

BAB II

LANDASAN TEORI

2.1 Konsep dasar sistem

Sistem sudah tercipta sebelum manusia diciptakan. banyak sekali sistem alami yang tanpa kita sadari, seperti halnya sistem tata surya. Bagaimana sistem itu terorganisir saling terhubung satu sama lain untuk mencapai sebuah tujuan penyeimbangan alam semesta. Begitu halnya dengan sistem informasi akuntansi dimana setiap sub sistem akuntansi yang saling berhubungan satu sama lain untuk menghasilkan keputusan keuangan yang diambil oleh pemakainya.

2.1.1 Perancangan Sistem

“Perancangan sistem merupakan sekumpulan prosedur yang dilakukan untuk mengubah spesifikasi logis menjadi desain yang dapat diimplementasikan kesistem komputer organisasi” Diana dan Setiawan (2011:48) .

“Perancangan sistem sangat mirip dengan tata arsitektur suatu rumah. Dalam tahap perencanaan, arsitek merumuskan rencana umum yang cocok dengan tata letak keseluruhan. Pada tahap perancangan, hasil rancangannya dapat dimanfaatkan oleh para ahli listrik, tukang semen, dan tukang kayu. Dalam pola serupa, perancang sistem membuat rancangannya yang dapat diimplementasikan oleh akuntan, pemrogram komputer” Puspitawati dan Anggadini (2011:151) .

Menurut Yakub (2012:145) menyatakan bahwa” Perancangan sistem didefinisikan sebagai penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen terpisah kedalam satu kesatuan yang utuh dan berfungsi”.

Berdasarkan kumpulan teori diatas dapat disimpulkan bahwa pengertian perancangan sistem itu ialah sekumpulan elemen prosedur yang dikumpulkan melalui rancangan yang dapat diimplementasikan.

2.1.2 Definisi Sistem

Sistem merupakan serangkaian bagian yang saling tergantung dan bekerja sama untuk mencapai tujuan tertentu. Suatu sistem pasti tersusun dari sub-sub sistem yang lebih kecil yang juga saling bergantung dan bekerja sama untuk mencapai tujuan. (Diana dan Setiawan 2011:3)

“Sistem didefinisikan sebagai sekumpulan prosedur yang saling berkaitan dan saling terhubung untuk melakukan suatu tugas bersama-sama”. (Menurut Pratama 2014:7)

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

Menurut Puspitawati dan Anggadini (2011:2) menyatakan bahwa “sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu”.

Berdasarkan teori diatas dapat disimpulkan bahwa sistem adalah sekumpulan beberapa elemen yang saling berkaitan satu sama lain untuk mencapai tujuan yang sama.

2.1.3 Karakteristik Sistem

Supaya sistem itu dikatakan sistem yang baik memiliki karakteristik yang seperti ditulis oleh Hutahaean (2014:3) dalam bukunya, diantaranya adalah:

1. Komponen (*Component*)

Suatu sistem terdiri dari sejumlah komponen-komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan.

Komponen sistem terdiri dari komponen yang berupa subsistem atau bagian-bagian dari sistem.

2. Batasan Sistem (*boundary*)

Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem (*enviromten*)

Lingkungan Luar Sistem (*enviromten*) adalah diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan dapat bersifat menguntungkan yang harus tetap dijaga dan yang merugikan harus dijaga dan dikendalikan, kalau tidak akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung Sistem (*interface*)

Penghubung Sistem merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari subsistem ke subsistem lain. Keluaran (*output*)

dari subsistem akan menjadi masukan (*input*) untuk subsistem lain melalui penghubung.

5. Masukkan Sistem (*input*)

Masukkan adalah energi yang dimasukkan ke dalam sistem, yang dapat berupa perawatan (*maintenance input*), dan masukkan sinyal (*signal input*). *Maintenance Input* adalah energi yang dimasukkan agar sistem dapat beroperasi. *Signal Input* adalah energi yang diproses untuk didapatkan keluaran. Contoh dalam sistem komputer program adalah *maintenance input* sedangkan data adalah signal input untuk diolah menjadi informasi.

