

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Dalam proses pengembangan sistem informasi, diperlukan adanya pemahaman mengenai konsep-konsep dari sistem informasi tersebut. Pendekatan sistem yang merupakan jaringan kerja dari prosedur lebih menekankan urutan-urutan operasi didalam sistem. Sistem juga dapat diartikan suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau tujuan tertentu.

Teori tentang sistem melahirkan konsep-konsep futuristik seperti sibernetika (*cybernetics*), konsep yang berkaitan dengan upaya-upaya untuk menerapkan berbagai disiplin ilmu perilaku, fisika, biologi dan teknik. Unsur-unsur yang mewakili suatu sistem secara umum adalah masukan (*input*), pengolahan (*processing*) dan keluaran (*output*). Disamping itu suatu sistem senantiasa tidak terlepas dari lingkungan sekitarnya. Maka umpan balik (*feed back*) dapat berasal dari *output* atau dari lingkungan sistem yang di maksud.

2.1.1. Pengertian Sistem

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

Menurut Mulyani (2016:2) menyatakan bahwa “sistem bisa diartikan sebagai sekumpulan sub sistem, komponen ataupun element yang saling bekerja

sama dengan tujuan yang sama untuk menghasilkan *output* yang sudah ditentukan sebelumnya”.

Menurut Djahir dan Pratita (2015:7) mengemukakan bahwa “sistem dikelompokkan menjadi dua bagian yang menekankan pada prosedurnya dan ada yang menekankan pada elemennya. kedua kelompok ini adalah benar dan tidak bertentangan, yang berbeda adalah cara pendekatannya”.

Dari pengertian di atas dapat disimpulkan bahwa sistem adalah sekumpulan komponen, jaringan atau himpunan kerja dari prosedur-prosedur atau variabel-variabel yang saling berkaitan dan saling bekerja sama, terorganisasi, saling berinteraksi, dan saling bergantung sama lain untuk membentuk suatu jaringan kerja agar dapat mencapai sasaran atau tujuan tertentu.

2.1.2. Karakteristik Sistem

Menurut Hutahean (2015:3) supaya sistem itu dikatakan sistem yang baik memiliki karakteristik, yaitu:

1. Komponen

Suatu sistem terdiri dari jumlah 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 (*environment*)

Batas dari sistem yang mempengaruhi operasi sistem. Lingkungan dapat bersifat menguntungkan yang harus tetap dijaga dan yang merugikan yang harus dijaga dan dikendalikan, kalau tidak akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung sistem (*interface*)

Merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari subsistem akan menjadi masukan (*input*) untuk subsistem lain melalui penghubung.

5. Masukan (*input*)

Energi yang dimasukkan kedalam sistem, yang dapat berupa perawatan (*maintanance input*), dan masukan sinyal (*signal input*). *Maintanance input* adalah energi yang dimasukkan agar sistem dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran. Contoh dalam sistem computer program adalah *maintanance input* sedangkan data adalah *signal input* untuk diolah menjadi informasi.

6. Keluaran sistem (*output*)

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

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

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

2.1.3. Klasifikasi Sistem

Sistem dapat diklasifikasikan berdasarkan sudut pandang dari pengguna sistem. Menurut Hutahean (2015:5) 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 alamiah (*natural system*)

Sistem alamiah 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 dengan 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 (*probalistic system*)

Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilistik.

4. Sistem diklasifikasikan sebagai:

a. Sistem tertutup (*close system*)

Sistem tertutup adalah sistem yang tidak berpengaruh 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.

2.1.4. Pengertian Informasi

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

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

Menurut Hutahaeen (2015:9) “informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya”.

Dapat disimpulkan bahwa informasi merupakan data yang diolah menjadi bentuk yang berguna untuk membuat keputusan. Informasi berguna untuk membuat keputusan karena informasi ketidak pastian (atau meningkatkan pengetahuan). Informasi menjadi penting karena berdasarkan informasi itu para pengelola dapat mengetahui hasil pengolahandata atau fakta yang dikumpulkan dengan metode ataupun cara-cara tertentu.

