

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Pemahaman mengenai konsep dasar sistem informasi ini memerlukan pendekatan-pendekatan mengenai sistem, sistem informasi dan sistem informasi akuntansi yang mencakup pengertian, karakteristik sistem, klasifikasi sistem, sistem informasi dan sistem informasi akuntansi.

2.1.1. Pengertian Sistem

Menurut Fauzi (2017:2) mendefinisikan bahwa “Sistem adalah suatu kesatuan yang terdiri dari interaksi subsistem untuk mencapai tujuan yang sama”. Sedangkan menurut Tyoso (2016:1) mengemukakan bahwa, “Sistem merupakan suatu kumpulan dari komponen-komponen yang membentuk satu kesatuan”.

Dapat ditarik suatu kesimpulan bahwa sistem merupakan sekumpulan elemen, komponen atau subsistem yang saling berhubungan, bekerja sama dan membentuk satu kesatuan dalam upaya mencapai tujuan.

2.1.2. Karakteristik Sistem

Sistem memiliki karakteristik atau ciri-ciri agar dikategorikan sebagai suatu sistem yang baik. Karakteristik dari sistem (Fauzi, 2017:2) diuraikan sebagai berikut:

1. Komponen sistem

Suatu sistem terjadi dikarenakan adanya sejumlah komponen yang melakukan interaksi. Suatu sistem yang sekecil apapun akan selalu mengandung komponen-komponen.

2. Batas sistem

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya.

3. Lingkungan luar sistem

Lingkungan luar dari suatu sistem adalah daerah di luar batas dari suatu sistem yang mempengaruhi operasi sistem.

4. Penghubung sistem

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari suatu subsistem ke subsistem yang lainnya. Keluaran dari sistem menjadi masukan untuk subsistem lainnya.

5. Masukan sistem

Masukan sistem adalah energi yang dimasukkan ke dalam sistem.

6. Keluaran sistem

Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisi pembuangan.

7. Pengolah sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan mengubah masukan mejadi keluaran.

8. Sasaran sistem

Suatu sistem pasti mempunyai tujaun (*goal*) atau sasaran (*objective*). Kalau suatu sistem tidak mempunyai sasaran maka sistem tersebut tidak berguna.

2.1.3. Klasifikasi Sistem

Beberapa aspek dari sistem mengizinkan pengguna untuk mengklarifikasikan sistem berdasarkan sudut pandang. Klasifikasi sistem yang dimaksud (Tyoso, 2016:5), yaitu:

1. Sistem Alamiah (*Natural System*) dan Sistem Buatan Manusia (*Artificial System*)
 - a. Sistem alamiah merupakan sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem tata surya, sistem galaksi, sistem reproduksi dan lain-lain.
 - b. Sistem buatan manusia merupakan sistem yang dirancang oleh manusia. Sistem buatan yang melibatkan interaksi manusia, misalnya sistem akuntansi, sistem informasi, dan lain-lain.
2. Sistem Deterministik (*Deterministic System*) dan Sistem Probabilistik (*Probabilistic System*)
 - a. Sistem deterministik merupakan sistem yang beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan, misalnya sistem komputer, adalah contoh sistem yang tingkah lakunya dapat dipastikan berdasarkan program-program komputer yang dijalankan.
 - b. Sistem probabilistik merupakan sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas, misalnya sistem manusia.

3. Sistem Terbuka (*Opened System*) dan Sistem Tertutup (*Closed System*)

- a. Sistem terbuka merupakan sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Lebih spesifik dikenal juga yang disebut dengan sistem terotomasi, yang merupakan bagian dari sistem buatan manusia dan berinteraksi dengan kontrol oleh satu atau lebih komputer sebagai bagian dari sistem yang digunakan dalam masyarakat modern. Sistem ini menerima masukan dan menghasilkan keluaran untuk subsistem lainnya, misalnya sistem kebudayaan manusia.
- b. Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa campur tangan dari pihak luar. Secara teoritis sistem tersebut ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system* (secara relatif tertutup, tidak benar-benar tertutup).

