BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Sistem menangani fungsi yang secara terus-menerus dan berulang-ulang atau yang rutin terjadi. Umumnya sistem dibuat sebagai suatu alat untuk membantu dalam meningkatkan kualitas suatu pekerjaan. Konsep dasar sistem menekankan pada dasar pemahaman tentang pengertian, klasifikasi dan karakteristik dari sistem.

2.1.1. Pengertian Sistem

Secara umum suatu sistem dapat diartikan sebagai kumpulan dari elemen, komponen yang terorganisir, berinteraksi, dan bekerja sama untuk mencapai tujuan bersama.

Menurut Diana dan Setiawati (2011:3), "sistem merupakan serangkaian bagian yang saling tergantung dan bekerja sama untuk mencapai tujuan tertentu". Sedangkan menurut Jogiyanto dalam bukunya Yakub (2012:1), "sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk tujuan tertentu". Lain halnya menurut Darmawan (2013:4), "sistem sebagai kumpulan dari beberapa bagian komponen apa pun baik fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan".

Berdasarkan kutipan dari para ahli di atas, dapat disimpulkan bahwa sistem merupakan suatu jaringan kerja yang terdiri dari kumpulan komponen, prosedur atau variabel yang terorganisir, saling berinteraksi, berhubungan dan

bekerja sama untuk menangani dan menyelesaikan sesuatu serta mencapai tujuan yang diinginkan secara rutin atau berulang. Komponen, prosedur atau variabel yang dimaksud tidak terbatas pada bentuk fisik tetapi juga hal yang bersifat abstrak atau konseptual seperti visi, misi, dan lain-lain.

A. Karakteristik Sistem

Setiap sistem memiliki ciri-ciri atau karakteristik yang mencirikan bahwa hal tersebut bisa dikatakan sebagai sistem. Adapun karakteristik menurut Sutabri (2012:21) yang dimaksud adalah sebagai berikut:

1. Komponen Sistem (*Component*)

   Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2. Batasan Sistem (*Boundary*)

   Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisahkan.

3. Lingkungan Luar Sistem (*Environment*)

   Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut lingkungan luar sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4. Penghubung (*Interface*)

   Media yang menghubungkan sistem dengan subsistem yang lain disebut penghubung sistem atau *interface*. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lain. Bentuk keluaran dari satu subsistem akan menjadi masukan untuk subsistem lain melalui penghubung tersebut. Dengan demikian, dapat terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. Masukan Sistem (*Input*)

   Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*).

6. Keluaran Sistem (*Output*)

   Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini dapat menjadi masukan bagi subsistem yang lain seperti sistem informasi. Keluaran yang dihasilkan adalah informasi. Informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang menjadi input bagi subsistem lain.

7. Pengolah Sistem (*Process*)

   Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

8. Sasaran (*Objectives*)

   Suatu sistem mempunyai tujuan dan sasaran yang pasti dan bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

9. Tujuan (*Goal*)

   Suatu sistem dikatakan berhasil apabila mengenai sasaran atau tujuannya. Jika suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya.

B. Klasifikasi Sistem

Suatu sistem akan terlihat berbeda, ada yang tampak secara fisik, ada juga yang bersifat abstrak. Semua itu tergantung dari sudut pandang pengguna sistem dalam menentukan klasifikasi dari sistem tersebut. Adapun klasifikasi sistem menurut Mulyanto (2009:8) yaitu:

1. Sistem Abstrak dan Sistem Fisik
   a. Sistem abstrak (*abstract system*) merupakan sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik (sistem teologia yang merupakan suatu sistem yang menggambarkan hubungan antara Tuhan dengan manusia).
   b. Sistem Fisik (*Physical System*) diartikan sebagai sistem yang nampak secara fisik sehingga setiap mahluk dapat melihatnya, misalnya sistem komputer, sistem akuntansi, sistem produksi, dan lain-lain.
2. Sistem Alamiah dan Sistem Buatan Manusia
   a. Sistem Alamiah (*Natural System*) merupakan sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem tata surya, sistem galaksi, sistem reproduksi dan lain-lain.
   b. Sistem Buatan Manusia (*Human Made System*) merupakan sistem yang dirancang oleh manusia. Sistem buatan yang melibatkan interaksi manusia, misalnya sistem akuntansi, sistem informasi, dan lain-lain.

