

BAB II

LANDASAN TEORI

Bab ini akan membahas mengenai tinjauan pustaka yang berisi tentang hasil penelitian yang pernah dilakukan, kerangka pemikiran, dan landasan teori yang membahas teori-teori dasar pendukung untuk penelitian ini.

2.1 Tinjauan Pustaka

Penelitian Safira (2014) dengan judul Katalog dan Katalogisasi membahas tentang Katalogisasi adalah proses pembuatan katalog. Secara luas kegiatan tersebut dapat dibagi kepada dua (2) macam yaitu *katalogisasi deskriptif* dan *katalogisasi subyek*. Katalogisasi deskriptif adalah kegiatan merekam dan mengidentifikasi data bibliografi, yakni data mengenai pengarang, judul, tempat terbit, penerbit, tahun terbit, edisi dan data buku lainnya yang diperlukan. Katalogisasi subyek ialah proses menentukan tajuk subyek dan nomor klasifikasi. Dalam hal terakhir ini prosesnya disebut juga klasifikasi.

Penelitian dari Prasetyo dan Dahlan (2015) dengan judul Pembuatan Aplikasi Pengolahan Data Transaksi Dan Penjualan Aksesoris Handphone Pada Toko Yoyo' Cell membahas tentang NetBeans merupakan *IDE (Integreted Development and Environment)* untuk membuat aplikasi java. Karena didukung dengan fasilitas *drag and drop* komponen yaitu *rapid application development* (pemrograman berbasis visual). *Software* yang dibuat oleh Sun Microsystem ini gratis digunakan. NetBeans mengacu pada dua hal, yaitu platform untuk pengembangan aplikasi desktop java, dan sebuah *Integreted Development and Environment (IDE)* yang dibangun dengan *platform NetBeans*.

Penelitian dari Wardoyo (2016) dengan judul Validasi Data Katalog Melalui Kegiatan *Stock Opname* Dan Pengadaan koleksi Bahan Pustaka Perpustakaan Soeman Hs Tahun Anggaran 2014 membahas tentang Katalog perpustakaan dapat diartikan sebagai suatu daftar yang sistematis dari bahan pustaka dalam perpustakaan, dengan informasi deskriptif tentang pengarang, judul, penerbit, tahun terbit, bentuk

fisik, subjek, ciri khas bahan dan tempatnya. Katalog juga mencatat informasi bibliografis dari suatu dokumen atau bahan pustaka. Katalog perpustakaan juga merupakan susunan yang sistematis dari seperangkat cantuman bibliografis yang mempresentasikan kumpulan dari suatu koleksi tertentu. Koleksi tersebut bisa terdiri dari berbagai jenis bahan pustaka seperti: buku, terbitan berkala, peta, rekaman suara, gambar, notasi music dan lainnya