2.1.4. Sistem Informasi

Sistem informasi merupakan kombinasi *software*, *hardware*, *brainware* dan infrastruktur yang kuat yang bekerja sama dalam mengumpulkan, memasukkan, mengontrol dan memproses data untuk mendapatkan hasil berupa informasi lainnya yang mendukung tujuan dari pengguna sistem informasi (Maulana & Purwaningtias, 2016). Sedangkan menurut 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”.

Berdasarkan pendapat dari beberapa ahli di atas, dapat disimpulkan bahwa sistem informasi merupakan sekumpulan dari orang, perangkat lunak, perangkat keras, dan prosedur yang saling berinteraksi, bekerja sama dalam menyelesaikan sesuatu untuk menghasilkan informasi yang dapat dijadikan sebagai dasar pengambilan keputusan.

Komponen-komponen dari sistem disebut dengan blok bangunan (*building block*). Penjelasan blok bangunan (Hutahaean, 2015:13) diuraikan sebagai berikut:

1. Blok masukan (*input block*)

Blok masukan merupakan blok yang bertugas dalam *input* data agar masuk ke dalam sistem informasi. Blok masukan bertugas dalam merekam data yang akan dimasukkan, biasanya berupa dokumen-dokumen dasar.

2. Blok model (*model block*)

Blok model terbentuk dari kombinasi prosedur, logika dan model matematik yang memproses data *input* dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Blok keluaran (*output block*)

Sistem informasi menghasilkan keluaran (*output*) yaitu informasi yang berkualitas dan berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok teknologi (*technology block*)

Teknologi digunakan merupakan kotak alat dalam sistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran berupa informasi dan membantu pengendalian dari sistem secara menyeluruh. Blok teknologi

perangkat lunak (*software*) dan perangkat keras (*hardware*) yang dioperasikan oleh teknisi (*brainware*).

5. Blok basis data (*database block*)

Basis data (*database*) merupakan media untuk menyimpan data yang saling berhubungan satu sama lainnya, tersimpan di perangkat keras komputer dan dapat dipergunakan kembali, diperlukan perangkat lunak untuk memanipulasinya.

6. Blok kendali (*control block*)

Sistem informasi memiliki kontrol kendali untuk menanggulangi gangguan-gangguan terhadap sistem apabila terlanjur terjadi kesalahan maka dapat langsung diantisipasi atau diatasi.

2.1.5. Basis Data

Menurut Lubis (2016:3) mendefinisikan bahwa “Basis data adalah tempat berkumpulnya data yang saling berhubungan dalam suatu wadah (organisasi/perusahaan) bertujuan agar dapat mempermudah dan mempercepat untuk pemanggilan atau pemanfaatan kembali data tersebut”. Sedangkan menurut Yanto (2016:11) mengemukakan bahwa “Basis data merupakan kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi), untuk memenuhi berbagai kebutuhan”.

Maka dari itu bahwa basis data merupakan suatu wadah yang menampung data-data yang saling berhubungan, dapat digunakan kembali, manipulasi, tanpa pengulangan untuk memenuhi berbagai kebutuhan pengguna data.

Komponen dasar sistem basis data digunakan untuk membantu kelancaran dari pembuatan dan manajemen basis data (Lubis, 2016:7), yang terdiri dari:

1. Data

Data pada sistem data mempunyai dua (2) ciri, yaitu data yang tersimpan secara terintegrasi (*integrated*) dan data dapat dipakai bersama-sama (*shared*).

- a. *Integrated* yaitu kumpulan dari berbagai macam *file* dari aplikasi-aplikasi yang berbeda yang disusun dengan cara menghilangkan bagian-bagian yang rangkap (*redundant*).
- b. *Shared* yaitu masing-masing bagian dari *database* dapat diakses oleh pemakai dalam waktu yang bersamaan, untuk aplikasi yang berbeda.