3. Sistem Deterministik dan Sistem Probabilistik
    a. Sistem Deterministik (*Deterministic System*) merupakan sistem yang beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan, misalnya sistem komputer, adalah contoh sistem yang tingkah lakunya dapat dipastikan berdasarkan program-program komputer yang dijalankan.
    b. Sistem Probabilistik (*Probabilistic System*) merupakan sistem yang kondisi masa depanya tidak dapat diprediksi karena mengandung unsur probabilitas, misalnya sistem manusia.
4. Sistem Terbuka dan Sistem Tertutup
    a. Sistem Terbuka (*Open System*) merupakan sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Lebih sepesifik dikenal juga yang disebut dengan sistem terotomasi, yang merupakan bagian dari sistem buatan manusia dan beriteraksi dengan kontrol oleh satu atau lebih komputer sebagai bagian dari sistem yang digunakan dalam masyarakat modern. Sistem ini menerima masukan dan menghasilkan keluaran untuk subsistem lainnya, misalnya sistem kebudayaan manusia.
    b. Sistem Tertutup (*Closed System*) merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa danya campur tangan dari pihak luar. Secara teoritis sistem tersebut ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system* (secara relatif tertutup, tidak benar-benar tertutup).

2.1.2. Sistem Informasi

Sistem menghasilkan informasi yang bersumber dari data-data yang telah diolah, dimana informasi tersebut akan digunakan sebagai dasar pertimbangan suatu perusahaan untuk menentukan langkah ke depan.

A. Informasi

Informasi merupakan hasil dari pengolahan terhadap data yang memiliki nilai tertentu dan bisa digunakan untuk menambah pengetahuan bagi penerimanya.

Menurut Jogiyanto, (2009:36), ”Informasi adalah data yang diolah menjadi bentuk yang berguna bagi pemakainya”. Sedangkan menurut Darmawan (2013:2), “Informasi adalah sejumlah data yang sudah diolah atau diproses melalui prosedur pengolahan data dalam rangka menguji tingkat kebenarannya”.

Berdasarkan beberapa pendapat para ahli di atas maka dapat ditarik kesimpulan bahwa informasi merupakan sejumlah data yang telah diolah menjadi bentuk yang lebih berguna.

B. Pengertian Sistem Informasi

Sistem informasi sering digunakan sebagai jaringan kerja suatu perusahaan agar kinerja yang dihasilkan lebih menjanjikan dan praktis.

Menurut Mulyanto (2009:29), “Sistem informasi adalah suatu komponen yang terdiri dari manusia, teknologi informasi, dan prosedur kerja yang memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk mencapai suatu tujuan”. Sedangkan menurut Sutarman (2012:13), “Sistem informasi adalah sistem dapat didefinisikan dengan mengumpulkan, memperoses, menyimpan, menganalisis, menyebarkan informasi untuk tujuan tertentu”.

Dari pendapat di atas dapat disimpulkan bahwa sistem informasi merupakan gabungan dari manusia, *hardware, software,* jaringan komunikasi dan data yang saling berinteraksi untuk menyimpan, mengumpulkan, memproses, dan mendistribusikan informasi untuk mendukung pengambilan keputusan dalam suatu organisasi.

2.1.3. Basis Data

Setiap sistem informasi memiliki basis data sebagai media pengelola data-data yang tersimpan. Data-data yang tersimpan dalam basis data dapat digunakan kembali sesuai keinginan penggunanya.

A. Definisi Basis Data

Basis data merupakan aspek penting dalam sistem informasi, basis data berfungsi sebagai gudang penyimpanan data dan dapat digunakan kembali. Basis data (*database*) dapat diartikan sebagai kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak untuk menghasilkan informasi.

