

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

Sistem pada dasarnya banyak memberikan manfaat dalam mengambil keputusan, serta memahami lingkungan satu sama lain dan bersama-sama berfungsi untuk mencapai tujuan tertentu. Terdapat dua kelompok dalam mendefinisikan sistem yaitu menekankan pada prosedur dan menekankan pada elemen. Yang menekankan pada prosedur adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Sedangkan yang menekankan pada elemen sistem adalah kumpulan dari elemen-elemen yang berinteraktif mencapai suatu tujuan tertentu.

2.1.1 Pengertian Sistem

Menurut Hartono (2013:9) “Sistem adalah suatu himpunan dari berbagai bagian atau elemen, yang saling berhubungan secara terorganisasi berdasarkan fungsi-fungsinya, menjadi suatu kesatuan.”

2.1.2 Karakteristik Sistem

Sebuah sistem memiliki beberapa karakteristik sebagai berikut:

1. Komponen (*components*). Bagian-bagian atau element-elemen yang dapat berupa benda atau manusia, berbentuk nyata atau abstrak, dan disebut subsistem.

2. Penghubung antarbagian (*interface*). Sesuatu yang bertugas menjembatani satu bagian dengan bagian lain, dan memungkinkan terjadinya interaksi/komunikasi antarbagian.
3. Batas (*boundary*). Sesuatu yang membedakan antara satu sistem dengan sistem atau sistem-sistem lain.
4. Lingkungan (*environment*). Segala sesuatu yang berada di luar sistem dan dapat bersifat menguntungkan atau merugikan sistem yang bersangkutan.
5. Masukan (*input*). Sesuatu yang merupakan bahan untuk diolah atau diproses oleh sistem.
6. Mekanisme pengolahan (*processing*). Perangkat dan prosedur untuk mengubah masukan menjadi keluaran dan menampilkannya.
7. Keluaran (*output*). Berbagai macam bentuk hasil atau produk yang dikeluarkan dari pengolahan.
8. Tujuan (*goal/objective*). Sesuatu atau keadaan yang ingin dicapai oleh sistem, baik dalam jangka pendek maupun jangka panjang.
9. Sensor dan kendali (*sensor & control*). Sesuatu yang bertugas memantau dan menginformasikan perubahan-perubahan di dalam lingkungan dan dalam diri sistem kepada sistem.
10. Umpan-balik (*feedback*). Informasi tentang perubahan-perubahan lingkungan dan perubahan-perubahan (penyimpangan) dalam diri sistem.

2.1.3 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang di antaranya sebagai berikut:

1. Sistem Abstrak (*abstrak system*) dan Sistem Fisik (*physic system*)

Sistem abstrak adalah sistem yang berupa pemikiran atau ide yang tidak tampak secara fisik, misalnya sistem teknologi yaitu sistem yang berupa pemikiran hubungan antar manusia dengan tuhan, sedangkan sistem fisik merupakan sistem yang ada secara fisik, misalnya sistem penjualan, sistem produksi dan sebagainya.

2. Sistem Alamiah (*natural system*) dan Sistem Buatan Manusia (*human made system*)

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem perputaran bumi, terjadi siang dan malam, pergantian musim. Sedangkan sistem buatan manusia adalah sistem yang dirancang oleh manusia yang melibatkan interaksi manusia dengan mesin.

3. Sistem Tertentu (*deterministic system*) dan Sistem Tak Tentu (*probabilistic system*)

Sistem tertentu adalah sistem yang beroperasi dengan tingkah laku yang dapat diprediksi atau diramalkan secara tepat, misalnya sistem computer contoh dari sistem yang tingkah lakunya dapat dipastikan berdasarkan program-program komputer yang dijalankan. Sedangkan sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi atau diramalkan secara tepat, misalnya ramalan cuaca.

4. Sistem Tertutup (*closed system*) dan Sistem Terbuka (*open system*)

Sistem tertutup adalah sistem yang tidak terpengaruh dengan lingkungan luarnya, sedangkan sistem terbuka adalah sistem yang terpengaruh dengan lingkungan luarnya.