2. Perangkat keras

Perangkat keras ini terdiri dari semua peralatan perangkat keras komputer yang digunakan untuk pengelolaan sistem *database*, antara lain:

- a. Peralatan untuk penyimpanan, *disk*, *drum*, dan lain-lain.
- b. Peralatan *input* dan *output*.
- c. Peralatan komunikasi data.

3. Perangkat lunak

Perangkat lunak berfungsi sebagai perantara (*interface*) antara pemakai dengan data fisik *database*, dapat berupa *database management system* (DBMS) atau program-program aplikasi dan prosedur-prosedur.

4. Pemakai

Pemakai ini terbagi menjadi dua (2) bagian, yaitu:

- a. *Programmer*, orang/*team* yang membuat program aplikasi yang mengakses *database* dengan menggunakan bahasa pemrograman.

- b. *End user*, orang yang mengakses *database* melalui terminal dengan menggunakan *query language* atau program aplikasi yang dibuat oleh *programmer*.

Penggunaan basis data ini memiliki beberapa keuntungan (Lubis, 2016:8), diantaranya:

1. Terkontrolnya kerangkapan data dan inkonsistensi.
2. Terpeliharanya keselarasan data.
3. Data dapat dipakai secara bersama-sama.
4. Memudahkan penerapan standarisasi.
5. Memudahkan penerapan batasan-batasan penggunaan.
6. Terpeliharanya integritas data.
7. Terpeliharanya keseimbangan atas perbedaan kebutuhan data dari setiap aplikasi.
8. Program/data *independent*.

Basis data juga memiliki beberapa kerugian dalam penggunaannya. Adapun kerugian basis data (Lubis, 2016:8), yaitu:

1. Mahal dalam implementasinya.
2. Rumit/kompleks.
3. Penanganan proses *recovery & back up* sulit.
4. Kerusakan pada sistem basis data dapat mempengaruhi departemen terkait.

2.1.6. Website

Media informasi khususnya *website* dan *internet* merupakan hal yang tidak asing lagi karena merupakan bagian teknologi di kalangan masyarakat, banyak pula aplikasi berbasis jaringan (*web-based application*). *Website* merupakan salah satu

media pemasaran yang cukup menjanjikan. Situs *web* yang menarik dan informatif dapat dibuat dengan HTML dan PHP (Anna, 2016).

Menurut Ginanjar (2014:5) mengemukakan bahwa “*Website* adalah rangkaian atau sejumlah halaman di *internet* yang memiliki topik saling terkait untuk mempresentasikan suatu informasi”. Sedangkan menurut Yuhefizar (2013:2), mendefinisikan bahwa “*Website* adalah keseluruhan halaman-halaman *web* yang terdapat dari sebuah domain yang mengandung informasi”.

Dapat disimpulkan bahwa *website* merupakan rangkuman dari keseluruhan halaman-halaman *web* yang ada pada sebuah domain yang mengandung informasi teks, gambar diam atau gerak, animasi, suara, yang bersifat dinamis atau statis yang membentuk suatu rangkaian bangunan yang saling terkait dan memerlukan *internet*.

A. *Web Server*

Web server merupakan sebuah perangkat lunak *server* yang berfungsi menerima permintaan dari klien yang dikenal dengan *browser web* dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman *web* melalui protokol HTTP atau HTTPS dan bertugas mengelola halaman-halaman *web* dan dokumen-dokumen lainnya (Solichin, 2016:6).

Menurut Supono & Putratama (2016:6) mengemukakan bahwa: Paket *web server* adalah sebuah perangkat lunak *server* yang berfungsi untuk menerima permintaan dalam bentuk situs *web* melalui HTTP atau HTTPS dari klien itu, yang dikenal sebagai *browser web* dan mengirimkan kembali (reaksi) hasil dalam bentuk situs yang biasanya merupakan dokumen HTML.

Berdasarkan pendapat para ahli di atas, dapat disimpulkan bahwa *web server* merupakan perangkat lunak yang mengandung bermacam *protocol web* untuk dapat menjalankan perintah dari *client*, *server internet* yang digunakan sebagai koneksi dan transfer data (HTML, asp, aspx, php, js, dan lain).