Menurut Romney dan Steinbart dalam Mulyani (2016:18) mengemukakan bahwa kriteria informasi yang terdiri dari:

1. Relevan

Informasi harus bisa dikatakan relevan apabila informasi yang termuat di dalamnya dapat mempengaruhi keputusan pengguna dengan membantu mereka mengevaluasi peristiwa masa lalu atau masa kini, dan memprediksi masa depan, serta menegaskan atau mengoreksi hasil evaluasi mereka di masa lalu.

2. Andal

Informasi harus bebas dari pengertian yang menyesatkan dan kesalahan material, menyajikan setiap fakta secara jujur, serta dapat diverifikasi.

3. Lengkap

Informasi disajikan selengkap mungkin, yaitu mencakup semua informasi yang dapat mempengaruhi pengambilan keputusan.

4. Tepat waktu

Informasi disajikan tepat waktu sehingga dapat berpengaruh dan berguna dalam pengambilan keputusan.

5. Dapat dipahami

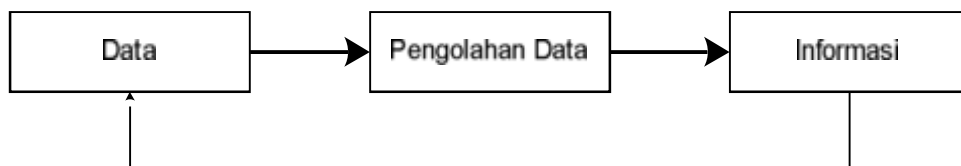
Informasi yang disajikan dalam bentuk serta istilah yang disesuaikan dengan batas pemahaman para pengguna.

6. Dapat diverifikasi

Informasi yang disajikan dapat diuji, dan apabila pengujian dilakukan lebih dari sekali oleh pihak yang berbeda, hasilnya tetap menunjukkan simpulan yang tidak berbeda jauh.

7. Dapat diakses

Informasi yang tersedia pada saat dibutuhkan dan dengan format yang dapat digunakan.



Sumber: Mulyani (2016:21)

Gambar II.1. Ilustrasi Antara Hubungan Data dan Informasi

2.1.5. Pengertian Sistem Informasi

Menurut Yakub dalam bukunya Muslihudin dan Oktafianto (2016:11) “sistem informasi merupakan suatu kumpulan dari komponen-komponen dalam organisasi yang berhubungan dengan proses penciptaan aliran informasi”.

Hutahaean (2015:13) mengemukakan bahwa:

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan.

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

Dari beberapa definisi diatas dapat disimpulkan bahwa sistem informasi merupakan kegiatan atau aktifitas yang melibatkan suatu organisasi yang saling mempertemukan guna untuk saling mendapatkan informasi pada pengelolaan transaksi dengan laporan yang dibutuhkan dan harus dapat diandalkan sebagai sarana untuk memudahkan setiap pelaksanaan aktifitas yang terjadi.

Komponen/elemen yang dimaksud disebut dengan blok bangunan (*building block*). Adapun uraian dari blok bangunan (Hutahaean, 2015:13) tersebut yaitu:

1. Blok masukan (*input block*)

Input mewakili data yang masuk ke dalam sistem informasi. *Input* disini termasuk metode-metode dan media yang digunakan untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen dasar.

2. Blok model (*model block*)

Blok ini terdiri dari kombinasi prosedur, logika dan metode matematik yang akan memanipulasi data *input* dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang sudah diinginkan.

3. Blok keluaran (*output block*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok teknologi (*technology block*)

Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian diri secara keseluruhan. Teknologi terdiri dari unsure utama yaitu:

- a. Teknisi (*humanware* atau *brainware*)
- b. Perangkat lunak (*software*)
- c. Perangkat keras (*hardware*)

5. Blok basis data (*database block*)

Merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

6. Blok kendali (*control block*)

Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah atau bila terlanjur terjadi kesalahan dapat langsung diatasi.

2.1.6. Pengertian Sistem Informasi Akuntansi

Menurut Mulyani (2016:24) mengemukakan bahwa “sistem informasi akuntansi digunakan sebagai alat untuk melakukan analisis keputusan ataupun sebagai pembuat keputusan yang terkait dengan transaksi-transaksi perusahaan”.