6. Keluaran Sistem (*output*)

Keluaran Sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Contoh komputer menghasilkan panas yang merupakan sisa pembuangan, sedangkan informasi adalah keluaran yang dibutuhkan.

7. Pengolah Sistem (*processing system*)

Suatu sistem menjadi bagian pengolah yang akan merubah masukan menjadi keluaran. Sistem produksi akan mengolah bahan baku menjadi bahan jadi. Sistem akuntansi akan mengolah data menjadi laporan-laporan keuangan.

8. Sasaran Sistem (*target system*)

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Sasaran dari sistem sangat menentukan input yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

2.1.4 Klasifikasi Sistem

Menurut Hutahaean (2014:6) Sistem dapat diklasifikasikan dalam beberapa sudut pandang yaitu:

1. Klasifikasi sistem sebagai :
 - a. Sistem Abstrak (*Abstract System*)

Sistem Abstrak adalah sistem yang berupa pemikiran-pemikiran atau ide-ide yang tidak tampak secara fisik.
 - b. Sistem Fisik (*Physical System*)

Sistem Fisik adalah sistem yang ada secara fisik.
2. Sistem diklasifikasikan sebagai :
 - a. Sistem Alamiyah (*natural system*)

Sistem Alamiyah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia. Misalnya sistem perputaran bumi.
 - b. Sistem Buatan Manusia (*human made system*)

Sistem Buatan Manusia adalah sistem yang dibuat oleh manusia yang melibatkan interaksi antara manusia dan mesin (*human machine system*).
3. Sistem diklasifikasikan sebagai :
 - a. Sistem Tertentu (*deterministic system*)

Sistem Tertentu adalah sistem yang beroperasi dengan tingkah laku yang sudah dapat diprediksi, sebagai keluaran sistem yang dapat diramalkan.
 - b. Sistem Tak Tentu (*probabilistic system*)

Sistem Tak Tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probalistik.

4. Sistem diklasifikasikan sebagai :

a. Sistem Tertutup (*close system*)

Sistem Tertutup adalah sistem yang tidak terpengaruh dan tidak berhubungan dengan lingkungan luar, sistem bekerja otomatis tanpa ada turut campur lingkungan luar. Secara teoritis sistem tertutup ini ada, kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanya *relatively closed system*.

b. Sistem Terbuka (*open system*)

Sistem Terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima input dan output dari lingkungan luar atau subsistem lainnya. Karena sistem terbuka terpengaruh lingkungan luar maka harus mempunyai pengendali yang baik.

2.1.5 Definisi Informasi

Menurut Puspitawati dan Anggadini (2011:13) “Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

“Informasi merupakan hasil pengolahan data dari satu atau berbagai sumber, yang kemudian diolah, sehingga memberikan nilai, arti, dan manfaat.(Pratama 2014:8)

Menurut McLeod dalam buku Yakub (2012:8) “Informasi adalah data yang diolah menjadi bentuk lebih berguna dan lebih berarti bagi yang menerimanya.

Berdasarkan teori diatas dapat disimpulkan bahwa informasi adalah data yang diproses melalui pengumpulan dari sumber-sumber menjadikan informasi yang memiliki arti.

2.1.6 Definisi Sistem Informasi

Menurut O'Brian dalam buku Yakub (2012:17) sistem informasi merupakan kombinasi teratur dari orang-orang, perangkat keras (*hardware*), perangkat lunak (*software*), jaringan komunikasi, dan sumber daya, data yang mengumpulkan, mengubah, menyebar informasi dalam sebuah organisasi.

Menurut Azhar Susanto dalam buku Puspitawati dan Anggadini (2011:14) sistem informasi merupakan komponen-komponen dari sub sistem yang saling berhubungan dan bekerja sama secara harmonis untuk mencapai satu tujuan yaitu mengolah data menjadi informasi.