B. *Web Browser*

Menurut Solichin (2016:9) “peramban *web* atau lebih dikenal dengan *web browser* merupakan perangkat lunak yang berfungsi untuk menerima dan menyajikan sumber informasi di *internet*”.

Web browser merupakan perangkat lunak yang dapat memproses paket HTTP dan menampilkannya kembali kepada *user* dengan format HTML (Supono & Putratama, 2016:5).

Berdasarkan pengertian diatas, dapat disimpulkan bahwa *web browser* adalah aplikasi yang digunakan untuk menampilkan halaman *web* untu proses pengolahan informasi, pengambilan dan penyajian informasi pada *website*.

2.1.7. Bahasa Pemrograman

Bahasa pemrograman berbasis *web* yang digunakan terdiri *hypertext preprocessor* (PHP), *hypertext markup language* (HTML), *cascading style sheet* (CSS), Javascript, JQuery dan Bootstraps. Pembahasan dari bahasa pemrograman yang digunakan diuraikan sebagai berikut:

A. *Hypertext Preprocessor* (PHP)

PHP merupakan sebuah bahasa pemrograman yang bersifat *open source*, yaitu pengguna dapat mengembangkan kode-kode fungsi PHP sesuai dengan kebutuhan (Maulana, 2014). Sedangkan Menurut Supono & Putratama (2016:3) “PHP (PHP: *Hypertext Preprocessor*) adalah suatu bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang berbasis *server-side* yang dapat ditambahkan ke dalam HTML”.

Berdasarkan pendapat dari para ahli di atas, dapat disimpulkan bahwa PHP merupakan bahasa pemrograman yang dapat mengolah *database*, *content website* sehingga *website* yang dibuat merupakan *web* dinamis.

B. Hypertext Markup Language (HTML)

Menurut Solichin (2016:10) “HTML merupakan bahasa pemrograman *web* yang memberitahukan peramban *web* (*web browser*) bagaimana menyusun dan menyajikan konten di halaman *web*”.

Hypertext markup language (HTML) adalah bahasa *markup* standar untuk membuat dan menggambarkan struktur halaman *web* yang terdiri dari elemen HTML sebagai blok bangunan sebuah halaman dan direpresentasikan dengan *tag* (Atmoko, 2018:1).

Berdasarkan definisi dari para ahli yang telah dikemukakan di atas, dapat disimpulkan bahwa *hypertext markup language* (HTML) merupakan bahasa standar yang digunakan untuk menyusun dan menyebarkan informasi serta menampilkan halaman *web*.

C. Cascading Style Sheet (CSS)

Cascading style sheet (CSS) merupakan bahasa pemrograman yang berfungsi untuk mempercantik tampilan *web* (Solichin, 2016:10).

Menurut Prasetio (2014:252) menyatakan bahwa “CSS adalah suatu teknologi yang digunakan untuk memperindah tampilan halaman *website* (situs)”.

Berdasarkan pengertian diatas, maka dapat disimpulkan bahwa *cascading style sheets* (CSS) merupakan bahasa yang digunakan dan membantu *programmers* dalam merancang sebuah tampilan *website* dan menimbulkan efek animasi yang bagus.

D. *Javascript*

Javascript dikembangkan oleh Netscape dengan nama awal LiveScript yang berfokus pada proses pengolahan data di sisi *client* dan menyajikan komponen *web* yang lebih interaktif serta berfungsi untuk menambah fungsionalitas dan kenyamanan halaman *web* (Solichin, 2016:11).

Menurut Suryana & Koesheryatin (2014:181) “Javascript adalah bahasa *script* berdasar pada objek yang memperbolehkan pemakai untuk mengendalikan banyak aspek interaksi pemakai pada suatu dokumen HTML dimana. Dimana objek tersebut dapat berupa suatu *window*, *frame*, URL, dokumen, *form*, *button*, atau *item* yang lain”.

Dapat disimpulkan bahwa *javascript* merupakan bahasa pemrograman yang berbasis *client* dan *script* untuk tampilan pendukung pada *website* sehingga membuat halaman menjadi lebih menarik dan interaktif.

