasi antara suatu sistem dengan sistem lainnya dinamakan dengan batasan sistem. Batasan sistem memungkinkan sistem dipandang sebagai satu kesatuan.

3. Lingkungan luar sistem

Lingkungan luar adalah apapun yang berada di luar batas sistem yang mempengaruhi sistem, lingkungan luar yang bersifat menguntungkan wajib dipelihara dan lingkungan luar yang bersifat negatif wajib dikendalikan atau dikontrol.

4. Penghubung sistem

Media penghubung ini berguna untuk menghubungkan antara suatu sub sistem dengan sub sistem lainnya, penghubung ini memungkinkan sumber daya mengalir secara tepat antar sub sistem.

5. Masukkan sistem

Apapun yang masuk di dalam sistem dinamakan dengan masukkan sistem. Masukan ini dapat berupa data-data atau berupa perawatan terhadap sistem.

6. Keluaran sistem

Hasil dari masukkan diolah oleh pengolah dan menghasilkan keluaran yang digunakan sebagai sumber informasi bagi pengguna sistem.

7. Pengolah sistem

Sistem dapat berperan sebagai pengolah data-data yang masuk menjadi keluaran yang berguna.

8. Sasaran sistem

Tujuan dan sasaran sangat menentukan masukkan yang dibutuhkan oleh sistem dan apa saja keluaran yang dihasilkan.

2.1.3. Klasifikasi Sistem

Sistem dapat diklasifikasikan dalam beberapa sudut pandang, adapun klasifikasi sistem (Hutahaean, 2015:6), yaitu:

1. Sistem abstrak dan sistem fisik

Sistem dikatakan abstrak apabila sistem tersebut berupa pemikiran atau ide yang tidak tampak secara fisik, sedangkan sistem fisik merupakan sistem tampak, atau ada secara fisik.

2. Sistem alamiah dan sistem buatan manusia

Sistem alamiah adalah sistem yang terjadi karena proses alam dan tanpa campur tangan manusia, sedangkan sistem buatan manusia adalah sistem yang tercipta oleh manusia yang melibatkan interaksi antara manusia dan mesin.

3. Sistem tertentu dan sistem tak tentu

Sistem tertentu merupakan sistem yang beroperasi dengan tingkah laku sudah diprediksi dan ditentukan, sedangkan sistem tak tentu adalah sistem yang beroperasi dengan tingkah laku yang tidak dapat diramalkan.

4. Sistem tertutup dan sistem terbuka

Sistem tertutup merupakan sistem yang tidak terpengaruh oleh lingkungan luar, namun faktanya tidak ada sistem yang benar-benar tertutup, sedangkan sistem terbuka adalah sistem yang berhubungan dengan lingkungan luar dan terpengaruh oleh lingkungan luar tersebut.

2.1.4. Informasi

Informasi akan menjadi berguna apabila objek yang menerima informasi membutuhkan informasi tersebut. Informasi ini juga dapat dijadikan sebagai landasan dalam pengambilan keputusan.

Mulyani (2016:17) mendefinisikan bahwa “informasi merupakan data yang sudah diolah yang ditujukan untuk seseorang, organisasi ataupun siapa saja yang membutuhkan”.

Menurut Djahir dan Pratita (2015:10) mengutarakan bahwa “informasi merupakan hasil dari pengolahan data menjadi bentuk yang lebih berguna bagi yang menerimanya yang menggambarkan suatu kejadian-kejadian nyata dan dapat digunakan sebagai alat bantu untuk pengambilan suatu keputusan”.

Berdasarkan kutipan dari para ahli, dapat disimpulkan bahwa informasi merupakan sekumpulan data yang telah diolah menjadi bentuk yang lebih berguna bagi pengguna informasi yang menggambarkan kejadian nyata dan dapat digunakan sebagai dasar pengambilan keputusan.

2.1.5. Sistem Informasi

Sistem berkolaborasi dengan teknologi menghasilkan sistem informasi untuk mencapai fungsi yang lebih efektif dan efisien. Sistem informasi sangat berguna untuk organisasi/perusahaan dengan skala besar.

Menurut Rahmat dalam Djahir dan Pratita (2015:14) menyatakan bahwa “sistem informasi kegiatan atau aktifitas yang melibatkan serangkaian proses, berisi informasi-informasi yang digunakan untuk mencapai tujuan”.