2.1.4 *System Development Life Cycle (SDLC)*

1. Pengertian SDLC

Menurut Rossa dan Shalahudin (2014:26) mengemukakan bahwa SDLC atau *Software Development Life Cycle* atau sering disebut juga System Development Life Cycle adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan best practice atau cara-cara yang sudah teruji baik).

2. Tahapan SDLC

Menurut Rossa dan Shalahudin (2014:26) Tahapan-tahapan yang ada pada SDLC secara global adalah sebagai berikut:

a. Inisiasi (*initiation*)

Tahap ini biasanya ditandai dengan pembuatan proposal proyek perangkat lunak.

b. Pengembangan konsep sistem (*system concept development*)

Mendefinisikan lingkup konsep termasuk dokumen lingkup sistem, analisis manfaat biaya, manajemen rencana, dan pembelajaran kemudahan sistem.

c. Perencanaan (*planning*)

Mengembangkan rencana manajemen proyek dan dokumen perencanaan lainnya. Menyediakan dasar untuk mendapatkan sumber daya (*resources*) yang dibutuhkan untuk memperoleh solusi.

d. Analisis kebutuhan (*requirements analysis*)

Menganalisis kebutuhan pemakai sistem perangkat lunak (*user*) dan mengembangkan kebutuhan *user*. Membuat dokumen kebutuhan fungsional.

e. Desain (*design*)

Mentransformasikan kebutuhan detail menjadi kebutuhan yang sudah lengkap, dokumen desain sistem fokus pada bagaimana dapat memenuhi fungsi-fungsi yang dibutuhkan.

f. Pengembangan (*development*)

Mengonversi desain ke sistem informasi yang lengkap termasuk bagaimana memperoleh dan melakukan instalasi lingkungan sistem yang dibutuhkan; membuat basis data dan mempersiapkan berkas atau file mengujian, pengkodean, pengompilasian, memperbaiki dan membersihkan program; peninjauan pengujian.

g. Integrasi dan pengujian (*integration and test*)

Mendemonstrasikan sistem perangkat lunak bahwa telah memenuhi kebutuhan yang dispesifikasikan pada dokumen kebutuhan fungsional. Dengan diarahkan oleh staf penjamin kualitas (*quality assurance*) dan *user*. Menghasilkan laporan analisis pengujian.

h. Implementasi (*implementation*)

Termasuk pada persiapan implementasi, implementasi perangkat lunak pada lingkungan produksi (lingkungan pada *user*) dan menjalankan resolusi dari permasalahan yang teridentifikasi dari fase integrasi dan pengujian.

i. Operasi dan pemeliharaan (*operations and maintenance*)

Mendeskripsikan pekerjaan untuk mengoperasikan dan memelihara sistem informasi pada lingkungan produksi (lingkungan pada *user*), termasuk implementasi akhir dan masuk pada proses peninjauan.

j. Disposisi (*disposition*)

Mendeskripsikan aktifitas ahir dari pengembangan sistem dan membangun data yang sebenarnya sesuai dengan aktifitas *user*.

3. Metode SDLC

a. Model *waterfall*

Menurut Rossa dan Sahalahudin (2014:28) model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut mulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*).

b. Model Prototipe

Menurut Rossa dan Shalahudin (2014:31) model prototipe dimulai dari mengumpulkan kebutuhan pelanggan terhadap perangkat lunak yang akan

dibuat. Lalu dibuatlah program prototipe agar pelanggan lebih terbayang dengan apa yang sebenarnya diinginkan. Program ini biasanya merupakan program yang belum jadi.

c. Model *Rapid Application Development* (RAD)

Menurut Rossa dan Shalahudin (2014:34) “*Rapid Application Development* (RAD) adalah model proses pengembangan perangkat lunak yang bersifat inkremental terutama untuk waktu pengerjaan yang pendek.”

d. Model Iteratif

Menurut Rossa dan Shalahudin (2014:38) menyatakan bahwa “model iteratif (*iterative model*) mengkombinasikan proses-proses pada model air terjun dan prototipe pada model iteratif.”

e. Model Spiral

Menurut Rossa dan Shalahudin (2014:39) mengemukakan bahwa “model spiral (*spiral model*) memasangkan iteratif pada model prototipe dengan kontrol dan aspek sistematis yang diambil dari model air terjun.”