Menurut Hirin dan Virgi (2011:29) memberikan batasan bahwa "*database* atau basis data yaitu sekumpulan informasi atau data secara sistematik sehingga dapat diperiksa oleh program komputer untuk memperoleh informasi dari basis data tersebut".Rosa dan Shalahuddin (2015:43) mengemukakan bahwa "sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan".

Berdasarkan pendapat dari para ahli, dapat disimpulkan bahwa basis data merupakan kumpulan informasi atau data yang bertujuan untuk mengelola data yang sudah diolah atau informasi dan dapat digunakan kembali.

B. Aplikasi Basis Data

Untuk mengelola basis data, diperlukan sebuah aplikasi. Aplikasi yang digunakan oleh penulis dalam penulisan laporan Tugas Akhir ini menggunakan MySQL.

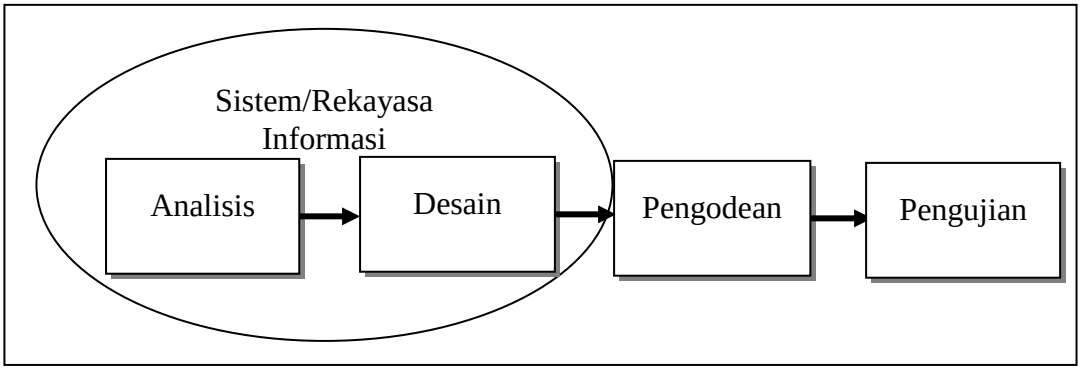
Menurut Arief (2011:152) "*My*SQL adalah salah satu jenis *database* server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi yang menggunakan *database* sebagai sumber dan pengolahan datanya". Menurut Hirin dan Virgi (2011:27) "*My*SQL adalah satu perangkat lunak sistem manajemen basis data (*database*) SQL atau sering disebut dengan DBMS (*Database Management System*)".

Berdasarkan pengertian para ahli, maka *My*SQL merupakan aplikasi manajemen basis data yang digunakan untuk mengelola data yang tersimpan di basis data dan dapat digunakan kembali.

2.1.4. Metode Pengembangan Perangkat Lunak

Pengembangan perangkat lunak digunakan sebagai disiplin ilmu sebagai dasar pemodelan pengembangan perangkat lunak. Penulis menggunakan model air terjun (*waterfall*) sebagai model pengembangan perangkat lunak.

Menurut Rosa 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*)".



Sumber : Rosa dan Shalahuddin (2015:29)

Gambar II.1. Ilustrasi Model *Waterfall*

Adapun penjelasan dari tahapan model air terjun (*waterfall*) menurut Rosa 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 secar adari 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.

2.1.5. Sistem Penggajian

Setiap karyawan yang bekerja akan mendapatkan gaji dari perusahaan. Untuk mengatur penggajian karyawan tersebut diperlukan sistem penggajian. Sistem penggajian merupakan fungsi penting yang menjadi tanggung jawab manajemen sumber daya manusia, dan gaji merupakan pemberian finansial yang diterima oleh para karyawan sebagai ganti kontribusi mereka terhadap perusahaan.

Sistem penggajian dapat diartikan sebagai metode yang dipakai dalam menentukan upah yang didapat oleh pegawai (Mulyadi, 2008:381). Sedangkan menurut Haryadi (2009:104), "Sistem penggajian adalah proses yang menentukan tingkat penggajian pegawai, memantau dan mengawasi, mengembangkan dan mengendalikan gaji pegawai".