Hutahaean (2015:13) menyatakan bahwa “sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan”.

Dapat disimpulkan bahwa sistem informasi merupakan kolaborasi antara sistem dan teknologi yang melibatkan serangkaian proses berisi informasi-informasi yang dibutuhkan.

Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (Hutahaean, 2015:13), adapun blok bangunan tersebut diuraikan sebagai berikut:

1. Blok masukan

Masukkan mewakili data yang masuk ke dalam sistem informasi biasanya berupa dokumen dasar.

2. Blok model

Blok model merupakan kombinasi dari prosedur, logika dan metode matematik yang digunakan untuk memanipulasi data yang disimpan di dalam

basis data dengan cara yang sudah ditentukan sebelumnya untuk menghasilkan keluaran yang diinginkan.

3. Blok keluaran

Hasil dari sistem informasi ini adalah keluaran yang merupakan informasi berkualitas dan tingkat dokumentasi yang berguna untuk semua tingkatan manajemen/pemakai sistem.

4. Blok teknologi

Teknologi digunakan untuk menerima masukan, menjalankan model dan menyimpan data untuk menghasilkan keluaran yang diinginkan serta membantu pengendalian sistem secara keseluruhan.

5. Blok basis data

Data-data yang masuk akan tersimpan di dalam basis data yang dapat digunakan kembali serta tersimpan di dalam perangkat keras.

6. Blok kendali

Pengendalian perlu dirancang untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dikontrol atau dikendalikan.

2.1.6. Metode Pengembangan Perangkat Lunak

Menurut Sukamto dan Shalahuddin (2015:28), “Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain pengodean, pengujian dan tahap pendukung (*support*)”. Metode air terjun menurut Sukamto dan Shalahuddin (2015:29) yaitu:

1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara insentif 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 fokus 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. Pembuatan Kode Program

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

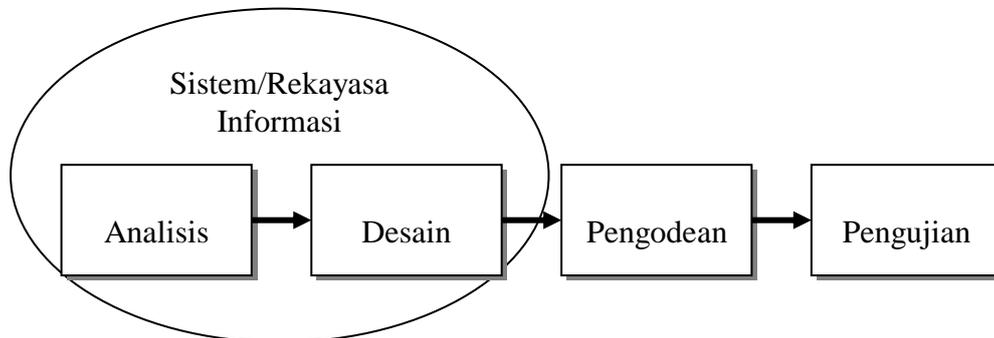
4. Pengujian

Pengujian fokus pada perangkat lunak secara ad-hoc dari segi *logic* dan fungsional serta memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5. Pendukung (*support*) atau Pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. 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 sudah ada, tapi tidak untuk membuat perangkat lunak baru.



Sumber : Salim dan Shalahuddin (2015:29)

Gambar II.1. Model *Waterfall*

2.1.7. Basis Data

Basis data berguna sebagai tempat pengolahan informasi yang sangat penting dalam upaya menciptakan suatu aplikasi yang terintegrasi. Basis data biasa tersimpan di perangkat keras dan memerlukan suatu aplikasi untuk mengolah basis data tersebut.

Sukanto dan Shalahuddin (2015:43) menyatakan bahwa “sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan”.

Menurut Lubis (2016:2) mengemukakan bahwa “basis data merupakan gabungan *file* data yang dibentuk dengan hubungan/relasi yang logis dan dapat diungkapkan dengan catatan serta bersifat independen”.

Dapat ditarik kesimpulan bahwa basis data merupakan kumpulan data yang diolah menjadi informasi dan dapat digunakan kembali jika dibutuhkan dalam suatu organisasi atau perusahaan.