Menurut Mahatmyo (2014:9) mengemukakan bahwa “sistem informasi akuntansi merupakan sekelompok struktur dalam sebuah entitas yang mengelola sumber daya fisik dan sumber daya lain untuk mengubah data ekonomi menjadi informasi akuntansi, agar dapat memenuhi kebutuhan informasi berbagai pihak”.

Berdasarkan pengertian dari para ahli di atas, dapat disimpulkan bahwa sistem informasi akuntansi adalah kumpulan elemen dan prosedur yang berada dalam suatu entitas yang mengelola sumber daya fisik dan sumber daya lain untuk melakukan analisis keputusan atau sebagai pembuat keputusan yang terkait dengan data ekonomi menjadi informasi akuntansi yang memenuhi kebutuhan informasi berbagai pihak.

Menurut Romney dan Steinbart dalam Mulyani (2016:26) terdapat enam (6) komponen dalam sistem informasi akuntansi, yaitu:

1. *User*, yaitu orang yang menggunakan atau mengoperasikan sistem.
2. *Procedure* atau *instructions*, yaitu pemrosesan dan penyimpanan data kegiatan organisasi.
3. *Data*, yaitu representasi dari dunia nyata terkait dengan organisasi.
4. *Software*, yaitu kumpulan program komputer yang digunakan untuk memproses data.
5. *Information technology infrastructure*, yaitu struktur yang akan digunakan oleh sistem seperti misalnya, struktur jaringan komputer.

6. *Internal control and integrity measure.*

2.1.7. **Laporan Keuangan**

Menurut Harahap (2008:105) menyatakan bahwa “laporan keuangan menggambarkan kondisi keuangan dan hasil usaha suatu perusahaan pada saat tertentu atau jangka waktu tertentu”.

Menurut Priyatno (2009:89) mengemukakan bahwa:

Laporan keuangan merupakan laporan yang menginformasikan tentang posisi keuangan perusahaan yang tersusun secara rinci dan lengkap yang meliputi neraca, laba rugi, neraca saldo dan sebagainya. Laporan ini digunakan bagi pihak *internal* dan eksternal untuk dapat menilai suatu perusahaan, seperti menilai keberhasilan perusahaan dalam usahanya atau menilai perkembangan perusahaan tersebut.

Sedangkan menurut Kasmir (2008:7) “secara umum laporan keuangan adalah laporan yang menunjukkan kondisi keuangan perusahaan pada suatu periode tertentu”.

Jadi dapat disimpulkan bahwa laporan keuangan adalah suatu proses yang merupakan ringkasan transaksi hasil akhir dari kegiatan akuntansi yang mencerminkan kondisi keuangan yang terjadi selama satu tahun. Informasi tentang kondisi keuangan dan hasil operasi perusahaan sangat berguna dari berbagai pihak, baik pihak-pihak yang berada didalam (*intern*) perusahaan maupun yang berada diluar (eksternal) perusahaan.

Menurut Kusrini dan Koniyo (2007:38) laporan keuangan terdiri dari empat (4) laporan dasar, yaitu:

1. Neraca

Neraca adalah aktiva (harta). Kewajiban dan modal pemilik sebagai suatu kesatuan bisnis pada waktu tertentu, biasanya pada tanggal akhir bulan atau akhir bulan.

2. Laporan Laba Rugi

Laporan laba rugi adalah ringkasan pendapatan (*revenue*) dan beban (*expenses*) dari satu kesatuan bisnis (*business entity*) untuk periode tertentu.

3. Laporan Perubahan Modal atau laba ditahan

Laporan perubahan modal atau laba ditahan adalah ringkasan perubahan modal pemilik sebagai satu kesatuan bisnis yang terjadi selama suatu periode tertentu.

4. Laporan Arus Kas

Laporan arus kas adalah ringkasan penerimaan dan pengeluaran kas dari satu kesatuan bisnis untuk suatu periode tertentu.

