

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Sistem adalah suatu kumpulan elemen-elemen, subsistem yang saling berinteraksi, mempengaruhi dan saling bergantung satu sama lain untuk mencapai tujuan yang sama.

2.1.1. Pengertian Sistem

Menurut Fatta (2007:3) menjelaskan bahwa “Sistem adalah kumpulan dari bagian-bagian yang bekerja sama untuk mencapai tujuan bersama atau sekumpulan objek-objek yang saling berinteraksi serta hubungan antar objek bisa dilihat sebagai satu kesatuan yang dirancang untuk mencapai suatu tujuan”. Menurut Kadir (2008:54) yaitu “Sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan”. Sedangkan menurut Mulyanto (2009: 1) juga menyimpulkan bahwa “ Sistem adalah kumpulan suatu tujuan tertentu sebagai satu kesatuan”.

Sujarweni (2015:1) juga menjelaskan bahwa sistem dilihat dari elemen-elemennya sistem adalah sekumpulan elemen yang saling berkaitan dan bekerja sama dalam melakukan kegiatan untuk mencapai suatu tujuan, sistem dilihat dari masukan dan keluarannya adalah suatu rangkaian yang berfungsi menerima input (masukan), mengelola input dan menghasilkan output (keluaran).

Berdasarkan definisi diatas, sistem adalah suatu jaringan kerja dari kumpulan atau himpunan dari unsur, komponen-komponen, prosedur-prosedur, atau variabel yang terorganisir, yang saling berinteraksi, saling ketergantungan

antara satu dan lainnya dimana setiap sistem dibuat agar bersama-sama dapat menangani dan menyelesaikan suatu saran tertentu secara rutin.

2.1.2. Karakteristik Sistem

Menurut Kusriani dan Andri Koniyo (2007:6) menjelaskan bahwa “Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu”. Adapun karakteristik yang dimaksud adalah sebagai berikut :

1. Komponen Sistem (*Component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi dan bekerjasama membentuk satu kesatuan.

2. Batasan Sistem (*Boundary*)

Merupakan daerah yang membatasi satu sistem dan sistem lainnya atau dengan lingkungan luar.

3. Lingkungan Luar Sistem (*Environment*)

Segala sesuatu yang berada diluar batas sistem yang mempengaruhi operasi sistem baik itu yang bersifat merugikan ataupun menguntungkan.

4. Penghubung Sistem (*Interface*)

Media penghubung antara suatu subsistem yang memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lain.

5. Masukan Sistem (*Input*)

Energi yang dimasukkan kedalam sistem, yang dapat berupa masukan perawatan (*Maintenance input*) dan masukan sinyal (*Signal input*).

6. Keluaran Sistem (*Output*)

Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dari sisa pembuangan.

7. Pengolahan Sistem (*Process*)

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

8. Sasaran Sistem (*Object*)

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*object*), jika suatu sistem tidak mempunyai tujuan yang jelas, maka semua operasi sistem tidak ada gunanya.

2.1.3. Klasifikasi Sistem

Menurut Sutabri (2007:14) menjelaskan bahwa “Sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen lain karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi didalam sistem tersebut”.

Sistem dapat diklasifikasikan dari beberapa sudut pandang, yaitu sebagai berikut:

1. Sistem abstrak (*Abstrac system*) dan sistem fisik (*Phisical system*)

Sistem abstrak (*Abstrac system*) adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya sistem teologia yaitu pemikiran tentang hubungan antara manusia dengan Tuhan. Sedangkan sistem fisik (*Phisical system*) adalah sistem yang ada secara fisik, misalnya seperti komputer, sistem produksi, sistem penjualan, sistem administrasi personalia.

2. Sistem alamiah (*Natural system*) dan sistem buatan manusia (*Human made system*)

Sistem alamiah (*Natural system*) adalah sistem yang terjadi melalui proses alam tidak dibuat oleh manusia, misalnya sistem perputaran alam.