Dari pengertian di atas dapat dikatakan sistem penggajian merupakan sebuah proses yang memiliki peran penting bagi sumber daya manusia dimana proses tersebut bertujuan untuk menentukan tingkat penggajian karyawan, memantau dan mengawasi serta mengendalikan gaji pegawai.

2.2. Teori Pendukung

Teori pendukung merupakan pendapat dari para ahli yang digunakan oleh penulis dalam mendukung keberhasilan penulisan laporan Tugas Akhir. Adapun teori pendukung yang digunakan terdiri dari:

2.2.1. Diagram Alir Data (DAD)

Diagram alir data (DAD) digunakan sebagai alat untuk menggambarkan suatu sistem dengan alur data yang terarah. Dalam DAD ini pelaku (entitas), tahapan sistem (proses), data-data yang mengalir (alir data), dan tempat penyimpanan data (*data store*) berhubung menjadi satu kesatuan.

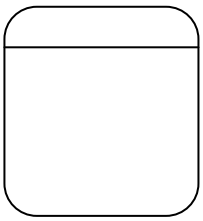
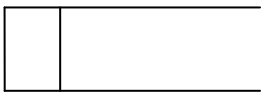
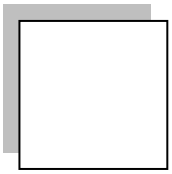
Menurut Indrajani (2011:11) "*Data Flow Diagram* (DFD) adalah sebuah alat yang menggambarkan aliran data sampai sebuah sistem selesai, dan kerja atau proses dilakukan dalam sistem tersebut". Menurut Rosa dan Shalahuddin (2015:70) "DFD tidak sesuai untuk memodelkan sistem yang menggunakan pemrograman berorientasi objek".

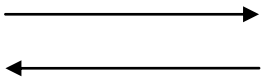
Berdasarkan pendapat dari para ahli di atas, dapat ditarik kesimpulan bahwa *Data Flow Diagram* atau Diagram Arus Data merupakan suatu gambaran gratis dan suatu sistem yang menggunakan sejumlah bentuk-bentuk simbol untuk menggambarkan bagaimana data mengalir melalui suatu proses yang berkaitan.

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

Tabel II.1.

Notasi-Notasi pada Diagram Alir Data

Notasi	Keterangan
Proses (*Process*) 	Proses atau fungsi atau prosedur; Pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program. Catatan: nama yang diberikan pada sebuah proses biasanya berupa kata kerja.
Data Store 	*File* atau basis data atau penyimpanan (*storage*); Pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel pada basis data (*Entity Relationship Diagram* (ERD), *Conceptual Data Model* (CDM), *Physical Data Model* (PDM). Catatan: nama yang diberikan pada sebuah penyimpanan biasanya kata benda.
External Entity 	Entitas luar (*external entity*) atau masukan (*input*) atau keluaran (*output*) atau orang yang memakai/berinteraksi dengan perangkat lunak yang terkait dengan aliran data dari sistem yang dimodelkan. Catatan: nama yang digunakan pada masukan (*input*) atau keluaran (*output*) biasanya berupa kata benda.

Arus Data (*Data Flow*) 	Aliran data; Merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (*input*) atau keluaran (*output*). Catatan: nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya "data siswa" atau tanpa kata misalnya "siswa".

Sumber: Rosa dan Shalahuddin (2015:71)

Tahap pembuatan diagram alir data (DAD) Menurut Kristanto (2008:70) dibagi menjadi tiga tingkatan kontruksi diagram alir data (DAD) yaitu :

1. Diagram Konteks

   Diagram ini dibuat untuk menggambarkan sumber serta tujuan data yang akan diproses atau dengan kata lain diagram tersebut untuk menggambarkan sistem secara global dari keseluruhan sistem yang ada.

2. Diagram Nol

   Diagram ini dibuat untuk menggambarkan tahap-tahap proses yang akan ada didalam konteks atau penjabaran secara rinci.

3. Diagram Detail

   Diagram ini dibuat untuk menggambarkan arus data secara lebih detail dan terperinci dari tahapan proses yang ada dalam diagram.