2.2. Peralatan Pendukung (*Tools System*)

Peralatan pendukung (*tools system*) berupa alat yang digunakan untuk menggambarkan bentuk logika dari suatu sistem yang menggunakan simbol-simbol, lambang, diagram yang menunjukkan secara tepat arti dan fungsinya. Adapun peralatan pendukung guna untuk merancang suatu sistem.

2.2.1. *Object Oriented Programming (OOP)*

Pemrograman berorientasi objek (OOP) sering digunakan oleh para pengembang program yang tidak hanya mendefinisikan tipe data dari struktur data tetapi juga tipe dari operasi (fungsi) yang dapat di aplikasikan dari struktur data pada cara ini struktur data menjadi objek yang memiliki data dan fungsi. Sebagai tambahan, pengembang dapat membuat relasi antara suatu objek dengan yang lainnya.

Menurut Salim dan Shalahuddin (2015:100) “pemograman berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya”.

Menurut Supardi (2010:320) menyatakan bahwa:

OOP adalah cara berpikir, pandangan atau paradigma baru untuk membuat program atau merancang sistem dengan memperhatikan objek, ciri objek dan perilakunya. OOP ini sangat berbeda dengan membuat program prosedural yang memperhatikan aspek *input*, proses dan *output*.

Jadi OOP dapat disimpulkan bahwa OOP adalah pemograman yang dengan cara berpikirnya dapat mengaplikasikan struktur data menjadi sebuah objek.

Konsep OOP menekankan sebagai berikut (Salim dan Shalahuddin, 2015:100):

1. Kelas (*class*)

Kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh “*class of dog*” adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku atau turunan dari anjing. Sebuah *class* adalah dasar dari modularitas dan struktur dalam pemograman berorientasi objek.

2. Metode (*method*)

Metode merupakan suatu operasi berupa fungsi-fungsi yang dapat dikerjakan oleh suatu objek. *Method* didefinisikan pada *class* akan tetapi dipanggil melalui objek. Metode menentukan perilaku objek, yakni apa yang terjadi ketika objek itu dibuat serta berbagai operasi yang dapat dilakukan objek sepanjang hidupnya.

3. Abstraksi (*abstraction*)

Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti, setiap objek dalam sistem melayani sebagai model dari “perilaku” abstrak yang dapat melakukan kerja, laporan dan perubahan keadaan, dan berkomunikasi dengan objek lainnya.

4. Objek (*object*)

Abstraksi dan sesuatu yang mewakili dunia nyata seperti benda, manusia, satuan organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya.

5. Enkapsulasi (*encapsulation*)

Enkapsulasi adalah pembungkusan atribut data dan layanan (operasi-operasi) yang dimiliki objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya.

6. Pewarisan (*inheritance*)

Pewarisan adalah mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dari dirinya.

7. Antarmuka (*interface*)

Antarmuka atau *interface* sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi.

8. *Reusability*

Reusability adalah pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tertentu.

9. Generalisasi dan spesialisasi

Generalisasi dan spesialisasi menunjukkan hubungan antara kelas dan objek yang khusus. Misalnya kelas yang lebih umum (generalisasi) adalah kendaraan darat dan kelas khususnya (spesialisasi) adalah mobil, motor.

10. Komunikasi antar objek

Komunikasi antar objek dilakukan lewat pesan (*message*) yang dikirim dan satu objek ke objek lainnya.

11. Polimorfisme (*polymorphism*)

Polimorfisme adalah kemampuan suatu objek untuk digunakan dibawah tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.

12. *Package*

Package merupakan sebuah kontainer atau kemasan yang dapat digunakan untuk mengelompokkan kelas-kelas sehingga memungkinkan beberapa kelas yang bernama sama disimpan dalam *package* yang berbeda.

2.2.2. ***Unified modeling language (UML)***

Sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau perangkat lunak berbasis objek sering menggunakan *unified modeling language (UML)*. UML juga menjadi salah satu cara untuk mempermudah pengembangan aplikasi yang berkelanjutan.

Menurut Nugroho (2010:6) mengemukakan bahwa:

Unified modeling language adalah bahasa permodelan untuk sistem atau perangkat lunak yang berparadigma atau berorientasi objek. Permodelan atau modeling sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

Menurut Salim dan Shalahuddin (2015:137) “*Unified modeling language* adalah teknik pemograman berorientasi objek, memunculkan sebuah standarisasi bahasa permodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik berorientasi objek”.