Sistem Informasi merupakan gabungan dari empat bagian utama. Keempat bagian utama tersebut mencakup perangkat lunak (*software*), perangkat keras (*hardware*), infrastruktur, dan sumber daya manusia (SDM) yang terlatih. (Pratama 2014:10)

Berdasarkan teori diatas dapat disimpulkan bahwa Sistem Informasi adalah gabungan dari suatu perangkat keras yang dapat memproses suatu data yang telah di kumpulkan untuk menjadikan suatu informasi dan dapat disebar luaskan.

2.1.7 Definisi Sistem Informasi Akuntansi

Sistem Informasi Akuntansi adalah “sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan dengan transaksi keuangan.(Diana dan setiawati 2011:4)

Menurut Puspitawati dan Anggadini (2011:57) Sitem Informasi Akuntansi adalah “Suatu sistem yang berfungsi untuk mengorganisasi formulir, catatan dan laporan yang dikoordinasi untuk menghasilkan informasi keuangan yang dibutuhkan dalam pembuatan keputusan manajemen dan pimpinan perusahaan dan dapat memudahkan pengelolaan perusahaan.

Dari kumpulan teori diatas dapat disimpulkan bahwa Sistem Informasi Akuntansi ialah sistem yang bertujuan untuk mengumpulkan data, laporan untuk menghasilkan informasi yang berkaitan dengan keuangan.

2.1.8 Sumbangan Pengembangan Pendidikan (SPP)

Pembayaran SPP merupakan ketentuan yang harus dibayar oleh siswa/i setiap bulannya, tujuan siswa/i melakukan pembayaran sekolah sebagai salah satu sarana untuk digunakan bagi keperluan dari kewajiban penyelenggaraan dan pembinaan pendidikan. Pembayaran SPP merupakan sebagian dari kewajiban siswa/i sebagai timbal balik atas jasa yang didapat dari tenaga pendidik, selain itu sebagai wujud ikut serta memajukan pendidikan dan kebudayaan yang diselenggarakan oleh pemerintah sehingga dapat membantu mewujudkan cita-cita bangsa menjadi masyarakat yang maju baik didalam segi teknologi, ekonomi dan budaya.

2.2 Peralatan pendukung (*tools system*)

Data adalah fakta-fakta mentah yang harus diolah dan disusun serta diorganisir dari suatu peristiwa yang terjadi sehingga nantinya bisa untuk dipahami dan menghasilkan suatu informasi yang berguna.

Tools System atau peralatan pendukung yang penulis gunakan dalam penulisan Tugas Ilmiah ini meliputi OOP, UML, MySQL, NetBeans IDE 8.0.1, ERD dan LRS yang membantu penulis dalam pembuatan Tugas Akhir ini.

2.2.1 Object Oriented Programming (OOP)

Menurut Rossa dan Shalahuddin (2015:100) “Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadap nya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientasi objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas. Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek dan pengujian berorientasi objek”. Keuntungan menggunakan metodologi berorientasi objek adalah sebagai berikut :

1. Meningkatkan produktivitas

Karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lainya 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 dengan 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)

Menurut Rosa dan Shalahudin (2015:133) Menyatakan bahwa” *Unified Modeling Language*(UML) adalah salah standar bahasa yang banyak digunakan didunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemograman berorientasi objek”.

Pada jenis pemograman berbasis objek (*object oriented*), misalkan dengan bahasa java, digunakan pemodelan UML. UML (*unified Modeling Language*) adalah standarisasi internasional untuk notasi dalam bentuk grafik, yang menjelaskan tentang analisis dan desain perangkat lunak yang dikembangkan dengan pemograman berorientasi objek”. (Pratama 2014:48)

2.2.2.1 Use case Diagram

Use case atau diagram *usecase* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang dibuat (Rosa dan Shalahudin 2015:155). *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui

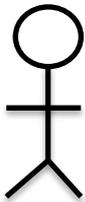
fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja berhak menggunakan fungsi-fungsi itu.

