

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Dalam bab ini diperlukan teori-teori yang dapat mendukung kemudahan dalam mempelajari serta merancang sistem aplikasi yang diharapkan dapat digunakan seoptimal mungkin. Dengan demikian akan sangat membantu untuk mempermudah proses *inventory asset* pada SMK Bina Karya 1 Karawang

A. Model Pembelajaran Berbasis Web

Menurut Khosim (2017:5) Metode pembelajaran dapat diartikan “Cara yang digunakan untuk mengimplementasikan rencana yang sudah disusun dalam bentuk kegiatan nyata dan praktis untuk mencapai tujuan pembelajaran”. Istilah model pembelajaran mempunyai 4 ciri khusus yang tidak dimiliki oleh strategi atau metode pembelajaran:

- 1) Rasional teoritis yang logis yang disusun oleh pendidik.
- 2) Tujuan pembelajaran yang akan dicapai.
- 3) Langkah-langkah mengajar yang diperlukan agar model pembelajaran dapat dilaksanakan secara optimal.
- 4) Lingkungan belajar yang diperlukan agar tujuan pembelajaran dapat dicapai.

Menurut Sadiman dalam Jalinus dan Ambiyar (2016:5) menyampaikan fungsi media (media pendidikan) secara umum, sebagai berikut:

- 1) Memperjelas penyajian pesan agar tidak terlalu bersifat visual.
- 2) Mengatasi keterbatasan ruang, waktu, dan daya indra.

- 3) Meningkatkan kegairahan belajar, memungkinkan siswa belajar sendiri berdasarkan minat dan kemampuannya, dan mengatasi sikap pasif siswa.
- 4) Memberikan rangsangan yang sama, dapat menyamakan pengalaman dan persepsi siswa terhadap isi pembelajaran.

Menurut Rusman dalam Januarisman dan Ghufroon (2016:171) mengatakan pembelajaran berbasis *web* merupakan “Suatu kegiatan pembelajaran yang memanfaatkan media situs (*website*) yang bisa diakses melalui jaringan *internet*”. Pembelajaran berbasis *web* atau yang dikenal dengan “*web based learning*” merupakan salah satu penerapan dari pembelajaran elektronik (*E-Learning*).

Banyak kelebihan atau keuntungan yang diberikan oleh aplikasi berbasis *web* dari pada aplikasi berbasis *desktop*, sehingga aplikasi berbasis *web* banyak digunakan oleh perusahaan sebagai bagian dari strategi teknologi informasinya.

Menurut Rarung (2018:2) kelebihan *website* yakni:

- 1) Akses informasi mudah.
- 2) Informasi mudah didistribusikan.
- 3) Bebas *platform*, informasi dapat disajikan oleh *browser web* pada sistem operasi mana saja karena adanya standar dokumen berbagai tipe data dapat disajikan.

Sedangkan kelebihan model pembelajaran berbasis *web* menurut Rusman dalam Januarisman dan Ghufroon (2016b:172), yaitu:

- 1) Memungkinkan setiap orang dimanapun, kapanpun, untuk mempelajari apapun.
- 2) Pebelajar dapat belajar sesuai dengan karakteristik karena pembelajaran berbasis *web* membuat pembelajaran menjadi sifat *individu*.

- 3) Kemampuan untuk membuat tautan (*link*) sehingga pebelajar dapat mengakses informasi dari berbagai sumber, baik didalam maupun diluar lingkungan belajar.
- 4) Sangat potensial sebagai sumber belajar bagi pebelajar yang tidak memiliki cukup waktu untuk belajar.
- 5) Dapat mendorong pebelajar untuk lebih aktif dan mandiri didalam belajar.
- 6) Menyediakan sumber belajar tambahan yang dapat digunakan untuk memperkaya materi pembelajaran.
- 7) Menyediakan mesin pencari yang dapat digunakan untuk mencari informasi yang mereka butuhkan.
- 8) Isi dari materi pelajaran dapat di*up-date* dengan mudah.