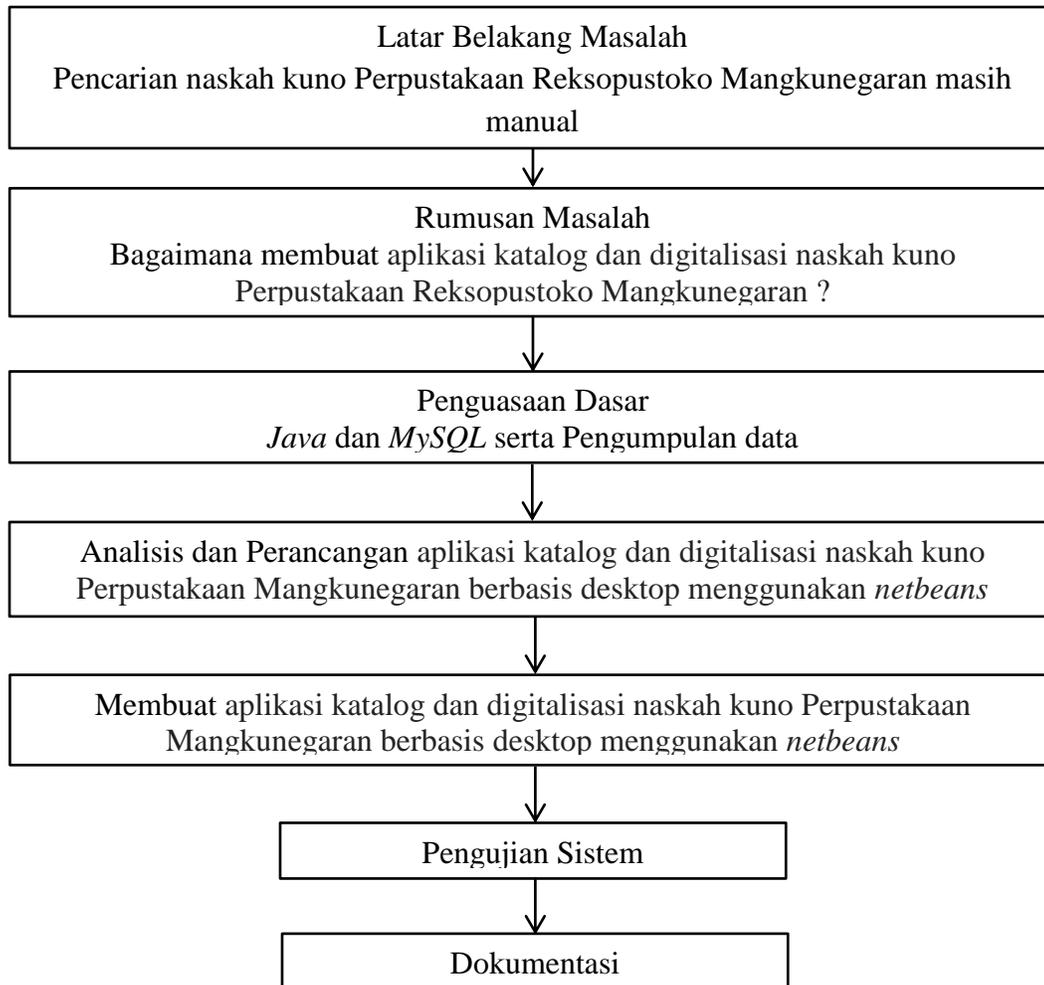
Penelitian dari setiawan (2017) dengan judul Program Aplikasi Perpustakaan Menggunakan *Java Netbeans 8.0.2. Dan Catalogue Online Dengan Database Mysql* membahas tentang penerapan teknologi informasi digunakan sebagai Sistem Informasi Manajemen Perpustakaan. Bidang pekerjaan yang dapat diintegrasikan dengan sistem informasi perpustakaan adalah pengadaan, inventarisasi, katalogisasi, sirkulasi bahan pustaka, pengelolaan anggota, statistik dan lain sebagainya. Fungsi ini sering diistilahkan sebagai bentuk Automasi Perpustakaan. Penerapan teknologi informasi sebagai sarana untuk menyimpan, mendapatkan dan menyebarkan informasi ilmu pengetahuan dalam format digital. Bentuk penerapan teknik informatika dalam perpustakaan ini sering dikenal dengan Perpustakaan Digital. Kedua fungsi penerapan teknologi informasi ini dapat terpisah maupun terintegrasi dalam suatu sistem informasi tergantung dari kemampuan *software* yang digunakan, sumber daya manusia dan infrastruktur peralatan teknologi informasi yang mendukung keduanya.

Penelitian dari setiawan (2017) dengan judul Program Aplikasi Perpustakaan Menggunakan *Java Netbeans 8.0.2. dan Catalogue Online Dengan Database Mysql* membahas tentang Katalog adalah keterangan singkat atau wakil dari suatu dokumen. Katalog perpustakaan elektronik adalah jantung dari sebuah sistem perpustakaan yang terautomasi. Sebuah sistem katalog yang dirancang dengan baik merupakan faktor kunci keberhasilan penerapan automasi perpustakaan.