Sedangkan sistem buatan manusia (*Human made system*) adalah sistem yang melibatkan hubungan manusia dengan mesin (*human machine system*), misalnya komputer yang berinteraksi dengan manusia.

3. Sistem deterministik (*Deterministic system*) dan sistem probabilitistik (*Probabilistic system*)

Sistem deterministik (*Deterministic system*) adalah sistem yang beroperasi dengan tingkah laku yang dapat diprediksikan, misalnya program-program komputer yang dijalankan. Sedangkan sistem probabilitistik (*Probabilistic system*) adalah sistem yang kondisi masa depannya tidak dapat di prediksi, karena mengandung unsur probabilitas.

4. Sistem Terbuka (*Open system*) dan sistem tertutup (*Closed system*)

Sistem Terbuka (*Open system*) adalah sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya, misalnya menerima masukan dan menghasilkan keluaran untuk sistem lainnya. Sedangkan sistem tertutup (*Closed system*) adalah sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luarnya, misalnya sistem ini bekerja secara otomatis tanpa campur tangan dari pihak luar.

2.1.4. Pengertian Informasi

Menurut Mulyanto (2009:12) menyimpulkan bahwa “Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya”.sedangkan menurut Fatta (2007:9) “Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang”.

Berdasarkan definisi diatas, informasi adalah kumpulan data-data yang diproses menjadi bentuk yang memiliki arti bagi penerimanya yang menggambarkan suatu kejadian-kejadian nyata. 3 (tiga) hal penting yang menjadi dasar dalam menentukan kualitas dari suatu informasi, yaitu:

1. Akurat

Informasi harus bebas dari kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerimaan informasi kemungkinan banyak terjadi gangguan (*noise*) yang dapat merubah atau merusak informasi tersebut.

2. Tepat Pada Waktu

Informasi yang datang pada penerimaan tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan didalam pengambilan keputusan.

3. Relevan

Informasi tersebut mempunyai manfaat untuk pemakaiannya. Relevan informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda.

2.1.5. Pengertian Sistem Informasi

Menurut Kusriani dan Andri Koniyo (2007:8) menyimpulkan bahwa “Sistem Informasi adalah sistem yang terdiri dari atas rangkaian subsistem informasi terhadap pengolahan data untuk menghasilkan informasi yang berguna dalam pengambilan keputusan”. Sedangkan menurut Fatta (2007:14) juga menjelaskan bahwa “Sistem Informasi merupakan sistem dengan komponen-komponen yang bekerja untuk mengolah data menjadi informasi”.

Berdasarkan definisi di atas, sistem informasi adalah suatu sistem yang menyediakan informasi untuk manajemen dalam mengambil keputusan dan juga untuk menjalankan operasional perusahaan. Ada beberapa perbedaan pendapat tentang komponen-komponen secara umum sistem informasi, yang terdiri dari:

1. Blok Masukan (*Input Block*)

Merupakan input yang mewakili data yang masuk ke dalam sistem informasi, yang dapat berupa dokumen-dokumen dasar.

2. Blok Model (*Model Block*)

Blok ini terdiri kombinasi prosedur, logika model matematika, yang akan memanipulasi data input dan data yang tersedia dalam basis data untuk memperoleh keluaran yang diinginkan.

3. Blok Keluaran (*Output Block*)

Produk dalam sistem informasi adalah keluaran yang berupa informasi yang berkualitas dan dokumentasi yang berguna

4. Blok Teknologi (*Technology Block*)

Teknologi merupakan kotak alat (*tool-box*) dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan sekaligus mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan.

5. Blok Basis Data (*Database Block*)

Basis data merupakan kumpulan data yang saling berhubungan dan tersimpan diperangkat keras dan digunakan diperangkat lunak untuk memanipulasinya. Karena itu, data perlu disimpan dan diorganisasikan dalam basis data untuk menghasilkan informasi yang berkualitas.