B. Sistem

1. Pengertian Sistem dan Sistem Informasi

Menurut McLeod dalam Darmawan dan Fauzi (2013:4) “Sistem adalah sekelompok elemen-elemen yang terintegrasi dengan tujuan yang sama untuk mencapai tujuan”. Sedangkan pengertian sistem Menurut Anggraeni dan Irviani (2017:1) “Sistem adalah kumpulan orang yang saling bekerja sama dengan ketentuan-ketentuan aturan yang sistematis dan terstruktur untuk membentuk satu kesatuan yang melaksanakan suatu fungsi untuk mencapai tujuan”. Secara sederhana, suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain dan terpadu. Dari definisi ini dapat dirinci lebih lanjut pengertian sistem secara umum, yaitu:

- a) Setiap sistem terdiri dari unsur-unsur.
- b) Unsur-unsur tersebut merupakan bagian terpadu sistem yang bersangkutan.
- c) Unsur sistem tersebut bekerja sama untuk mencapai tujuan sistem.
- d) Suatu sistem merupakan bagian dari sistem lain yang lebih besar.

Pengertian sistem informasi menurut Hutahaean (2014:13) “Suatu sistem didalam 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 Punjul (2016:1) “Sistem informasi adalah sistem fisik dan sosial yang ditata sedemikian rupa untuk mencapai tujuan tertentu”.

Sedangkan Menurut Anggraeni dan Irviani (2017:2) “Sistem informasi merupakan suatu kombinasi teratur dari orang-orang, *hardware*, *software*, jaringan komunikasi dan sumber daya data yang mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi”.

2. Sistem Berorientasi Objek

Menurut Abdulloh (2017:1) OOP (*Object Oriented Programming*) merupakan “Teknik pemrograman dengan menggunakan konsep objek”. Sedangkan menurut Rosa 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 terhadapnya”.

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, perancangan berorientasi objek, pemrograman berorientasi objek dan pengujian berorientasi objek. Keuntungan menggunakan metodologi berorientasi objek sebagai berikut:

a) Meningkatkan Produktivitas

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

b) Kecepatan Pengembangan

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

c) Kemudahan Pemeliharaan

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

d) Adanya Konsistensi

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

e) 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.

Saat ini sudah banyak bahasa pemrograman berorientasi objek. Banyak orang berfikir bahwa pemrograman berorientasi objek identik dengan bahasa pemrograman *Java*. Namun sebenarnya bahasa pemrograman yang mendukung pemrograman berorientasi objek tidak hanya bahasa *Java* salah satunya yakni: Bahasa Pemrograman (*Web*) PHP, PHP4 dengan versi-versi akhir menuju PHP5 sudah mendukung pemrograman berorientasi objek. PHP merupakan bahasa pemrograman yang digunakan untuk pemrograman *web*.

3. Pemesanan

Menurut Kotler dalam Puspita dkk (2013:1) Pemesanan adalah “Kegiatan transaksi yang menyatakan keinginan untuk memiliki atau membeli barang dan jasa tersebut sebelum melakukan transaksi finansial berupa pembayaran finansial terhadap produk atau jasa yang diinginkan”.

Sedangkan menurut Wahyudi dan Astuti (2013:8) mengatakan bahwa: Pemesanan adalah “Sebuah proses perjanjian berupa pemesanan sebuah produk baik barang maupun jasa dimana pada saat itu telah terdapat kesepakatan antara konsumen dengan produsen mengenai produk tersebut namun belum ditutup oleh sebuah transaksi jual-beli”.

C. Website

Menurut Irwansyah dan Moniaga (2014:25) *Internet (Interconnection Networking)* adalah “Seluruh jaringan komputer yang saling terhubung menggunakan standar sistem global *Transmission Control Protocol/ Internet Protocol Suite* (TCP/IP) sebagai protokol pertukaran paket (*Packet Switching Communication Protocol*) untuk melayani miliaran pengguna diseluruh dunia”.

Sedangkan menurut Abdulloh (2016:1) “*Internet* adalah jaringan yang digunakan untuk mengirim informasi pada *website*”.