Dapat disimpulkan bahwa *unified modeling language* bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak sebagai dasar pemodelan menggunakan teknik berorientasi objek.

1. *Use case diagram*

Menurut Muslihudin dan Oktafianto (2016:64) mengemukakan bahwa “*use case* menggambarkan external view dari sistem yang akan kita buat modelnya”.

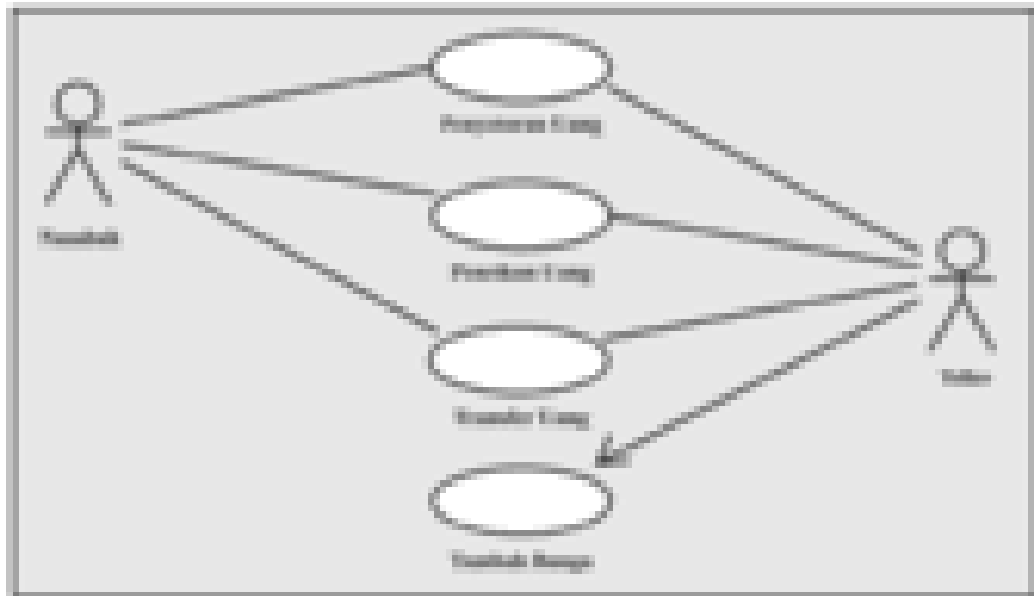
Menurut Fatta (2007:91) mengemukakan bahwa:

use case adalah metode berbasis teks untuk menggambarkan dan mendokumentasikan proses yang kompleks. *Use case* menambahkan detail untuk kebutuhan yang telah dituliskan pada definisi sistem kebutuhan. *Use case* dikembangkan oleh analis sistem bersama-sama dengan pengguna. Pada tahapan selanjutnya, berdasarkan *use case* ini, analis menyusun model data dan model proses.

Berdasarkan kutipan para ahli di atas, dapat disimpulkan bahwa *use case diagram* merupakan diagram yang digunakan untuk menggambarkan secara ringkas siapa yang menggunakan sistem dan apa saja yang bisa dilakukannya. Diagram *use case* tidak menjelaskan secara detail tentang penggunaan *use case*, namun hanya memberi gambaran singkat hubungan antara *use case*, aktor, dan sistem.

Komponen pembentuk diagram *use case* (Muslihudin dan Oktafianto, 2016:64) terdiri dari:

- a. Aktor (*actor*), menggambarkan pihak-pihak yang berperan dalam sistem.
- b. *Use case*, aktifitas/sarana yang disiapkan oleh bisnis/sistem.
- c. Hubungan (*link*), aktor mana saja yang terlibat dalam *use case* ini.