6. Blok Kendali (*Control Block*)

Blok yang digunakan untuk mengatasi hal-hal yang dapat merubah sistem, baik yang berupa sabotase, tidak efisien, kecurangan, kegagalan dan sebagainya.

2.1.6. Pengertian Sistem Informasi Akuntansi

Menurut Mujilan (2012:3) Sistem informasi akuntansi adalah kumpulan sumberdaya, seperti manusia dan peralatan, yang diatur untuk mengubah data menjadi informasi. Informasi ini dikomunikasikan kepada beragam pengambil keputusan. SIA mewujudkan perubahan ini secara manual atau terkomputerisasi.

SIA juga merupakan sistem yang paling penting di organisasi dan merubah cara menangkap, memproses, menyimpan, dan mendistribusikan informasi (Xu, 2009). Saat ini, digital dan informasi online semakin digunakan dalam sistem informasi akuntansi. Organisasi perlu menempatkan sistem di lini depan, dan mempertimbangkan baik segi sistem ataupun manusia sebagai faktor yang terkait ketika mengatur sistem informasi akuntansi.

Sistem Informasi Akuntansi pada umumnya meliputi beberapa siklus pemrosesan transaksi:

1. Siklus pendapatan. Berkaitan dengan pendistribusian barang dan jasa ke entitas lain dan pengumpulan pembayaranpembayaran yang berkaitan.
2. Siklus pengeluaran. Berkaitan dengan perolehan barang jasa dari entitas lain dan pelunasan kewajiban yang berkaitan.
3. Siklus produksi. Berkaitan dengan pengubahan sumberdaya menjadi barang dan jasa.

4. Siklus keuangan. Kejadian-kejadian yang berkaitan dengan perolehan dan manajemen dana-dana modal, termasuk kas.

Menurut Sutabri (2012:83) juga menjelaskan bahwa “Sistem informasi akuntansi adalah sebuah sistem informasi yang menangani segala sesuatu yang berkenaan dengan akuntansi”.

Berdasarkan definisi di atas, sistem informasi akuntansi adalah sistem yang memproses data dan transaksi guna menghasilkan informasi yang berhubungan dengan akuntansi.

2.1.7. Pengertian Pembayaran

Menurut Yunirman (2009: 36) Pembayaran mengandung beberapa arti, yakni dalam arti sempit pembayaran adalah pelunasan hutang oleh debitur kepada kreditur bisa dilakukan dalam bentuk uang atau barang. Dalam pengertian secara yuridis pembayaran tersebut bisa saja dalam bentuk jasa, misalnya jasa tukang cukur, jasa guru music, dan lain-lain.

2.1.8. Pengertian Komite Sekolah

Menurut Ade Irawan (2004:42) Komite Sekolah merupakan badan mandiri yang mewadahi peranserta masyarakat dalam rangka meningkatkan mutu, pemerataan dan efisiensi pengelolaan pendidikan di satuan pendidikan, baik pada pendidikan pra sekolah, jalur pendidikan sekolah maupun jalur pendidikan non sekolah.

2.2 Peralatan Pendukung (*Tool System*)

Didalam merancang sebuah sistem, diperlukan suatu peralatan yang dapat mendukung terciptanya sebuah rancangan. Peralatan pendukung (*Tool System*) merupakan alat yang digunakan untuk menggambarkan bentuk logika model dari

suatu sistem dengan menggunakan simbol, lambang, dan diagram yang menunjukkan secara tepat arti dan fungsinya.

Adapun peralatan pendukung yang digunakan untuk merancang suatu sistem baru yaitu berupa OOP, UML, MySQL, Net Beans IDE 8.0.1, ERD dan LRS yang membantu penulis dalam pembuatan tugas akhir ini.

2.2.1. Object Oriented Programming (OOP)

Rosa & Shalahuddin (2015:100) Suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya.