Pengertian *website* menurut Irwansyah dan Moniaga (2014:34) “*World Wide Web (WWW)* atau yang biasa disebut *website* merupakan salah satu layanan yang didapat oleh pemakai komputer yang terhubung ke *internet*”. Sedangkan menurut Abdulloh (2016:1) “*Website* dapat diartikan sekumpulan halaman yang terdiri atas beberapa laman yang berisi informasi dalam bentuk data digital, baik berupa teks, gambar, *video*, *audio*, dan animasi lainnya yang disediakan melalui jalur koneksi *internet*”. Lebih jelasnya, *website* merupakan halaman-halaman yang berisi informasi yang ditampilkan oleh *browser*. Berikut beberapa istilah seputar *website*, yaitu:

1) Penjelajahan *Web (Web Browser)*

Untuk membuka *web page*, kita memerlukan sebuah *web browser* untuk membukanya. Saat ini banyak *web browser* yang dapat digunakan untuk yang paling sering digunakan diantaranya *Mozilla Firefox*, *Google Chrome*, *Internet Explore*, *Opera*, dan *Safari*.

2) *Web Server*

Menurut Solichin (2016:1) *Web server* merupakan:

Sebuah perangkat lunak dalam *server* yang berfungsi menerima permintaan (*request*) berupa halaman *web* melalui HTTP atau HTTPS dari klien yang dikenal dengan *browser web* dan mengirimkan kembali (*response*) hasilnya dalam bentuk halaman-halaman *web* yang umumnya berbentuk dokumen HTML.

Web server yang dimaksud disini adalah simulasi dari sebuah *web server* secara fisik. *Web server* biasanya juga disebut *HTTP server* karena menggunakan *protocol* HTTP sebagai basisnya. Beberapa *web server* yang sering digunakan diantaranya adalah *PWS*, *IIS*, *Apache* dan sebagainya.

D. Basis Data (*Database*)

Menurut Rosa dan Shalahuddin (2015:43) “Sistem 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. Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apapun bentuknya, entah berupa *file* teks ataupun *Database Management System* (DBMS). Kebutuhan basis data dalam sistem informasi meliputi:

- 1) Memasukkan, menyimpan, dan mengambil data.
- 2) Membuat laporan berdasarkan data yang telah disimpan.

Saat ini, basis data yang digunakan diberbagai kalangan untuk melakukan penyimpanan dan pengolahan data, mulai dari kalangan akademis sampai ke industri besar dan kecil yaitu *MySQL*.

Menurut Winarto dan Zaki (2013:120) *MySQL* merupakan “ Tipe data relasional yang artinya *MySQL* menyimpan datanya dalam bentuk tabel-tabel yang saling berhubungan”. Sedangkan menurut Kadir dalam Taufik dan Ermawati (2017:2) “*MySQL* merupakan *software* yang tergolong *database server* dan bersifat *Open Source*”.

Menurut Rosa dan Shalahuddin (2015:2) Perangkat lunak (*Software*) adalah “Program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*)”.Perangkat lunak dan bahasa pemrograman yang mendukung penggunaan *websitedan database*, yaitu:

1) *Xampp*

Menurut Wicaksono dalam Firdyanthie dan Mahdiati (2016:129) menjelaskan bahwa “*Xampp* adalah sebuah *software* berbasis PHP dan menggunakan pengolah data *MySQL* di komputer lokal”. *Xampp* berperan sebagai *server web* pada komputer lokal. *Apache* merupakan aplikasi *web server*, dan bertugas untuk menghasilkan halaman *web* yang benar kepada *user* berdasarkan kode PHP yang dituliskan oleh pembuat halaman *web*. Sedangkan *MySQL* merupakan aplikasi *database server* yang digunakan untuk mengolah *database* beserta isinya.