2.2. Teori Pendukung

Teori-teori pendukung ini menjelaskan atau menguraikan tentang teknik-teknik/alat-alat yang digunakan untuk merancang sistem. Adapun teori pendukung tersebut terdiri dari diagram alir data, kamus data, ERD, LRS, dan struktur kode.

2.2.1. Diagram Alir Data (DAD)

Pengembang atau *programmer* menggunakan diagram alir data (DAD) untuk memodelkan suatu sistem yang akan dianalisa atau dirancang. Diagram alir data ini biasa juga disebut dengan *data flow diagram* (DFD).

Menurut Sukamto dan Shalahuddin (2015:70) “DFD tidak sesuai untuk memodelkan sistem yang menggunakan pemrograman berorientasi objek”.

McLeod dan Schell (2008:214) mengemukakan bahwa diagram alir data: Adalah penyajian grafis dari sebuah sistem yang mempergunakan empat bentuk simbol untuk mengilustrasikan bagaimana data mengalir melalui proses-proses yang saling tersambung. Simbol-simbol tersebut mencerminkan (1) unsur-unsur lingkungan dengan mana sistem berinteraksi, (2) proses, (3) arus data, dan (4) penyimpanan data.

Berdasarkan kutipan dari para ahli di atas, dapat disimpulkan bahwa diagram alir data merupakan teknik penyajian berupa grafis/gambar yang memodelkan bagaimana data mengalir di dalam sistem yang saling terkait dan tersambung.

Simbol atau lambang yang digunakan dalam membuat diagram alir data yang lazim digunakan, menurut (Sukamto dan Shalahuddin, 2015:71) DAD terdiri dari empat buah simbol yaitu:

1. Entitas/lingkungan luar (*external entity*)

Menggambarkan asal tujuan data, menunjukkan entitas atau kesatuan yang berhubungan dengan sistem, dapat berupa orang , organisasi atau sistem lainnya yang akan memberikan *input* atau menerima *output* dari sistem atau keduanya.

2. Proses (*process*)

Simbol ini digunakan untuk proses pengolahan atau transformasi data, menunjukkan kegiatan atau kerja yang dilakukam oleh orang, mesin atau komputer dan hasil suatu data yang masuk kedalam proses untuk menghasilkan arus data yang akan keluar dari proses.

3. Arus data (*data flow*)

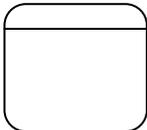
Simbol ini digunakan untuk menggambarkan aliran data yang berjalan, menunjukkan arus data yang berupa masukan untuk sistem atau hasil dari proses sistem yang mengalir diantara proses (*process*), simpanan data (*data store*) dan entitas (*external entity*).

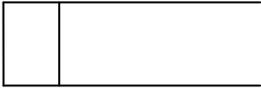
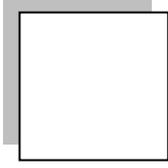
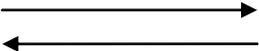
4. Simpanan data (*data store*)

Simbol ini digunakan untuk menggambarkan data flow yang sudah disimpan, menunjukkan suatu tempat penyimpanan data yang berupa suatu file di sistem komputer, arsip atau catatan manual, tabel acuan dan lain-lain.

Tabel II.1.

Simbol-Simbol Diagram Alir Data

Simbol	Keterangan
<p>Proses (<i>Process</i>)</p> 	<p>Bujursangkar dengan sudut membulat digunakan untuk menunjukan adanya proses transformasi dan aliran data yang meninggalkan suatu proses selalu diberi label yang berbeda dari aliran data yang masuk. Biasa berupa kata kerja.</p>

<p>Data Store</p> 	<p>Persegi Panjang degan ujung terbuka yang menunjukkan data. Dalam diagram alir data logika, jenis penyimpanan fisik (misal disk) tidak ditetapkan. Penyimpanan dapat berupa penyimpanan manual, seperti lemari <i>file</i>, atau sebuah <i>file</i> atau basis data terkomputerisasi. Biasa berupa kata benda.</p>
<p>External Entity</p> 	<p>Kotak rangkap dua digunakan untuk menggambarkan suatu entitas eksternal, misalnya sebuah perusahaan, seseorang atau sebuah mesin yang dapat mengirim data atau menerima data dari sistem dan merupakan sumber atau tujuan data. Entitas diberi nama dengan kata benda untuk melambangkan pelaku sistem.</p>
<p>Arus Data (Data Flow)</p> 	<p>Tanda panah, menunjukkan perpindahan data dari satu titik ke titik lain, aliran data yang muncul secara simultan bisa digambarkan dalam kata benda.</p>