Keuntungan menggunakan metode berorientasi objek adalah sebagai berikut :

1. Meningkatkan produktivitas

Karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lain yang melibatkan objek tersebut (reusable).

2. Kecepatan pengembangan

Karena sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengkodean.

3. Kemudahan pemeliharaan

Karena dengan model objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dan pola-pola yang mungkin sering berubah-ubah.

4. Adanya konsistensi

Karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan maupun pengkodean.

5. Meningkatkan kualitas perangkat lunak

Karena pendekatan pengembangan lebih dekat dengan dunia nyata dan adanya konsistensi pada saat pengembangannya, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.

2.2.2. Unified Modeling Language (UML)

Shalahuddin dan Rosa (2015:133) UML adalah Standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemograman berorientasi objek.

Pada jenis pemrograman berbasis objek(*object oriented*), misalkan dengan bahasa java, digunakan pemodelan UML (*Unified Modeling Language*) adalah standarisasi internasional untuk notasi dalam bentuk grafik, yang menjelaskan tentang analisis dan desain perangkat lunak yang dikembangkan dengan pemograman berorentasi objek.

2.2.2.1. Use Case Diagram

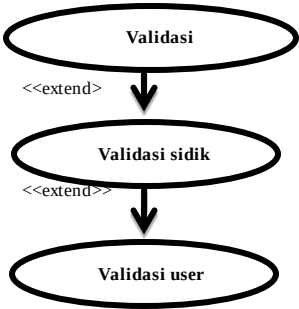
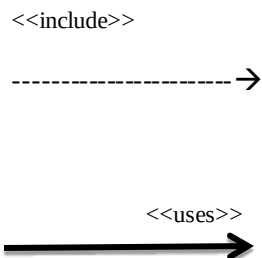
Shalahuddin dan Rosa (2015:155) Use case atau diagram use case merupakan pemodelan untuk melakukan (behavior) sisteminformasi yang akan dibuat. use case mendeskripsikan sebuah interaksi antara satu atau lebih actor digunakan untuk mengetahui fungsi apa saja yang ada pada sebuah sistem informasi apa saja yang berhak menggunakanfungsi-fungsi itu.

Syarat penamaan pada use case adalah nama di definisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada use case yaitu mendefenisikan apa yang disebut aktor dan use case.

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. Use case merupakan fungsionalitas yang di sediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Tabel 2.1
Simbol Use Case Diagram

Simbol	Deskripsi
<i>Use case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case.
aktor/actor  nama aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
Asosiasi/association 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau use case memiliki interaksi dengan aktor.

Simbol	Deskripsi
Ekstensi/ extend 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal  arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya
Generalisasi/generalization 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan/include/uses 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini. Ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i>: • Include berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan. • Include berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan.

Sumber : Shalahuddin dan rosa (2015:156)

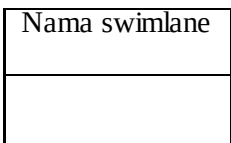
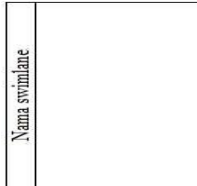
2.2.2.2. Activity Diagram

Menurut Rosa dan Shalahudin (2015:161) Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal sebagai berikut:

- Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- Urutan atau pengelompokan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
- Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
- Rancangan menu yang ditampilkan pada perangkat lunak.