Sumber: Muslihudin dan Oktafianto (2016:65)

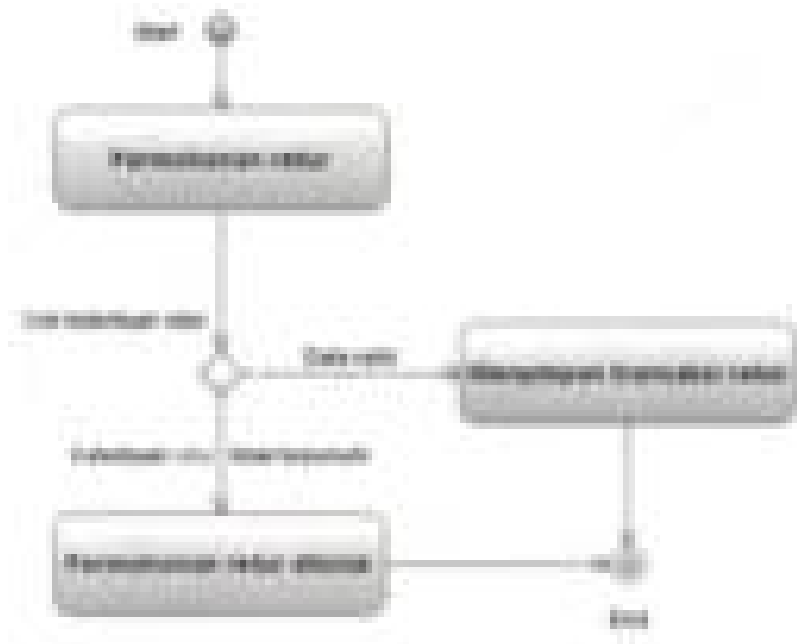
Gambar II.2. Ilustrasi Use case diagram

2. Activity diagram

Menurut Satzinger dalam Triandini dan Suardika (2012:37) menyatakan bahwa “*activity diagram* adalah sebuah diagram alur kerja yang menjelaskan berbagai kegiatan pengguna (atau sistem), orang yang melakukan masing-masing aktivitas, dan aliran sekuensial dari aktivitas-aktivitas tersebut”.

Menurut Nugroho (2010:62) mengemukakan bahwa “*diagram aktivitas (activity diagram)* sesungguhnya merupakan bentuk khusus dari *state machine* yang bertujuan memodelkan komputasi-komputasi dan aliran-aliran kerja yang terjadi dalam sistem/perangkat lunak yang sedang dikembangkan”.

Dapat disimpulkan bahwa *activity diagram* merupakan teknik pemodelan berorientasi objek yang merupakan bentuk khusus dari *state machine* yang menjelaskan berbagai kegiatan pengguna atau sistem dan aliran-aliran kerja yang terjadi pada sistem yang sedang dikembangkan.



Sumber: Huda dan Bunafit Komputer (2010:142)

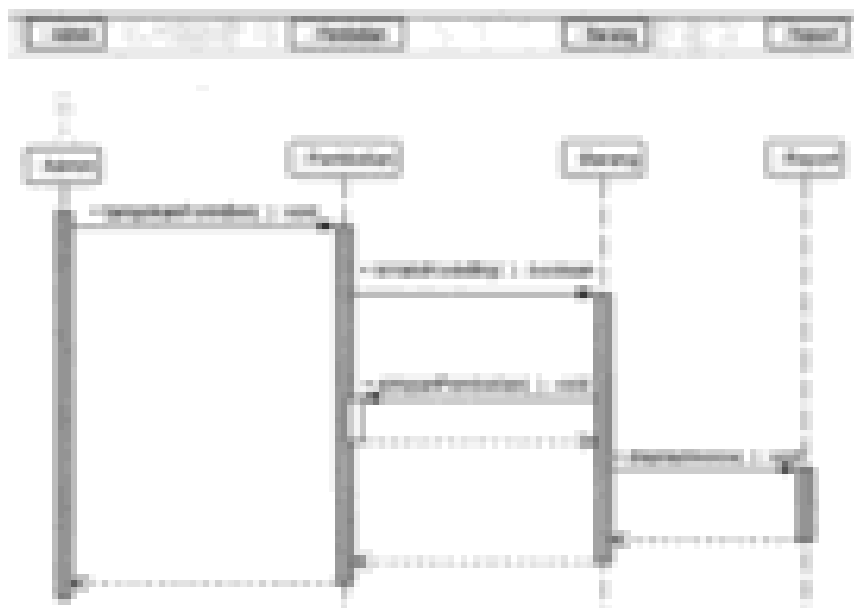
Gambar II.3. Ilustrasi Activity Diagram

3. *Sequence diagram*

Menurut Huda dan Bunafit Komputer (2010:143) mengemukakan bahwa *sequence diagram* adalah “diagram ini menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu”.