2) Bahasa Pemrograman

a) HTML (*Hyper Text Markup Language*)

“Skrip yang berupa tag-tag untuk membuat dan mengatur struktur *website*” (Abdulloh, 2016:2). HTML memiliki beberapa tugas utama dalam membangun *website*, diantaranya:

- 1) Menentukan *layout website*.
- 2) Memformat teks dasar, seperti paragraf dan format *font*.
- 3) Membuat *list* dan membuat tabel.
- 4) Menyisipkan gambar, *video*, dan *audio*.
- 5) Membuat *link* dan membuat formulir.

b) CSS (*Cascading Style Sheets*)

“Skrip yang digunakan untuk mengatur desain *website*” (Abdulloh, 2016:2). Fungsi CSS adalah memberikan pengaturan yang lebih lengkap agar struktur *website* yang dibuat dengan HTML terlihat lebih rapih dan elegan.

c) PHP (*Hypertext Preprocessor*)

Menurut Abdulloh (2016:3) “PHP merupakan *server-side programming*, yaitu: bahasa pemrograman yang diproses disisi *server*”. Fungsi utama PHP dalam membangun *website* adalah untuk melakukan pengolahan data pada *database*. Data *website* akan dimasukkan ke *database*, diedit, dihapus dan ditampilkan pada *website* yang diatur oleh PHP.

d) *Javascript*

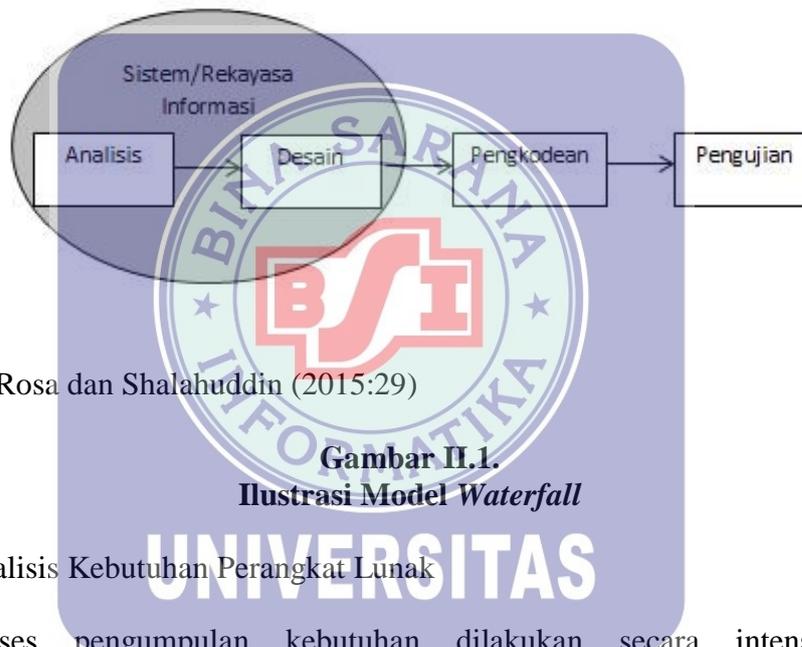
Menurut Abdulloh (2016:3) mengemukakan peran *javascript* dalam membuat *website* adalah “Memberikan efek animasi yang menarik dan interaktivitas dalam penanganan *event* yang dilakukan oleh pengguna *website*”.

e) *jQuery*

Menurut Hidayatullah dan Kawistara (2014:421) “*jQuery* adalah suatu *library Javascript* yang akan menjadikan *web* lebih bagus dalam hal *user interface*, lebih stabil, dan dapat mempercepat waktu kinerja dalam pembuatan *web* hanya perlu memanggil fungsinya saja tanpa harus membuat dari awal”. Banyak *website* yang memanfaatkan *library* ini untuk menyederhanakan fungsi-fungsi yang ada di *javascript* atau *Ajax*, sehingga dapat menghemat *coding* program, yang sebelumnya menggunakan *javascript* beberapa baris kode, namun dengan *jQuery* hanya satu baris saja.

E. Model Pengembangan Perangkat Lunak

Metode yang digunakan pada pengembangan perangkat lunak ini menggunakan model *Waterfall*. Menurut Rosa dan Shalahuddin (2015:28) “Model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*)”. Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung (*support*). Berikut gambar dari model air terjun (*waterfall*):



Sumber: Rosa dan Shalahuddin (2015:29)

Gambar II.1.
Ilustrasi Model Waterfall

1) Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk mengspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2) Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean. Tahap

ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3) Pembuatan Kode program

Desain harus ditranslasikan kedalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang dibuat pada tahap desain.

4) Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional serta memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5) Pendukung (*Support*) atau Pemeliharaan (*Maintenance*)

Perubahan terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak yang baru.

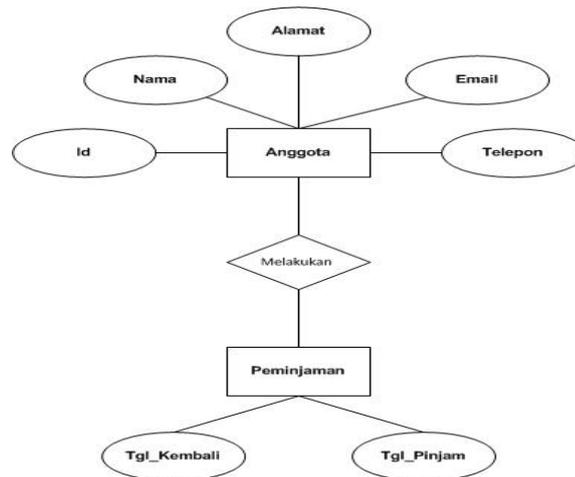
2.2. Teori Pendukung

Teori pendukung adalah pembahasan yang digunakan untuk mendapatkan logika model dari suatu sistem yang menggunakan simbol-simbol, lambang-lambang, diagram-diagram, yang menunjukkan secara tepat arti dan fungsinya. Teori pendukung (*Tools System*) yang akan digunakan dalam merancang suatu sistem informasi meliputi ERD (*Entity Relational Diagram*), Derajat Kardinalitas, LRS (*Logical Record Structure*), Diagram UML (*Unified Modeling Language*) yang meliputi *Activity Diagram*, *Use Case Diagram*, *Class Diagram*, dan *Sequence Diagram* diantaranya:

A. *Entity Relationship Diagram* (ERD)

ERD merupakan salah satu permodelan data konseptual yang paling sering digunakan dalam proses pengembangan basis data bertipe relational. Penggunaannya yang sangat luas diakibatkan beberapa faktor, yaitu: kemudahan penggunaan secara luas *Computer Aided Software Engineering* (CASE), dukungan konsep matematika (kalkulus relasional) yang tangguh, hubungan entitas antar entitas merupakan konsep permodelan alamiah yang sesuai dengan keadaan dunia nyata.

Menurut Yanto (2016:32) ERD adalah “Suatu diagram untuk menggambarkan desain konseptual dari model konseptual suatu basis data”. Sedangkan menurut Rosa dan Shalahuddin (2015:53) “ERD adalah bentuk paling awal dalam melakukan perancangan basis data relational”.



Sumber: Rosa dan Shalahuddin (2015:58)

Gambar II.2.
Diagram ERD Studi Kasus Perpustakaan

ERD juga merupakan gambaran yang merelasikan antara objek yang satu dengan objek yang lain dari objek di dunia nyata yang sering dikenal dengan hubungan antar entitas.

1) Derajat Kardinalitas

Derajat kardinalitas merupakan penjabaran dari hubungan antarentitas.

Derajat kardinalitas dibagi atas 3 (tiga) bagian yaitu:

a) *One to One*

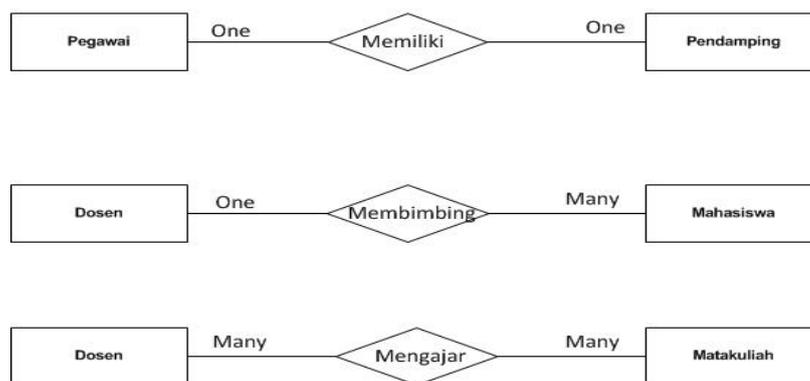
Derajat kardinalitas *one to one* terjadi jika satu entitas X hanya berelasi dengan satu entitas Y, ataupun sebaliknya.

b) *One to Many*

Derajat kardinalitas *one to many* terjadi jika satu entitas X berelasi dengan banyak entitas Y, ataupun sebaliknya.

c) *Many to Many*

Derajat Kardinalitas *many to many* terjadi jika banyak entitas X berelasi dengan banyak entitas Y, ataupun sebaliknya.