E. *JQuery*

Menurut Prasetya (2013:85) “*jQuery* merupakan sebuah *framework (library)* *JavaScript* yang ditujukan untuk membantu Anda dalam menghasilkan aplikasi-aplikasi *cross-platform* yang responsif dan mudah”.

jQuery merupakan sebuah *plugin* yang berfungsi sebagai *framework JavaScript* yang sering digunakan dalam pengembangan *web* (Winarno, 2014:52).

Dapat ditarik sebuah kesimpulan *jQuery* merupakan pustaka *javascript* yang memudahkan bagi para *programmer* dalam penulisan kode *javascript*.

F. *Bootstrap*

Menurut Subagia (2018:45) “*bootstrap* adalah *template* desain *web* dengan fitur plus (*framework CSS*). *Bootstrap* diciptakan untuk mempermudah proses

desain *web* bagi berbagai tingkat pengguna, mulai dari level pemula hingga yang sudah berpengalaman”.

Sedangkan menurut Rozi (2016:225) “*bootstrap* adalah paket aplikasi siap pakai untuk membuat *template web*”.

Berdasarkan pendapat dari para ahli di atas, dapat disimpulkan bahwa *bootstrap* merupakan paket aplikasi yang menyediakan *template* dengan format *cascading style sheet* (CSS) untuk mendesain *web* agar tampilan *web* menjadi lebih menarik.

2.1.8. Model Prototipe

Model prototipe digunakan untuk merancang sistem informasi. Model *prototype* memberikan kesempatan untuk pengembang program dan objek penelitian untuk saling berinteraksi selama proses perancangan sistem (Sukanto & Shalahuddin, 2015:33). Sedangkan menurut Yurindra (2017:47) model *prototype* adalah “suatu proses yang memungkinkan *developer* membuat sebuah model *software*, metode ini baik digunakan apabila *client* tidak bisa memberikan informasi yang maksimal mengenai kebutuhan yang diinginkannya”.

Maka dari itu, dapat disimpulkan bahwa model prototipe merupakan salah satu model pengembangan perangkat lunak dimana pengembang program dan objek penelitian dapat saling berkomunikasi dan memberikan informasi yang terdiri dari mendengarkan pelanggan atau analisa kebutuhan, membuat rancangan (*mock-up*) dan pengujian rancangan).

Model *prototype* ini memiliki beberapa tahapan (Sukanto & Shalahuddin, 2015:32), yaitu:

1. Mendengarkan Pelanggan

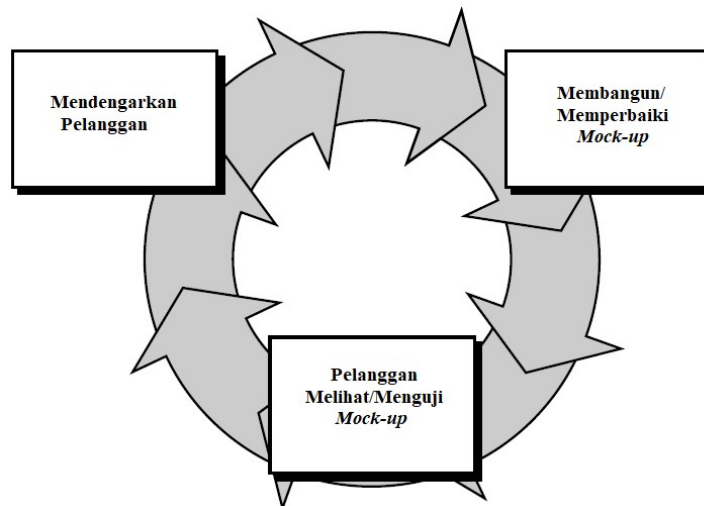
Pengembang program dan objek penelitian bertemu dan menentukan tujuan umum dan kebutuhan dasar. Detail kebutuhan mungkin pada awal pengumpulan kebutuhan.