Menurut Salim dan Shalahuddin (2015:165) mengemukakan bahwa: 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 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.

Berdasarkan kutipan dari para ahli di atas dapat disimpulkan bahwa *sequence diagram* merupakan teknik pemodelan berorientasi objek yang menggambarkan kelakuan objek, interaksi yang terjadi dan mendeskripsikan waktu hidup objek secara berurutan.



Sumber: Huda dan Bunafit Komputer (2010:143)

Gambar II.4. Ilustrasi *Sequence diagram*

2.2.3. *Entity Relationship Diagram (ERD)*

Menurut Salim dan Shalahuddin (2015:53) “ERD adalah bentuk paling awal dalam melakukan perancangan basis data relasional. Jika menggunakan OODMBS maka perancangan ERD tidak perlu dilakukan”.

Menurut Ladjamudin (2013:142) dalam bukunya menjelaskan “ERD adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak”.

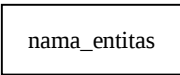
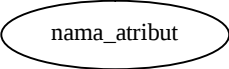
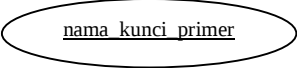
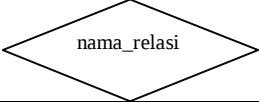
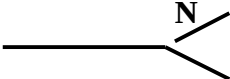
Berdasarkan pengertian para ahli tentang ERD maka penulis menyimpulkan bahwa ERD adalah suatu model yang menggambarkan suatu

penyimpanan basis data yang menggunakan susunan data dalam sistem secara abstrak dan menggunakan OODMBS.

Gambar atau simbol yang terdapat pada *entity relationship diagram* (ERD) sering disebut dengan komponen. Adapun komponen *entity relationship diagram* (ERD) menurut Chen dalam buku Salim dan Shalahuddin (2015:50) disajikan ke dalam bentuk tabel.

Tabel II.1.

Komponen-Komponen *Entity Relationship Diagram* (ERD)

Notasi	Komponen	Keterangan
	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 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/ <i>multivalue</i>	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki lebih dari satu.
	Relasi	Relasi yang menghubungkan antar entitas, biasanya diawali dengan kata kerja.
	Asosiasi/ <i>association</i>	Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i> 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 <i>one to many</i> menghubungkan entitas A dan entitas B.
--	--	---

Sumber: Salim dan Shalahuddin (2015:50)

2.2.4. *Logical Record Structure (LRS)*

Menurut Kadir (2007:18) *logical record structure* (LRS) merupakan “model relasional menggunakan kumpulan tabel-tabel untuk merepresentasikan data dan relasi antar data-data tersebut. Setiap tabel terdiri atas kolom-kolom dan setiap kolom memiliki nama yang unik”. LRS merupakan hasil transformasi diagram E-R (ERD) menggunakan aturan aturan tertentu. Aturan-aturan tersebut yaitu: (1) setiap *entity* akan diubah ke dalam bentuk sebuah kotak dengan nama *entity* berada di luar kotak dan atribut berada di dalam kotak, (2) sebuah relasi kadang disatukan dalam sebuah kotak bersama *entity*, kadang dipisah dalam sebuah kotak tersendiri (Ladjamudin, 2013:159).

Dari pernyataan diatas dapat diartikan bahwa, LRS merupakan hasil kardinalitas dari diagram E-R (ERD) menjadi LRS yang memiliki hubungan searah dari beberapa *record* dan akan mengikuti pola pemodelan tertentu dalam kaitannya dengan konversi LRS.