2.1.5 Sistem Informasi Manajemen

1. Pengertian Informasi

Menurut Gordon B. Davis dalam Hartono (2013:15) memberikan definisi informasi sebagai berikut: “Informasi adalah data yang telah diolah menjadi suatu bentuk yang berguna bagi penerimanya dan memiliki nilai bagi pengambilan keputusan saat ini atau di masa yang akan datang”.

2. Pengertian Sistem Informasi

Menurut Hartono (2013:16) menyatakan bahwa “sistem informasi adalah seperangkat komponen yang saling berhubungan, yang bekerja untuk mengumpulkan dan menyimpan data serta mengolahnya menjadi informasi untuk digunakan.”

3. Pengertian Sistem Informasi Manajemen

Menurut Hartono (2013:20) menyimpulkan bahwa “sistem informasi manajemen adalah sebuah sistem, rangkaian terorganisasi dari sejumlah bagian/komponen yang secara bersama-sama berfungsi atau bergerak menghasilkan informasi untuk digunakan dalam manajemen perusahaan.”

2.1.6 Sistem Informasi Akuntansi

Menurut Wijayanto dalam Mardi (2014:4) mengemukakan bahwa “Sistem Informasi Akuntansi adalah susunan berbagai dokumen, alat komunikasi, tenaga pelaksana, dan berbagai laporan yang di desain untuk mentransformasikan data keuangan menjadi informasi keuangan.”

2.1.7 Jurnal

Menurut Priyanti (2013:28) mendefinisikan bahwa “Jurnal adalah Catatan secara sistematis dan kronologis dari transaksi-transaksi finansial dengan menyebut perkiraan yang akan didebit dan dikredit, jumlah dan keterangan ringkas. Jurnal merupakan catatan transaksi finansial yang pertama karena itu disebut juga sebagai catatan yang asli (*book of original entry*)”.

Jurnal penjualan kredit adalah sebagai berikut:

1. Jurnal pada saat penjualan barang secara kredit

Piutang xxx

Penjualan xxx

Contoh :

Piutang Rp 1.000.000

Penjualan Rp 1.000.000

2. Jurnal pada saat terjadi retur penjualan secara kredit

Retur Penjualan xxx

Piutang Dagang xxx

Contoh :

Retur Penjualan Rp 250.000

Piutang Dagang Rp 250.000

2.1.8 Pengertian Penjualan Kredit

Adalah penjualan yang dilakukan secara kredit atau tidak langsung dibayarkan secara tunai, biasanya dalam penjualan ini terdapat syarat-syarat yang harus disepakati.

2.2 Peralatan Pendukung (*Tool System*)

Dalam penulisan tugas akhir ini serta untuk merancang suatu sistem diperlukan peralatan pendukung (*tool system*). Adapun peralatan pendukung yang penulis gunakan adalah:

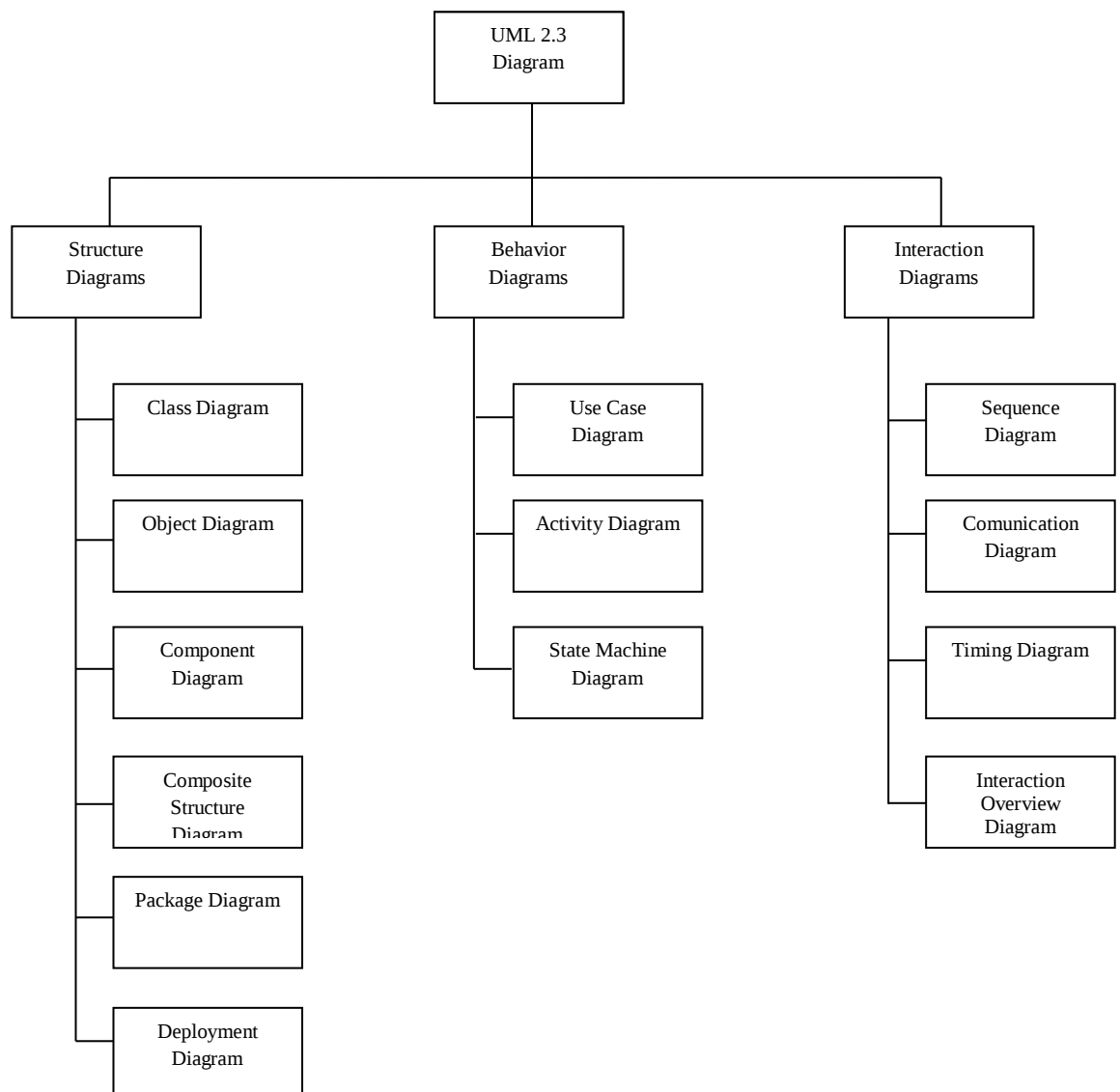
2.2.1 *Unified Modeling Language* (UML)

1. Pengertian UML

Menurut Rossa dan Shalahudin (2014:133) “UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

2. Diagram UML

Menurut Rossa dan Shalahudin (2014:140) “UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar di bawah.”



Gambar II. 1

Diagram UML (*Unified Modeling Language*)

a. *Use Case Diagram*

1. Pengertian *Use Case Diagram*

Menurut Rossa dan Shalahuddin (2013:155) “*Use case* diagram atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat”.