Tabel 2.2
Simbol Activity Diagram

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Swimlane  Atau 	Memisahkan sebuah organisme bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber: Shalahuddin dan rosa (2015:161)

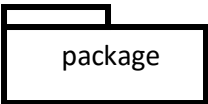
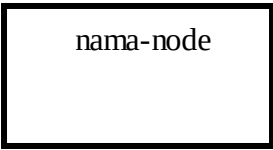
2.2.2.3. Deployment Diagram

Menurut Rosa dan Shalahudin (2015:154) Diagram *deployment* atau *deployment* diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat digunakan untuk memodelkan hal-hal berikut:

- Sistem tambahan (*embedded system*) yang menggambarkan rancangan *device*, *node*, dan *hardware*.
- Sistem *client/server*

Berikut adalah simbol-simbol yang ada pada diagram deployment:

Tabel 2.3
Simbol Deployment Diagram

Simbol	Deskripsi
Package 	<i>Package</i> merupakan sebuah bungkusan dari satu atau lebih <i>node</i> .
Node 	Biasanya mengacu pada perangkat keras(<i>hardware</i>), perangkat lunak yang dibuat sendiri(<i>software</i>), jika didalam node di sertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikuti sertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen.
Kebergantungan/ <i>dependency</i> 	Kebergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai.
Link 	Relasi antar <i>node</i> .

Sumber :Shalahuddin dan rosa (2015:161)

2.2.2.4. Sequence Diagram

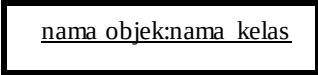
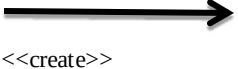
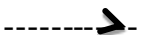
Menurut Rosa dan Shalahudin (2015:165) Diagram sequen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

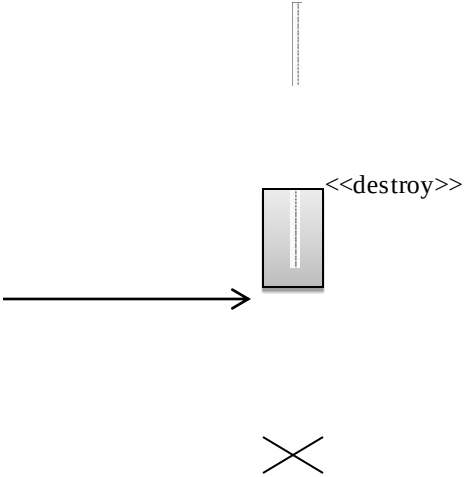
Berikut adalah simbol-simbol yang ada pada diagram sekuen:

Tabel 2.4
Simbol Sequence Diagram

Simbol	Deskripsi
 Aktor nama aktor atau nama Tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.

Simbol	Deskripsi
Garis hidup/ lifeline 	Menyatakan kehidupan suatu objek
objek 	Menyatakan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.
Pesan tipe create 	Menyatakan suatu objek membuat objek lain, arah panah mengarah pada objek yang dibuat.
Pesan tipe call 1:nama_metode() 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, arah panah mengarahkan pada objek yang memiliki operasi /metode maka operasi/metode yang dipanggil sesuai dengan kelas objek yang berinteraksi.
Pesan tipe send 1:masukan  Pesan tipe return 1:keluaran 	Menyatakan bahwa suatu objek yang mengirimkan data masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim. Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.

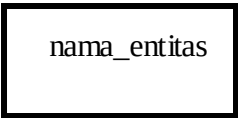
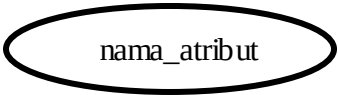
Sumber : Shalahuddin dan rosa (2015:165)

Simbol	Deskripsi
Pesan tipe destroy 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mngarah pada objek yang di akhiri,sebaiknya jika ada create maka ada destroy

2.2.3. Entity Relationship Diagram (ERD)

Menurut Rosa dan Shalahudin (2015:50) Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram* (ERD). ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data rasional. 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 beberapa notasi lain. Namun banyak digunakan adalah notasi dari Chen. Berikut adalah simbol-simbol yang digunakan pada ERD dengan notasi Chen:

Tabel 2.5
Simbol Entity Relationship Diagram

Simbol	Deskripsi
Entitas/entity 	Entitas merupakan data 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.
Atribut 	<i>Field</i> atau kolom data yang butuh disimoan dalam suatu entitas
Atribut kunci primer 	<i>field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> 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).
Atribut multivalai/multivalue 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja
Asosiasi/association 	Penghubung antara relasi dan entitas dimana kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas lain disebut dengan kardinalitas.