Syarat penamaan pada *use case* adalah nama didefinisikan sesimple mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

- a. Aktor merupakan 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.
- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Adapun simbol-simbol yang digunakan pada diagram *use case* sebagai berikut:

Tabel II.1
Simbol Diagram Use Case

Simbol	Deskripsi
<i>Usecase</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 <i>usecase</i>
Aktor / <i>actor</i> 	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 diawal frase nama aktor
Asosiasi / <i>association</i> 	Komunikasi antara aktor dan <i>usecase</i> yang berpartisipasi pada <i>usecase</i> atau <i>usecase</i> memiliki interaksi dengan aktor
Ekstensi / <i>extend</i>	Relasi <i>usecase</i> tambahan ke sebuah <i>usecase</i> dimana <i>usecase</i> yang

	<p>ditambahkan dapat dapat berdiri sendiri walaupun tanpa <i>usecase</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemograman berorientasi objek, biasanya usecase tambahan memiliki nama depan yang sama dengan <i>usecase</i> yang ditambahkan .</p>
---	---

Sumber: Rossa dan Shalahuddin (2015:155)

2.2.2.2 Activity Diagram

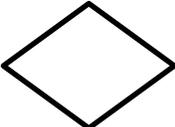
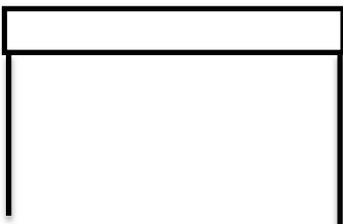
Menurut Rossa dan Shalahuddin (2015:161) Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas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

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

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

Adapun tabel simbol yang digunakan pada diagram aktifitas sebagai berikut:

Tabel II.2
Simbol *Diagram activity*

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
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber: Rossa dan Shalahuddin (2015:161)

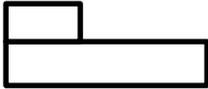
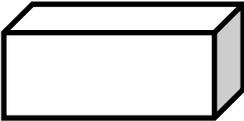
2.2.2.3 Deployment Diagram

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

1. sistem tambahan (*embedded system*) yang menggambarkan rancangan *device*, *node*, dan *hardware* .
2. sistem *client/server*
3. sistem terdistribusi murni
4. rekayasa ulang aplikasi

Adapun tabel simbol yang digunakan pada *diagram deployment* sebagai berikut:

Tabel II.3
Simbol *diagram deployment*

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

Sumber: Rossa dan Shalahuddin (2015:154)

2.2.2.4 Sequence Diagram

Menurut Rossa dan Shalahuddin (2015:165) diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena

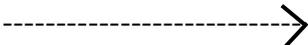
itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

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.

Adapun tabel simbol yang digunakan pada *sequence diagram* sebagai berikut:

Tabel II.4
Simbol *diagram sequence*

Simbol	Deskripsi
Aktor  Atau  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 diawal frase nama aktor
Garis hidup / <i>lifeline</i> 	Menyatakan kehidupan suatu objek
Objek 	Menyatakan objek yang beriteraksi pesan

Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang berhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.
Pesan tipe <i>create</i> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
Pesan tipe <i>call</i> 	Menyatakan suatu objek memanggil operasi/ metode yang ada pada objek lain atau pada dirinya sendiri, arah panah mengarah pada objek yang memiliki operasi / metode, karena ini memanggil operasi/ metode maka operasi / metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi
Pesan tipe <i>send</i> 	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
Pesan tipe <i>return</i> 	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
Pesan tipe <i>destory</i> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada <i>destory</i>

Sumber: Rossa dan Shalahuddi (2015:165)

2.2.3 Basis Data (*Database*)

Menurut Pratama (2014:16) menyatakan bahwa “Basis Data(*Database*) pada sistem informasi berfungsi sebagai media untuk menyimpan data dan informasi yang dimiliki oleh sistem informasi bersangkutan.

Menurut Rossa dan Shalahuddin (2015:42) menyatakan bahwa” Basis Data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data

yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya Basis Data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.