Kesimpulan dari seluruh penelitian diatas dapat disimpulkan bahwa belum adanya penelitian yang membahas tentang pengelolaan manuskrip kuno dan digitalisasi manuskrip kuno.

2.2 Kerangka Pemikiran

Berikut ini adalah tahapan kerangka pemikiran dalam membuat aplikasi katalog dan digitalisasi naskah kuno Perpustakaan Reksopustoko Mangkunegaran pada Gambar 2.1 berikut:



Gambar 2.1. Kerangka Pemikiran

Penjelasan dari kerangka pemikiran tersebut adalah :

1. Latar Belakang Masalah

Pencarian naskah kuno di Mangkunegaran masih secara manual, sehingga pengunjung memerlukan waktu yang lama untuk menemukan naskah tersebut.

2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan maka rumusan masalahnya adalah bagaimana membuat aplikasi katalog dan digitalisasi naskah kuno Perpustakaan Mangkunegaran berbasis desktop menggunakan *netbeans* ?

3. Penguasaan dasar *Java* dan MySQL

Penguasaan dasar *Java* dan MySQL merupakan suatu kunci terciptanya aplikasi katalog dan digitalisasi naskah kuno Perpustakaan Mangkunegaran tersebut, karena pembuatan aplikasi ini menggunakan bahasa pemrograman *Java* dan menggunakan database MySQL sebagai media penyimpan datanya.

4. Pengumpulan data

Tahap pengumpulan data pada penelitian ini melalui observasi, wawancara dan studi literatur. Pengumpulan data bertujuan untuk mengetahui permasalahan dan kebutuhan informasi mengenai naskah kuno yang berada di Mangkunegaran.

5. Analisis dan perancangan sistem

Analisis dan perancangan aplikasi katalog dan digitalisasi naskah kuno Perpustakaan Mangkunegaran bertujuan untuk mengetahui kesiapan data dan rancangan untuk membangun aplikasi tersebut, sehingga diharapkan dapat membantu pengunjung melakukan pencarian naskah kuno dengan efektif. Perancangan sistem pada penelitian ini menggunakan UML (*Unified Modeling Language*).

6. Rancang Bangun Sistem

Membuat aplikasi katalog dan digitalisasi naskah kuno Perpustakaan Mangkunegaran menggunakan bahasa pemrograman *Java* dan database MySQL sebagai media penyimpanan datanya.

7. Pengujian Sistem

Pengujian sistem merupakan tahap setelah sistem berhasil dibuat. Pengujian sistem bertujuan untuk mengetahui kelebihan, kelemahan serta mengetahui kelayakan suatu sistem untuk digunakan. Pengujian sistem pada penelitian ini menggunakan metode *blackbox* dan kuesioner.

8. Dokumentasi

Pada tahap dokumentasi ini menjelaskan dan memberikan tampilan hasil dari pembuatan aplikasi katalog dan digitalisasi naskah kuno Perpustakaan Mangkunegaran.

2.3 Teori – teori Pendukung

2.3.1 Program Aplikasi

Pengertian Program dalam Kadir (2013) yang dikutip Abidin (2016) adalah istilah yang biasa dipakai oleh pembuat program, sedangkan aplikasi adalah istilah dari sisi pemakai (*user*). Program adalah yang ditulis oleh pemrogram, sedangkan aplikasi adalah hasil terjemahan program.