2. Membangun atau Memperbaiki *Mock-Up*

Perancangan sistem dapat dikerjakan apabila data-data yang berkaitan telah dikumpulkan selama pengumpulan kebutuhan. Rancangan ini menjadi dasar pembuatan *prototype*. Pembuatan *prototype* ini merupakan tahapan perealisasi rancangan *prototype* menggunakan bahasa pemrograman.

3. Pelanggan Melihat dan Menguji *Mock-Up*

Objek penelitian mengevaluasi *prototype* yang dibuat dan dipergunakan untuk memperjelas kebutuhan *software*.



Sumber: (Sukamto & Shalahuddin, 2015:32)

Gambar II.1. Ilustrasi Model *Prototype*

2.2. Teori Pendukung

Teknik atau alat bantu digunakan dalam memvisualisasikan rancangan sistem. Teori-teori lain yang digunakan untuk mendukung penulisan ini terdiri dari *unified modeling language* (UML) yang terdiri dari *use case diagram* dan *activity diagram*, *entity relationship diagram* (ERD) dan *black box testing*.

2.2.1. *Unified Modeling Language* (UML)

UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem (Hendini, 2016). Sedangkan Menurut Sukamto & Shalahuddin (2015:137) mendefinisikan bahwa “UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung”.

Maka dari itu dapat ditarik kesimpulan bahwa *unified modeling language* (UML) merupakan suatu bahasa standar yang digunakan untuk pemodelan dan komunikasi rancangan perangkat lunak dengan menggunakan diagram atau simbol-simbol tertentu.




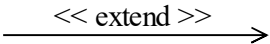
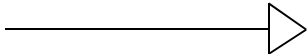
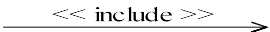
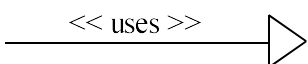
A. *Use Case Diagram*

Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat”. *Use case diagram* digunakan untuk mendeskripsikan tipikal interaksi antara pengguna dengan sistem informasi (Maulana, 2014). Sedangkan menurut Sukamto & Shalahuddin (2015:155) mendefinisikan bahwa “*Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat”.

Berdasarkan beberapa pendapat ahli di atas, dapat disimpulkan bahwa *use case diagram* merupakan diagram UML yang berfungsi sebagai alat bantu

pemodelan untuk menggambarkan tingkah laku (*behavior*) dari sudut pandang luar sistem untuk menjelaskan interaksi dan peran antara aktor dengan sistem yang dirancang.

Tabel II.1.
Simbol Use Case Diagram

Simbol	Deskripsi
<p><i>use case</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan kata kerja di awal frase nama <i>use case</i> .
<p>Aktor/<i>actor</i></p> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
<p>Asosiasi/<i>association</i></p> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
<p>Ektensi/<i>extend</i></p> 	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.
<p>Generalisasi/<i>generalization</i></p> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya.
<p>Menggunakan/<i>include/uses</i></p>  	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.


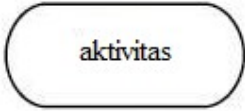
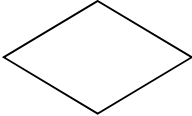


Sumber: (Sukanto & Shalahuddin, 2015:156)

B. Activity Diagram

Activity diagram merupakan diagram yang menerangkan tentang aktifitas-aktifitas yang dapat dilakukan oleh seorang *entity* atau pengguna yang akan diterapkan pada aplikasi (Meilinda, 2016). Sedangkan menurut Sukamto & Shalahuddin (2015:161) mendefinisikan bahwa “Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”

Dapat disimpulkan bahwa *activity diagram* merupakan diagram yang menggambarkan aktifitas-aktifitas sistem dimana setiap urutan aktifitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.

Tabel II.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.

<div>Swimlane</div> <div><div>nama swimlane</div></div>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
---	--

Sumber: (Sukamto & Shalahuddin, 2015:162)




C. Sequence Diagram


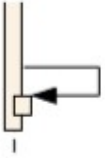


Sequence diagram merupakan UML yang menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu (Meilinda, 2016).