Menurut Yakub (2012:51) menyatakan bahwa “Basis Data (*database*) merupakan kumpulan data yang saling berhubungan (punya relasi), Relasi biasanya ditunjukkan dengan kunci (*key*) dari tiap file yang ada.

Jadi dapat disimpulkan dari keterangan diatas bahwa Basis Data (*data base*) adalah kumpulan data yang memiliki sistem informasi yang dapat disimpan dan dapat diakses dengan mudah dan cepat.

2.2.4 Structured Query Language (SQL)

MySQL ialah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis (Pipiet 2013:5)

Menurut Rossa dan Shalahuddin (2015:46) “SQL adalah bahasa yang digunakan untuk mengelola data pada RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus.”

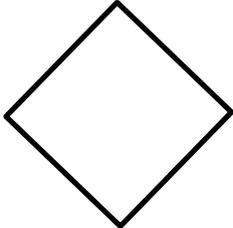
Dapat disimpulkan bahwa SQL adalah bahasa yang digunakan dalam pengolahan data pada database yang terorganisir secara akurat dan bisa dihubungkan untuk setiap tabel yang memiliki data pada tabel lain.

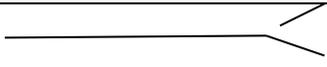
2.2.5 Entity Relationship Diagram (ERD)

Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan adalah menggunakan *Entity Relationship Diagram* (ERD). ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan *Object Oriented DataBase Management Sytem*

(OODBMS) maka perancangan basis data tidak perlu menggunakan ERD. ERD memiliki beberapa aliran notasi seperti notasi chen (dikembangkan oleh Peter Chen) Berikut adalah simbol-simbol yang digunakan pada ERD dengan notasi Chen” (Rossa dan Shalahuddin 2015:50):

Tabel II.5
Simbol ERD dengan notasi Chen

Simbol	Deskripsi
Entitas / <i>entity</i> 	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
Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas
Atribut kunci primer 	<i>Field</i> atau kolom data yang butuh disimpan didalam 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 multiniali / <i>multivalue</i> 	<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 / <i>association</i>	Penghubung antara relasi dan entitas dimana dikedua ujungnya memiliki

	<i>multiplicity</i> kemungkinan jumlah pemakaian, kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas.
---	---

Sumber: Rossa dan Shalahuddin (2015:50)

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

2.2.7 NetBeans IDE 8.0.

Netbeans adalah satu aplikasi ”*Integrate Develoment Environment (IDE)* yang digunakan oleh *developer software* komputer untuk menulis, meng-*complie*, mencari kesalahan, dan untuk menyebarkan program. Netbeans mempunyai sekumpulan *software* modul yang dipakai untuk membuat suatu aplikasi. Modul

merupakan arsip java (*java archive*) yang memuat kelas-kelas java yang berinteraksi dengan Netbeans”. (Pipiet 2015:20)

2.2.8 *iReport*

Menurut Kurniawan, Eri dan Nur (2011:38), *iReport* adalah *report designer visual* yang dibangun pada *JasperReport*. *iReport* bersifat intuitif dan mudah digunakan pembangun laporan *visual* atau desainer untuk *JasperReport* dan tertulis dalam kitab Java. Sebagai alternatif, terdapat *tools iReport* (dengan library *JasperReport*) yang dapat membantu dalam pembuatan laporan. Library *JasperReport* sendiri merupakan Java Library (JAR) yang bersifat *open* dan dirancang untuk menambahkan kemampuan pelaporan (*reporting capabilities*) pada aplikasi java. *JasperReport* memiliki sejumlah fitur, diantaranya adalah sebagai berikut:

1. *Layout dan desain* laporan yang *fleksibel*.
2. Dapat menampilkan laporan dalam bentuk *teks* maupun gambar (*chart*).
3. Dapat menghasilkan report dalam berbagai format, seperti: *html, pdf, rtf, xls, dan csv*.
4. Dapat menerima data dari berbagai sumber data, seperti: *JDBC, Bean Collection, ResultSet, CSV, XML, dan Hibernate*.