Sumber: Sukamto dan Shalahuddin (2015:71)

Diagram alir data (DAD) memiliki beberapa tingkatan dalam teknik penggambarannya. Adapun tingkatan diagram alir data tersebut (Ladjamudin, 2015:64) yaitu:

1. Diagram konteks

Diagram konteks digunakan untuk menggambarkan ruang lingkup sistem secara keseluruhan dan merupakan level tertinggi/pertama dari diagram alir data.

2. Diagram nol/zero (*overview diagram*)

Diagram nol digunakan untuk menggambarkan pandangan sistem secara menyeluruh mengenai sistem yang ditangani dan data yang mengalir dan merupakan level kedua dari diagram alir data atau menjelaskan lebih rinci tentang diagram konteks.

3. Diagram rinci (*level diagram*)

Diagram rinci digunakan untuk menguraikan secara spesifik atau rinci tentang proses apa saja yang terjadi pada diagram nol per tahapannya.

2.2.2. Kamus Data

Data-data yang mengalir pada diagram alir data (DAD) biasa berupa singkatan atau simbol-simbol. Singkatan dari data yang mengalir atau simbol-simbol tersebut dapat dijelaskan menggunakan kamus data (*data dictionary*).

Menurut Sukamto dan Salahudin (2015:73) “kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat difahami secara umum (memiliki standar cara penulisan)”.

McLeod dan Schell (2008:171), “kamus data (*data dictionary*) mencakup definisi-definisi dari data yang tersimpan dalam basis data dan dikendalikan oleh sistem manajemen basis data”.

Dapat disimpulkan bahwa kamus data merupakan daftar elemen dari suatu data yang mengalir di dalam suatu sistem dan tersimpan di dalam basis data serta memiliki standar penulisan.

Kamus data dalam implementasi program dapat menjadi parameter masukan atau keluaran dari sebuah fungsi atau prosedur. Biasanya kamus data (Sukamto dan Shalahuddin, 2015:74) berisikan:

1. Nama

Kamus data berisikan nama data yang mengalir di DAD.

2. Digunakan

Kamus data digunakan pada proses-proses terkait aliran data.

3. Deskripsi

Deskripsi disini menguraikan data-data yang mengalir menjadi lebih detail.

4. Informasi tambahan

Kamus data biasa berisikan informasi tambahan seperti tipe data, nilai data, batas nilai data, dan komponen yang membentuk data tersebut.

Standar penulisan/symbol-simbol tersebut menurut Rosa dan Shalahuddin (2015:74) dapat dilihat pada tabel berikut ini.

Tabel II.2.

Simbol-Simbol Kamus Data

Simbol	Keterangan
=	Disusun atau terdiri dari
+	Dan
[]	Baik ... atau
{ } ⁿ	n kali diulang/bernilai banyak
()	Data opsional
...	Batas komentar

Sumber: Sukamto dan Shalahuddin (205:74)

2.2.3. *Entity Relationship Diagram (ERD)*

Basis data terbentuk dari kumpulan tabel-tabel/entitas yang saling berelasi. Relasi yang terjadi antar entitas tersebut biasa digambarkan dengan *entity relationship diagram (ERD)*.

Menurut Sukamto dan Shalahuddin (2015:50) Model E-R (ERD) adalah “Penyimpanan basis data menggunakan OODBMS”.

Menurut McLeod dan Schell (2008:172), “diagram relasi entitas digunakan untuk menguraikan hubungan antara kumpulan-kumpulan data konseptual”.

Berdasarkan pernyataan yang telah diuraikan di atas, dapat disimpulkan bahwa *entity relationship diagram (ERD)* merupakan teknik pemodelan perancangan basis data yang menggambarkan hubungan/relasi antar entitas di dalam basis data yang bersifat konseptual.

Komponen/symbol yang digunakan untuk menggambarkan *entity relationship* (Sukamto dan Shalahuddin, 2015:50), yaitu:

1. Entitas

Entitas merupakan data inti yang akan disimpan, bakal tabel pada basis data, benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer. Penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.