Sukamto & Shalahuddin (2015:165) mengemukakan bahwa:
Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima oleh 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.

Dapat disimpulkan bahwa *sequence diagram* dapat diartikan sebagai alat pemodelan rancangan sistem yang menggambarkan alur atau urutan sistem yang bersinkronisasi dengan *use case diagram* untuk mendeskripsikan waktu hidup objek dan *message* yang dikirm atau diterima oleh objek tersebut.

Tabel II.3.
Simbol Sequence Diagram

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interfaces</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>form entry</i> dan <i>form cetak</i> .
	<i>Control Class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas,

	contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

Sumber: (Hendini, 2016)

D. Class Diagram

Class diagram merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem (Hendini, 2016). Sedangkan menurut Sukanto & Shalahuddin (2015:141), “diagram kelas atau *class diagram* menggambarkan struktur sistem dari segini pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”.

Maka dari itu, dapat disimpulkan bahwa *class diagram* merupakan diagram UML yang mirip dengan ERD atau LRS yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas dan memperlihatkan aturan serta tanggung jawab entitas serta berisikan metode-metode atau aturan tertentu.

Tabel II.4.
Multiplicity Class Diagram

<i>Multiplicity</i>	<i>Keterangan</i>
1	Satu dan hanya satu.
0..*	Boleh tidak ada atau 1 atau lebih.
1..*	1 atau lebih.
0..1	Boleh tidak ada, maksimal 1.
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimal 4.

Sumber: (Hendini, 2016)

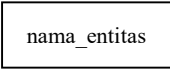
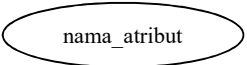
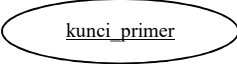
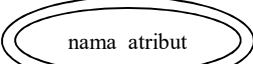

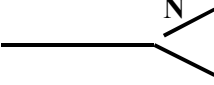
2.2.2. *Entity Relationship Diagram (ERD)*

Menurut Sukamto & Shalahuddin (2015:53) mendefinisikan bahwa “ERD adalah bentuk paling awal dalam melakukan perancangan basis data relasional. Jika menggunakan OODMBS maka perancangan ERD tidak perlu dilakukan”. *Logical record structure (LRS)* merupakan representasi dari struktur *record-record* pada tabel dimana tabel-tabel tersebut terbentuk dari hasil himpunan antar entitas pada *entity relationship diagram* yang telah ditransformasikan menjadi bentuk LRS (Pratama, Sihombing & Putra, 2014).

Dapat disimpulkan bahwa *entity relationship diagram (ERD)* merupakan teknik pemodelan struktur data secara konseptual yang menggambarkan entitas lengkap dengan atributnya dan hubungan yang terjadi antar entitas tersebut.

Entity relationship diagram (ERD) digambarkan dengan simbol-simbol yang saling berhubungan (Sukamto & Shalahuddin, 2015:50) yang diuraikan sebagai berikut:

Tabel II.5.
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>multivalu</i> <i>e</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, diawali dengan kata kerja.
	Asosiasi/ <i>associatio</i> <i>n</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: (Sukamto & Shalahuddin, 2015:50)

2.2.3. *Logical Record Structure (LRS)*

LRS merupakan hasil transformasi diagram E-R (ERD) menggunakan aturan aturan tertentu. (Ladjamudin, 2013:159). LRS adalah representasi dari struktur *record-record* pada tabel. Dimana tabel-tabel tersebut terbentuk dari hasil himpunan antar entitas pada ERD Pratama, Sihombing & Putra, 2014).

Dapat disimpulkan bahwa LRS merupakan diagram yang menggambarkan hasil kardinalitas dari ERD menjadi LRS yang memiliki hubungan dan mengikuti pola pemodelan sesuai dengan aturan pengkonversian.

Aturan pokok yang telah diuraikan (Ladjamudin, 2013:159) mempengaruhi langkah pentransformasian yaitu kardinalitas. Adapun kardinalitas tersebut (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.