2. Simbol dan Fungsi *Use Case* Diagram**Tabel II. 1**Simbol *Use Case* Diagram

	USECASE Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.
	ACTOR Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
	ASSOCIATION Komunikasi antara <i>actor</i> dan <i>usecase</i> yang berpartisipasi pada <i>usecase</i> atau <i>usecase</i> memiliki interaksi dengan aktor.
--<<include>>-->	INCLUDE Relasi <i>usecase</i> dimana proses bersangkutan akan dilanjutkan keproses yang dituju.
--<<extend>>-->	EXTEND Relasi <i>usecase</i> tambahan sebuah <i>usecase</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>usecase</i> tambahan itu.

b. Activity Diagram

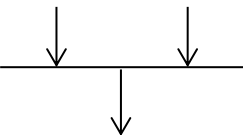
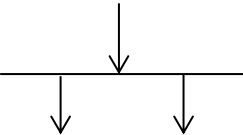
1. Pengertian Activity Diagram

Menurut Rossa dan Shalahuddin (2013:161) “Diagram aktivitas atau *Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”.

2. Simbol dan Fungsi Activity Diagram

Tabel II. 2

Simbol Activity Diagram

	INITIAL NODE Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
	ACTION Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	DECISION Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
	JOIN Digunakan untuk menunjukkan kegiatan yang digabungkan.
	FORK Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel.

	ACTIVITY FINAL Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
---	--

c. *Sequence Diagram*

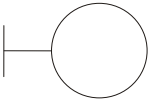
1. Pengertian *Sequence Diagram*

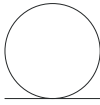
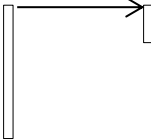
Menurut Rossa dan Shalahuddin (2013:165) “Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek”.

2. Simbol dan Fungsi *Sequence Diagram*

Tabel II. 3

Simbol *Sequence Diagram*

	ACTOR Orang, poses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari <i>actor</i> adalah gambar orang, tapi <i>actor</i> belum tentu merupakan orang.
	INTERFACE atau BOUNDARY OBJECT Sebuah obyek yang menjadi penghubung user dengan sistem.

	<i>CONTROL OBJECT</i> Mengkoordinasikan perilaku sistem dan dinamika dari suatu sistem, menangani tugas utama dan mengontrol alur kerja suatu sistem.
	<i>ENTITY OBJECT</i> Suatu Objek yang berisi informasi kegiatan yang terkait yang tetap dan disimpan kedalam suatu <i>database</i>.
	<i>OBJECT MESSAGE</i> Menggambarkan pesan/hubungan antar obyek, yang menunjukkan urutan kejadian yang terjadi.
	<i>MESSAGE to SELF</i> Menggambarkan pesan/hubungan objek itu sendiri, yang menunjukkan urutan kejadian yang terjadi.
	<i>LIFELINE</i> Garis titik-titik yang berhubungan dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.

	ACTIVATION Mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.
---	--

d. Deployment Diagram

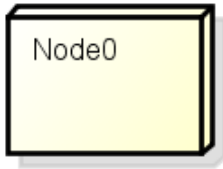
1. Pengertian *Deployment* Diagram

Menurut Rossa dan Shalahudin (2014:154) menyatakan bahwa “Diagram deployment atau deployment diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi.”

2. Simbol dan Fungsi *Deployment* Diagram

Tabel II. 4

Simbol *Deployment* Diagram

	NODE Sumber daya yang digunakan pada saat aplikasi dijalankan.
	ASSOCIATION Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>.

2.2.2 Entity Relationship Diagram (ERD)

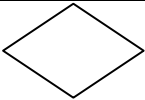
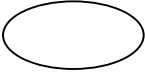
1. Pengertian Entity Relationship Diagram (ERD)

Rosa dan Shalahuddin (2013:50) dalam bukunya mengatakan “*Entity Relationship Diagram (ERD)* dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional”.

2. Simbol dan Fungsi ERD

Tabel II. 5

Simbol *Entity Relationship Diagram* (ERD)

	ENTITAS Suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
	RELASI Menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.
	ATRIBUT Berfungsi mendeskripsikan karakter entitas (atribut yang berfungsi sebagai <i>key</i> diberi garis bawah).
	LINE Relasi antara Entitas dan Atribut.