Sumber: Yanto (2016:40)

Gambar II.3.
Contoh Derajat Kardinalitas

2) LRS (*Logical Record Structure*)

Menurut Simarmata dalam Fridayanthie dan Mahdiati (2016:132) "*Logical Record Structure* (LRS) adalah representasi dari struktur *record-record* pada tabel-tabel yang terbentuk dari hasil relasi antar himpunan entitas". Menentukan kardinalitas, jumlah tabel, dan *Foreign Key* (FK). Berikut adalah cara membentuk skema *database* atau LRS (*Logical Record Structure*) berdasarkan *Entity Relationship Diagram* (ERD):

- a) Jika relasinya *one to one*, maka *foreign key* diletakkan pada salah satu dari dua entitas yang ada atau menyatukan kedua entitas tersebut.
- b) Jika relasinya *one to many*, maka *foreign key* diletakkan pada entitas *Many*.
- c) Jika relasinya *many to many*, maka dibuat *file connector* yang berisi dua *foreign key* yang berasal dari kedua entitas.

B. *Unified Modeling Language*(UML)

Menurut Mulyani (2017:35) UML adalah “Sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem”.

Menurut Rosa dan Shalahuddin (2015:140) diagram UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori, yaitu:

1) *Structure Diagrams*

Kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.

2) *Behavior Diagrams*

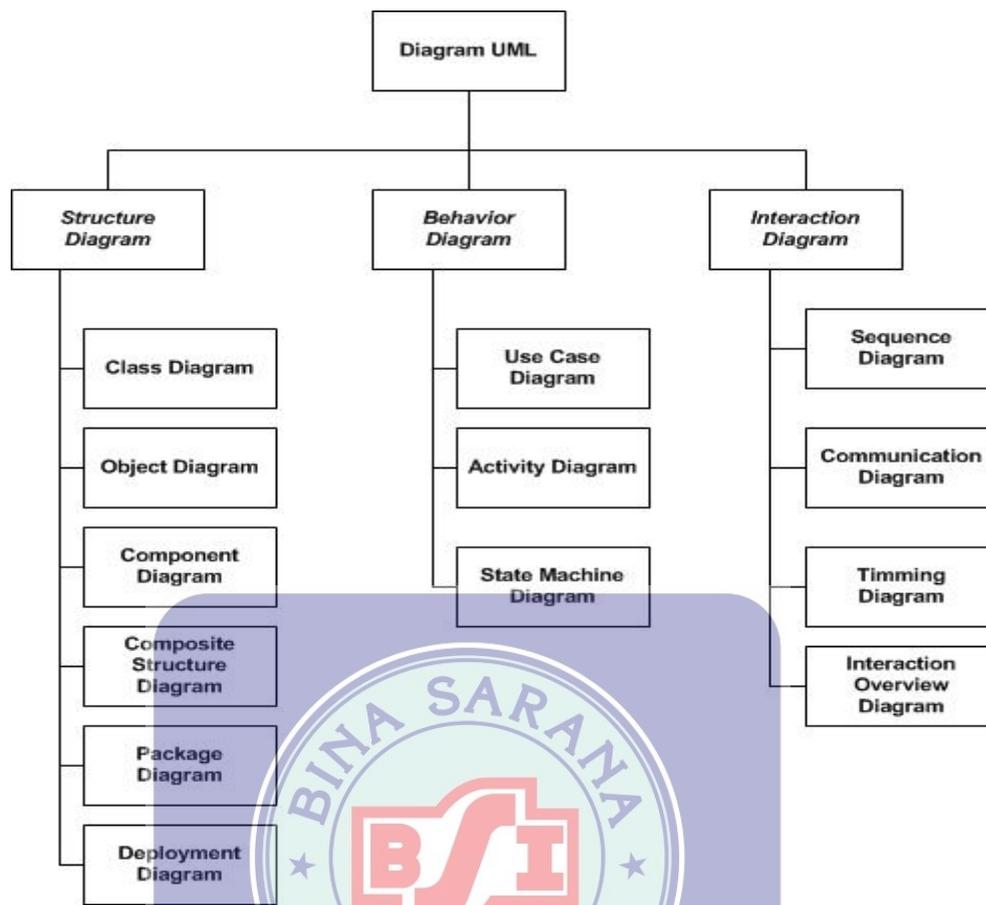
Kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