2.2.2. *Entity Relationship Diagram* (ERD)

Dalam perancangan basis data, ERD merupakan teknik pemodelan yang sering digunakan dalam menggambarkan basis data yang sedang dirancang.

A. Pengertian *Entity Relationship Diagram* (ERD)

ERD adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak

Menurut Ladjamudin (2013:142) "*Entity Relationship Diagram* (ERD) adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak". Menurut Rosa dan Shalahuddin (2015:53) "ERD adalah bentuk paling awal dalam melakukan perancangan basis data relasional. Jika menggunakan OODMBS maka perancangan ERD tidak perlu dilakukan".

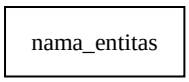
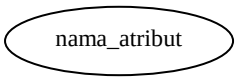
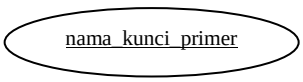
Dapat ditarik kesimpulan bawa ERD merupakan bentuk awal perancangan basis data yang menggunakan simbol-simbol yang terdiri dari entitas dan hubungannya.

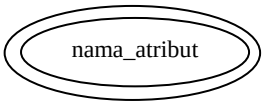
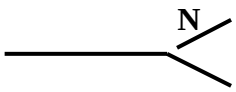
B. Komponen *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram (ERD) memiliki simbol-simbol atau komponen-komponen dalam teknik penggambarannya. Adapun simbol-simbol atau komponen-komponen pada *Entity Relationship Diagram* (ERD) dapat dilihat pada tabel di bawah ini.

Tabel II.2.

Komponen *Entity Relationship Diagram* (ERD)

Notasi	Keterangan
1. Entitas/*entity* 	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 multinilai/ *multivalue* 	*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/*association* 	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.

Sumber: Rosa dan Shalahuddin (2015:50)

C. Derajat *Relationship*

Menurut Indrajani (2011:110) mengemukakan bahwa "Derajat *Relationship* adalah jumlah entitas yang berpartisipasi dalam suatu *relationship*". Sedangkan menurut Ladjamudin (2013:144) mengemukakan bahwa "Derajat *Relationship* adalah jumlah entitas yang berpartisipasi dalam satu *Relationship*".

Derajat *relationship* yang sering dipakai didalam ERD Ladjamudin (2013:145) adalah sebagai berikut:

1. *Unary Relationship*

   Unary relationship adalah model yang terjadi diantara entity yang berasal dari *entity* set yang sama. Sering juga disebut sebagai *recursive relationship* atau *reflective relationship*.

2. *Binary Relationship*

   Adalah model *relationship* antara *instance-instance* dari suatu tipe entitas (dua *entity* yang berasal dari *entity* yang sama). *Relationship* ini paling umum digunakan dalam pembuatan model data.

3. *Ternary Relationship*

   Ternary Relationship merupakan *relationship* antara *instance-instance* dari tiga tipe entitas sepihak.

2.2.3. *Logical Record Structure* (LRS)

LRS merupakan hasil dari dari transformasi dalam tahapan kardinalitas dari ERD ke LRS dan menghasilkan atribut-atribut yang saling berelasi.

Menurut Ladjamudin (2013:159) *"Logical Record Structure* (LRS) merupakan hasil transformasi ERD ke LRS yang melalui proses kardinalitas dan menghasilkan atribut-atribut yang saling berelasi*"*.

Aturan pokok dalam melakukan transformasi E-R Diagram ke logical record structure sangat dipengaruhi oleh elemen yang menjadi titik perhatian utama pada langkah transformasi dengan proses kardinalitas, yang terdiri dari tiga kardinalitas yaitu sebagai berikut Ladjamudin (2013:159):

1. *One to One*

   Yaitu proses kardinalitas yang panahnya lebih diarahkan di entity dengan jumlah atribut yang lebih sedikit.

2. *One to Many*

   Relasi harus diagbungkan dengan entity pada pihak *many*, dan tidak perlu melihat banyak sedikitnya pada entity tersebut.

3. *Many to Many*

   Yaitu proses kardinalitas pada *relationship* berubah status menjadi file konektor, sehingga baik *entity* maupun relasi akan menjadi struktur *record* sendiri.