2. Atribut

Field atau kolom data yang butuh disimpan dalam suatu entitas.

3. Atribut kunci primer

Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses *record* yang diinginkan, biasanya berupa id. Kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).

4. Atribut multivali

Field atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki lebih dari satu.

5. Relasi

Relasi yang menghubungkan antar entitas, biasanya diawali dengan kata kerja.

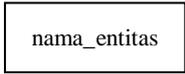
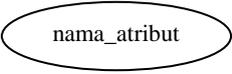
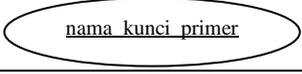
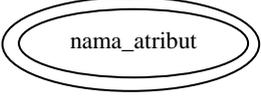
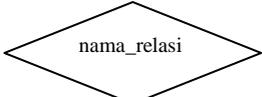
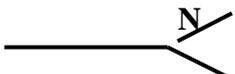
6. Asosiasi

Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki *multiplicity* kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut

dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan *one to many* menghubungkan entitas A dan entitas B.

Tabel II.3.

Komponen-Komponen *Entity Relationship Diagram* (ERD)

Notasi	Keterangan
1. Entitas/ <i>entity</i> 	Entitas merupakan kelompok data dalam <i>database</i> , biasanya disebut dengan tabel. Nama entitas biasanya berupa kata benda.
2. Atribut 	Atribut atau <i>field</i> merupakan bagian dari entitas atau uraian dari data-data yang terdapat pada suatu entitas.
3. Atribut kunci primer 	Atribut kunci primer sebenarnya memiliki pengertian yang sama dengan atribut, namun pada atribut kunci primer ini merupakan atribut kunci yang berada suatu entitas dan nama atribut tersebut diberi garis bawah.
4. Atribut multivalai/ <i>multivalue</i> 	Atribut dari suatu entitas yang memiliki nilai lebih dari satu atau bernilai ganda.
5. Relasi 	Relasi digunakan untuk menghubungkan antara entitas satu dengan entitas lainnya yang memiliki hubungan. Nama relasi biasanya berupa kata kerja.
6. Asosiasi/ <i>association</i> 	Asosiasi merupakan garis penghubung yang menghubungkan antar entitas melalui relasi. Pada asosiasi terdapat <i>multiplicity</i> atau kardinalitas relasi. Seperti 1:1, 1:M, atau M:M.

Sumber: Sukamto dan Shalahuddin (2015:50)

2.2.4. *Logical Record Structure* (LRS)

Selain *entity relationship diagram* (ERD), terdapat teknik pemodelan lain untuk antar entitas di dalam basis data, teknik tersebut biasa disebut dengan *logical record structure* (LRS).

Menurut Kadir (2007:18) *logical record structure* (LRS) merupakan “model relasional menggunakan kumpulan tabel-tabel untuk merepresentasikan data dan relasi antar data-data tersebut. Setiap tabel terdiri atas kolom-kolom dan setiap kolom memiliki nama yang unik”. Sedangkan, menurut Hasugian dan Shidiq (2012:608) memberikan batasan bahwa LRS adalah “sebuah model sistem yang digambarkan dengan sebuah *diagram*-ER akan mengikuti pola atau aturan permodelan tertentu dalam kaitannya dengan konvensi ke LRS”.

Dapat disimpulkan bahwa *logical record structure* (LRS) merupakan teknik pemodelan sistem untuk menguraikan hubungan yang terjadi antar entitas di dalam suatu basis data yang mengikuti aturan pemodelan tertentu sesuai dengan aturan penggambaran LRS.

Di dalam LRS, terdapat kardinalitas dalam penentuan hubungan (*multiplicity*). Adapun kardinalitas tersebut (Ladjamudin, 2013:160) yaitu:

1. 1:1 (*one to one*)

Relasi yang terjadi antara suatu *entity* dengan *entity* lainnya yang memiliki hubungan 1:1.

2. 1:M (*one to many*)

Relasi yang terjadi antara suatu *entity* dengan *entity* lainnya yang memiliki hubungan 1:M.

3. M:N (*many to many*)

Relasi yang terjadi antara suatu *entity* dengan *entity* lainnya yang memiliki hubungan M:N. Pada relasi ini biasa digunakan tabel bantuan untuk memecahkan relasi tersebut menjadi 1:1 atau 1:M.