3) *Interaction Diagrams*

Kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.



UNIVERSITAS



Sumber: Rosa dan Shalahuddin (2015:140)

Gambar II.4.
Diagram UML

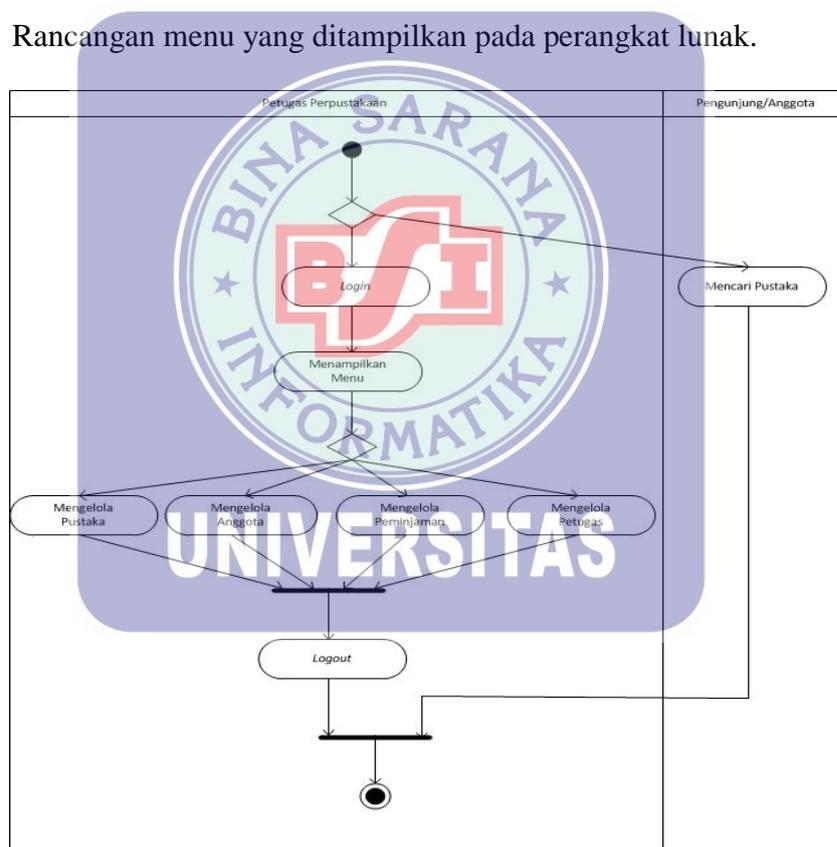
Dari penjelasan diatas penulis menggunakan 4 (empat) diagram yang digunakan sebagai acuan dalam penulisan Tugas Akhir ini, yaitu:

1) *Activity Diagram*

Menurut Rosa 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 adalah diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan oleh aktor, jadi aktivitas yang dapat dilakukan

oleh sistem. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

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



Sumber: Rosa dan Shalahuddin (2015:235)

Gambar II.5.
Diagram Activity Studi Kasus Perpustakaan

2) Use Case Diagram

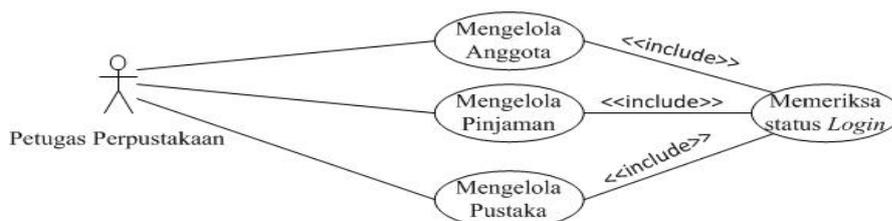
Menurut Rosa dan Shalahuddin (2015:155) “*Use case diagram* merupakan permodelan untuk kelakuan (*Behavior*) sistem informasi yang akan dibuat”. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel 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.