2.2.4. Struktur Kode

Pengkodean digunakan dalam perancangan suatu program dimana kode kunci yang dibuat tersusun rapi berdasarkan aturan-aturan yang dirancang.

Menurut Jogiyanto (2009:384), "struktur kode adalah suatu bentuk struktur yang berfungsi untuk mengklasifikasikan data, memasukkan data ke dalam komputer dan untuk mengambil bermacam-macam informasi yang berhubungan dengannya".

Beberapa kemungkinan susunan digit (angka), huruf dan karakter-karakter khusus dapat dirancang ke dalam bentuk kode. dalam merancang suatu kode harus diperhatikan beberapa hal (Jogiyanto, 2009:384) yaitu:

1. Mudah diingat

   Supaya kode mudah diingat maka dapat dilakukan dengan cara menghubungkan kode tersebut dengan objek yang diwakili dengan kodenya.

2. Unik

   Kode harus unik untuk masing-masing item yang diwakilinya.

3. Fleksibel

   Kode harus fleksibel sehingga memungkinkan perubahan-perubahan atau penambahan *item* baru dapat tetap diwakili oleh kodenya masing-masing.

4. Efisien

   Kode harus sesingkat mungkin, selain mudah diingat juga akan efisien apabila direkam dan disimpan di luar komputer.

5. Konsisten

   Kode harus konsisten dengan kode yang telah digunakan.

6. Harus distandarisasi

   Kode harus distandarisasi untuk seluruh tingkatan dan departemen dalam organisasi.

7. Hindari penggunaan spasi

   Spasi dalam kode sebaiknya dihindarkan karena dapat menyebabkan kesalahan dalam menggunakannya.

8. Hindari karakter yang mirip

   Karakter-karakter yang hampir mirip bentuk dan bunyi pengucapannya sebaiknya tidak digunakan dalam kode.

9. Panjang kode harus sama

   Masing-masing kode yang sejenis harus memiliki panjang kode yang sama.

Ada beberapa macam tipe dari kode yang dapat digunakan, menurut Jogiyanto (2009:386) sebagai berikut:

1. Kode Mnemonik (*Mnemonic Code*)

   Kode mnemonik dibuat dengan dasar singkatan atau mengambil sebagian karakter dari *item* yang akan diwakili dengan kode ini.

2. Kode Urut (*Sequential Code*)

   Kode yang nilainya urut antara satu kode dengan kode berikutnya.

3. Kode Blok (*Block Code*)

   Kode ini mengklasifikasikan *item* ke dalam kelompok blok tertentu yang mencerminkan satu klasifikasi tertentu atas dasar pemakaian maksimum yang diharapkan.

4. Kode *Group* (*Group Code*)

   Kode yang dibentuk berdasarkan *field-field* dan tiap-tiap *field* kode mempunyai arti.

5. Kode Desimal (*Decimal Code*)

   Kode ini mengklasifikasikan dalam bentuk sepuluh unit angka desimal dimulai dari angka 0 sampai dengan angka 9 atau dari 00 sampai angka 99 tergantung dari banyaknya kelompok.

2.2.5. *Hierarchy Input Process Output* (HIPO)

Hirarchy Input Process Output (HIPO) digunakan untuk mendokumentasikan rancangan sistem dan mengkomunikasikan spesifikasi sistem kepada para *programmer*.

Menurut Amansyah (2008:284), "bagan HIPO dibuat oleh IBM sebagai alat untuk mendokumentasikan program, secara jelas memperagakan apa yang dikerjakan suatu program, data apa yang digunakan, dan keluaran yang dihasilkan". Sedangkan menurut Jogiyanto (2009:787) "HIPO (*Hierarchy plus Input-Process-Output*) merupakan metodologi yang dikembangkan dan didukung oleh IBM".

Berdasarkan pengertian para ahli di atas, dapat disimpulkan bahwa HIPO merupakan metodologi yang dikembangkan dan didukung oleh IBM yang sebenarnya adalah alat dokumentasi program dan sekarang banyak digunakan sebagai alat *design* dan teknik dokumentasi dalam pengembangan sistem.