Sumber : Shalahuddin dan rosa (2015:50)

2.2.4. Logika Record Structure (LRS)

Logical Record Structure (LRS) sering juga dipakai oleh para pengembang aplikasi web untuk membuat sebuah aplikasi web yang dinamis. perbedaan ERD dan LRS terdapat pada *field-nya*, setiap nama field di ERD ditulis dari luar kotak (entity), sedangkan setiap nama field di LRS ditulis di dalam kotak dan memiliki sebuah nama yang unik.

Menurut Frieyadie (2007:13) “LRS merupakan hasil dari pemodelan *Entity Relational Ship* (ER) beserta atributnya sehingga bisa terlihat hubungan-hubungan antar entitas”. Dalam pembuatan LRS terdapat 3 hal yang dapat mempengaruhi yaitu:

1. Jika tingkat hubungan (*cardinality*) satu pada satu (*one-to-one*), maka di gabungkan dengan entitas yang lebih kuat (*strong entity*), atau digabungkan dengan entitas yang memiliki atribut yang lebih sedikit.
2. Jika tingkat hubungan (*cardinality*) satu pada banyak (*one-to-many*), maka hubungan relasi atau digabungkan dengan entitas yang tingkat hubungannya banyak.
3. Jika tingkat hubungan (*cardinality*) banyak pada banyak (*many-to-many*), maka hubungan relasi tidak akan digabungkan dengan entitas manapun, melainkan menjadi sebuah LRS.

2.2.5. MySQL

Wahana Komputer (2015:7) MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis. Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Kehandalan suatu sistem basisdata (DBMS) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, MySQL mendukung operasi basisdata transaksional maupun operasi basisdata non-transaksional. Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basisdata kompetitor lainnya. Namun demikian pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi blogging berbasis web (wordpress), CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basisdata transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional.

2.2.6. XAMPP

Menurut Pratama (2014:440) XAMPP adalah aplikasi *web server* bersifat instan (siap gaji) yang dapat digunakan baik disistem operasi linux maupun di sistem operasi Windows.

Sedangkan menurut Wahana Komputer (2015:55) XAMMP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP, Perl. XAMMP adalah *tool* yang menyediakan paket perangkat lunak dalam satu buah paket. Pada paket XAMMP telah terdapat Apache (web server), MySQL (database), PHP (server side scripting), Perl, FTP server, Php MyAdmin, dan berbagai pustaka bantu lainnya.

Dari defenisi-defenisi tersebut diatas dapat disimpulkan bahwa XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program.

2.2.7. Java

Wahana Komputer (2015:2) Java merupakan bahasa pemrograman tingkat tinggi yang memiliki karakteristik *simple*, *object-oriented*, *distributes*, *interpreted*, dan memiliki performa yang tinggi. Bahasa pemrograman Java merupakan *compiler* sekaligus interpreter, di mana sebagai *compiler* program yang telah dibuat akan diubah menjadi Java bytecodes. Java sebagai interpreter Java bytecodes tersebut dijalankan pada komputer.

2.2.8. NetBeans

Wahana Komputer (2015:20) NetBeans adalah salah satu aplikasi IDE yang digunakan oleh developer software komputer untuk menulis, meng-*compile*, mencari kesalahan, dan untuk menyebarkan program.NetBeans mempunyai

sekumpulan software modul yang dipakai untuk membuat suatu aplikasi. Modul merupakan arsip (Java Archive) yang memuat kelas-kelas Java yang berinteraksi dengan NetBeans Open API.

2.2.9. Report

Supriyatno (2010:3) IReport merupakan software/laporan pada tampilan GUI, sehingga anda tidak perlu mengetik secara manual file XML untuk membuat template reportnya.