Sumber: Rosa dan Shalahuddin (2015:204)

Gambar. II.6.
Diagram Use Case Studi Kasus Perpustakaan

3) *Class Diagram*

Menurut Rosa dan Shalahuddin (2015:141) “Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”. Kelas memiliki apa yang disebut *attribut* dan *metode* atau operasi.

a) *Attribut* merupakan variabel-variabel yang dimiliki oleh suatu kelas.

b) Operasi atau *metode* adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak (*software*) atau *programmer* dapat membuat kelas-kelas di dalam program perangkat sesuai dengan perancangan diagram kelas. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

1) Kelas Main

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2) Kelas yang menangani tampilan sistem (*view*)

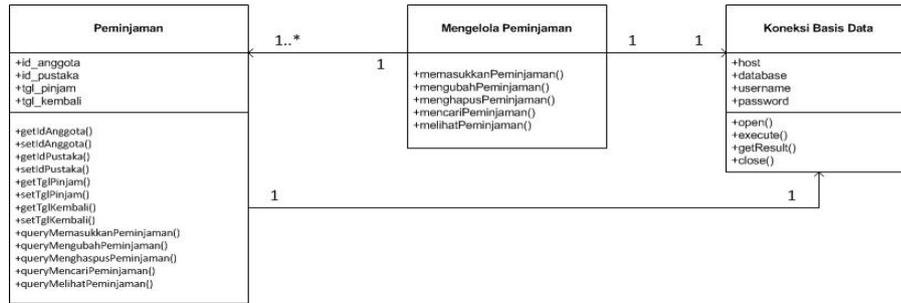
Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

3) Kelas yang diambil dari pendefinisian *use case* (*controller*)

Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*, kelas ini biasanya disebut dengan kelas proses menangani proses bisnis pada perangkat lunak.

4) Kelas yang diambil dari pendefinisian data (*model*)

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

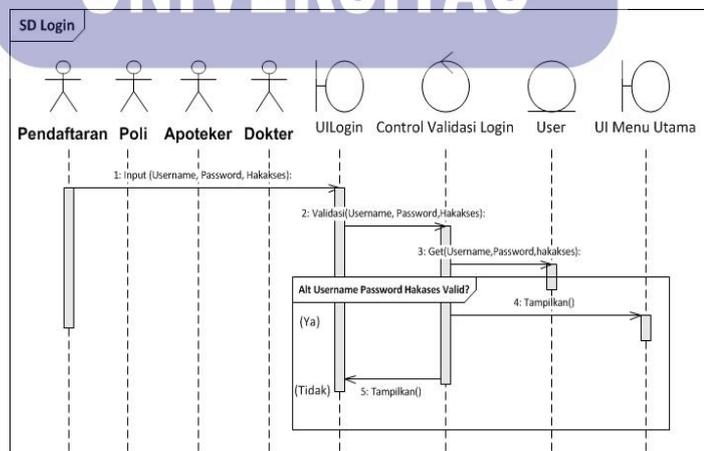


Sumber: Rosa dan Shalahuddin (2015:205)

Gambar II.7.
Diagram Class Studi Kasus Perpustakaan

4) *Sequence Diagram*

Menurut Rosa 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”. Sedangkan menurut Munawar dalam Putri Buani dan Suryani (2018:8) “*Sequence diagram* digunakan untuk menggambarkan perilaku pada sebuah *scenario*”. Diagram ini menunjukkan sebuah contoh objek dan *message* atau pesan yang diletakkan diantara objek-objek ini didalam *use case* diagram.



Sumber: Putri Buani dan Suryani (2018:14)

Gambar II.8.
Diagram Sequence Login