Kardinalitas yang terjadi pada LRS (Ladjamudin, 2013:160), yaitu:

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

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

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

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

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

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

2.2.5. Java

Menurut Supardi (2010:1) mengemukakan bahwa “java merupakan bahasa pemrograman yang dikembangkan dari bahasa pemrograman C++, sehingga pemrograman ini seperti bahasa C++”.

Java adalah bahasa pemrograman yang dapat dijalankan diberbagai komputer maupun telepon genggam. Bahasa pemrograman ini dibuat oleh James Goslin saat masih bergabung di Sun Microsystems, dimana saat ini merupakan bagian dari Oracle yang dirilis pada tahun 1995. Bahasa ini dapat banyak mengadopsi sintaksis yang terdapat pada C dan C++, tetapi dengan sistaksis model model objek yang lebih sederhana. Aplikasi ini hanya dapat dijalankan pada ruang lingkup yang terbatas. Berbeda dengan aplikasi web yang dapat diakses lewat *browser* dimana saja. Adapun *software database* yang digunakan pada java *desktop* ini yaitu MySQL. MySQL merupakan salah satu sistem *database* yang sangat handal karena menggunakan sistem SQL. Pada awalnya SQL berfungsi sebagai penghubung antara program *database* dengan bahasa pemrograman yang kita gunakan. Dengan adanya SQL maka para pemrograman jaringan dan aplikasi tidak mengalami kesulitan sama sekali didalam

menghubungkan aplikasi yang mereka buat. Setelah itu SQL dikembangkan lagi menjadi sistem *database* dengan munculnya MySQL (Wahana Komputer, 2010:2).

Berdasarkan kutipan para ahli di atas, dapat disimpulkan bahwa java merupakan bahasa pemrograman yang dikembangkan dari bahasa pemrograman C++ dari Sun Microsystems.

1. SQL

Salim dan Shalahuddin (2015:46) mengemukakan bahwa “SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS”. Begitu juga menurut Grolt dalam Fauzi dan Amin (2012:37) mengemukakan bahwa “SQL merupakan sebuah alat untuk melakukan proses organisasi, manajemen, dan pengambilan data yang tersimpan dalam sebuah *database*”.

Dapat disimpulkan bahwa SQL merupakan suatu bahasa standar yang digunakan untuk melakukan proses organisasi, manajemen, dan pengambilan data yang tersimpan di dalam basis data.

2. MySQL

MySQL merupakan *software database* untuk mengelola dan menyimpan data yang jenisnya beraneka ragam dan tipe data relational yang saling berhubungan (Zaki dan SmitDev Community, 2008:94).

Menurut Ahmar (2013:11) mengemukakan bahwa “MySQL adalah sistem yang berguna untuk melakukan proses pengaturan koleksi-koleksi struktur data (*database*) baik meliputi proses pembuatan atau proses pengelolaan *database*”.

Berdasarkan kutipan dari para ahli di atas, dapat disimpulkan bahwa MySQL merupakan aplikasi untuk mengelola basis data yang saling berhubungan yang meliputi proses pengaturan struktur data.

2.2.6. Netbeans

Netbeans merupakan salah satu proyek *open source* yang disponsori oleh Sun Microsystem. Proyek ini berdiri pada tahun 2000 dan telah menghasilkan dua produk yaitu Netbeans IDE dan Netbeans *Platform*. Netbeans IDE merupakan produk yang digunakan untuk melakukan pemrograman baik menulis kode, mengompilasi, mencari kesalahan dan mendistribusikan program. Sedangkan Netbeans *Platform* adalah sebuah modul yang merupakan kerangka awal atau fondasi dalam membangun aplikasi *desktop* yang besar.

Netbeans merupakan aplikasi yang digunakan *programmer* mengolah bahasa pemrograman java yang bersifat gratis dan *open source* (Hendry, 2008:3).

Menurut Nofriadi (2015:4) mengemukakan bahwa “Netbeans merupakan sebuah aplikasi *Integrated Development Environment* (IDE) yang berbasis Java dari Sun Microsystems yang berjalan di atas *swing* dan banyak digunakan sekarang sebagai *editor* untuk berbagai bahasa pemrograman”.



Sumber: Nofriadi (2015:5)

Gambar II.5. Tampilan Netbeans