3. *Mapping Cardinality*

Kardinalitas pemetaan atau rasio kardinalitas menunjukkan jumlah entity yang dihubungkan ke satu entity lain dengan suatu *relationship sets*. menurut Kusrini (2007:24) kardinalitas pemetaan meliputi:

- a. Hubungan satu ke satu (one to one). yaitu satu entity dalam A dihubungkan dengan maksimum satu entity B.
- b. Hubungan satu ke banyak (one to many). Yaitu satu entity dalam A dihubungkan dengan sejumlah entity B. Entity dalam B dihubungkan dengan maksimum satu entity dalam A.
- c. Hubungan banyak ke satu (many to one). Yaitu sejumlah entity dalam A dihubungkan dengan maksimum satu entity B. Satu entity dalam B dapat dihubungkan dengan sejumlah entity dalam A.
- d. Hubungan banyak ke banyak (many to many). Sejumlah entity dalam A dihubungkan dengan sejumlah entity dalam B. Sejumlah entity dalam B dihubungkan dengan sejumlah entity dalam A.

4. Tahapan Pembuatan ERD

Menurut Kusrini (2007:25) tahapan pembuatan *entity relationship diagram* (ERD) adalah :

- a. Mengidentifikasi dan menetapkan seluruh himpunan entitas (entity) yang akan terlihat.
- b. Menentukan atribut-atribut key dari masing-masing himpunan entitas.
- c. Menentukan *primary key* (PK) dari setiap entitas yang ada.

- d. Mengidentifikasi dan menetapkan seluruh himpunan relasi di antara himpunan entitas-himpunan entitas yang ada berserta *foreign key*-nya.
- e. Menentukan derajat/kardinalitas relasi untuk setiap himpunan relasi.

2.2.3 Logical Record Structure (LRS)

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”. Perubahan yang terjadi yaitu mengikuti aturan-aturan sebagai berikut (Hasugian dan Shidiq, 2012:608):

1. Setiap entitas akan diubah kebentuk kotak.
2. 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).
3. Sebuah relasi dipisahkan dalam sebuah kotak tersendiri (menjadi entitas baru) jika tingkat hubungan M:M (*many to many*) dan memiliki *foreign key* sebagai *primary key* yang diambil dari kedua entitas yang sebelumnya saling berhubungan.

2.2.4 Neatbeans

Menurut enterprise (2015:1) Java merupakan bahasa pemrograman yang mampu bekerja dengan sebuah database. Java merupakan pemrograman yang sangat populer karena rentang aplikasi yang bisa dibuat menggunakan bahasa ini sangatlah luas, mulai dari komputer hingga smartphone. Bahasa pemrograman

java dikembangkan pertama kali oleh *sun microsystem* yang dimulai oleh james gosling dan dirilis pada tahun 1995. Saat ini *sun microsystem* telah diakuisi oleh *oracle corporation*. Java bersifat *write once, run anywhere* (program yang ditulis satu kali dan dapat berjalan pada banyak platform). Dengan demikian tidak mengherankan apabila aplikasi yang dibuat menggunakan java bisa ditemukan di lingkungan komputer dan smartphone tanpa perbedaan yang berarti.

Netbeans merupakan *Integrated Development Environment* (IDE) mengacu kepada kedua platform kerangka untuk aplikasi desktop java dan sebuah lingkungan pengembangan terpadu (IDE) untuk pengembangan dengan Java, Java Script, PHP, Python, Ruby, Groovy, C, C++, Scala, Clojure dan lain-lain.

Netbeans IDE ditulis dalam java dan berjalan diberbagai macam seperti Windows, Linux, Mac OS x and Solaris. Sebuah JDK diperlukan untuk pengembangan fungsionalitas java, tetapi tidak diperlukan untuk pembangunan bahasa pemrograman lain.

Platform Netbeans memungkinkan aplikasi untuk dikembangkan dari satu set modular komponen *software* yang disebut modul. Aplikasi berbasis platform Netbeans (termasuk IDE Netbeans) dapat diperpanjang oleh pengembang pihak ketiga.