2.2.5. Struktur Kode

Setiap data memiliki identitas unik untuk menentukan identifikasi dari data tersebut. Pemberian identitas unik ini menggunakan struktur kode. Struktur kode memudahkan dalam pencarian dari setiap data yang ingin dicari.

Shatu (2016:106) mengemukakan bahwa “kode memudahkan proses pengolahan data karena dengan kode, data akan lebih mudah diidentifikasi”.

Sutabri dalam Puspitawati dan Anggadini (2011:96) menyatakan bahwa “sistem pengkodean terdiri dari himpunan karakter, simbol-simbol yang dapat diterima dan telah dinyatakan digunakan untuk mengidentifikasi objek tertentu”.

Dapat disimpulkan bahwa struktur kode merupakan kumpulan dari karakter, simbol, alfabetik yang membentuk serangkaian kode yang digunakan untuk proses pengolahan data agar lebih mudah diidentifikasi.

Dalam pembuatan sebuah kode kode yang baik memiliki persyaratan-persyaratan tertentu atau faktor-faktor yang perlu dipertimbangkan. Adapun faktor-faktor pertimbangan (Shatu, 2016:107) dalam pembuatan kode yaitu:

1. Kode yang disusun perlu disesuaikan dengan metode proses data.
2. Setiap kode harus mewakili hanya satu *item* sehingga tidak membingungkan.
3. Kode yang disusun harus memudahkan pemakai untuk mengingatnya.
4. Kode yang disusun harus fleksibel, dalam arti memungkinkan dilakukan perluasan tanpa perubahan menyeluruh.
5. Setiap kode harus menggunakan jumlah angka dan huruf yang sama.
6. Kode yang panjang perlu dipotong-potong (*chunking*) untuk memudahkan mengingat.

7. Dalam kode yang panjang perlu diberi kode yang merupakan *check digit*, yaitu untuk mengecek kebenaran kode.

Berikut ini adalah macam-macam kode (Shatu, 2016:108) yang dapat digunakan:

1. Kode urut nomor

Kode yang terbentuk dari susunan angka/nomor. Setiap kode memiliki jumlah angka yang sama (digit).

2. Kode kelompok

Kode kelompok bertujuan untuk membagi data dalam kelompok tertentu. Tiap kelompok akan diberi kode dengan angka atau huruf tertentu, sehingga masing-masing posisi angka/huruf dari kode mempunyai arti.

3. Kode blok

Setiap kelompok data diberi kode dalam blok nomor tertentu. Kode blok mirip dengan kode kelompok.

4. Kode desimal

Setiap kelompok data akan diberi kode dari 0 sampai dengan 9. Oleh karena itu pengelompokan data harus dilakukan maksimum dalam sepuluh kelompok.

5. Kode *mnemonic*

Kode *mnemonic* merupakan kode singkatan data yang digunakan untuk membantu pengguna kode ini dalam membaca maksud dari singkata tersebut.

6. Kode *bar*

Kode *bar* terdiri dari batangan-batangan hitam, biasa digunakan untuk perusahaan makanan dan minuman. Kode ini sebenarnya merupakan

transformasi dari angka menjadi batangan-batangan kode, pembedanya adalah ketebalan dari batangan-batangan (*bar*) tersebut.

2.2.6. *Hierarchy Input Process Output (HIPO)*

Hierarchy Input Process Output (HIPO) sering digunakan oleh para *programmer* dalam mendokumentasikan alur dari rancangan sistem dan komunikasi yang terjadi untuk setiap fungsi sistem.

Menurut Ladjamudin (2013:211), “*Hierarchy-Plus-Input-Process-Output (HIPO)* merupakan teknik untuk mendokumentasikan sistem pemograman”. Sedangkan menurut Fatta (2007:147) “HIPO merupakan teknik untuk mendokumentasikan suatu sistem yang dikembangkan oleh IBM”.

Maka dari itu, dapat disimpulkan bahwa *hierarchy input process output (HIPO)* merupakan suatu cara yang yang dikembangkan/didukung oleh IBM untuk mendokumentasikan program dan banyak digunakan oleh para *programmer* sebagai alat *design* dan teknik dokumentasi dalam pengembangan sistem.