2.3.2 Pengertian Sistem

Pengertian Sistem dalam Darmawan dan Deni (2013) yang dikutip Fridayanthie dan Charter (2016) adalah “ Kumpulan atau group dari bagian komponen apapun baik fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai suatu tujuan”

Menurut Sutabri (2012) yang dikutip Ermatita (2016) menjelaskan tentang karakteristik dari sistem adalah:

a. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem.

b. Batasan Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisah-pisahkan.

c. Lingkungan Luar Sistem (*Environment*)

Lingkungan luar sistem adalah bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut.

d. Penghubung Sistem (*Interface*)

Penghubung sistem atau *interface* adalah media yang menghubungkan sistem dengan subsistem yang lain.

e. Masukan Sistem (*Input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*).

f. Keluaran Sistem (*Output*)

Keluaran ini merupakan masukan bagi subsistem yang lain. Seperti contoh sistem informasi, keluaran yang dihasilkan adalah informasi, di mana informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang merupakan input bagi subsistem lain.

g. Pengolah Sistem (*Process*)

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

h. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik.

2.3.3 Perpustakaan

Menurut Pamartikawati (2017), perpustakaan merupakan sistem informasi yang di dalamnya terdapat aktivitas pengumpulan, pengolahan, pengawetan, pelestarian dan penyajian serta penyebaran informasi. Perpustakaan sebagaimana yang ada dan berkembang sekarang telah dipergunakan sebagai salah satu pusat informasi, sumber ilmu pengetahuan, penelitian, rekreasi, pelestarian khasanah budaya bangsa, serta memberikan berbagai layanan jasa lainnya. Selain itu menurut perpustakaan adalah sebuah ruangan, bagian sebuah gedung, ataupun gedung itu sendiri yang digunakan untuk menyimpan buku dan terbitan lainnya menurut tata susunan tertentu untuk digunakan pembaca, bukan untuk dijual.

2.3.4 Naskah Kuno

Menurut UU Nomor 43, pasal 1 ayat 4, tentang Perpustakaan (2007) Naskah kuno adalah semua dokumen tertulis yang tidak dicetak atau tidak diperbanyak dengan cara lain, baik yang berada didalam negeri maupun di luar negeri yang berumur sekurang-kurangnya 50 (lima puluh) tahun, dan yang memiliki nilai penting bagi kebudayaan nasional, sejarah dan ilmu pengetahuan

2.3.5 Katalog

Menurut Setiawan (2017) Katalog adalah keterangan singkat atau wakil dari suatu dokumen. Katalog perpustakaan elektronik adalah jantung dari sebuah sistem perpustakaan yang terautomasi. Sebuah sistem katalog yang dirancang dengan baik merupakan faktor kunci keberhasilan penerapan automasi perpustakaan.

2.3.6 Digitalisasi

Menurut Saifudin (2014) Digitalisasi adalah proses alih media dari bentuk tercetak, audio, maupun video menjadi bentuk digital. Sistem digitalisasi dokumen ini dalam pengalihan dokumen fisik ke dokumen digital mengatur beberapa hal seperti scanning, pengindeksan dokumen elektronik dikembalikan ke media kertas.

2.3.7 Java

Menurut Rusli (2016), *Java* adalah bahasa pemrograman yang tergolong pada *high level language* (mudah bagi manusia untuk memahami), mengingat kata-kata/statemennya menyerupai bahasa manusia (english). Namun demikian dalam penelitiannya memerlukan aturan (*Syntax*) yang ketat.

2.3.8 Netbeans

Menurut Ahmadian, dkk (2017), Netbeans adalah *tool* (alat) yang akan digunakan dalam menulis kode-kode pemrograman *java* dalam mata kuliah pemrograman visual ini. Netbeans merupakan salah satu alat *Integrated Development Environment* (IDE) yang sejatinya tidak hanya dipergunakan dalam pengembangan perangkat lunak berbasis *java*, akan tetapi juga dipergunakan oleh beberapa bahasa pemrograman seperti: C++ ataupun PHP. IDE secara sederhana dapat diartikan sebagai aplikasi komputer yang memiliki beberapa fasilitas yang diperlukan dalam

pembangunan perangkat lunak. Tujuan dari IDE adalah untuk menyediakan semua utilitas yang diperlukan dalam membangun perangkat lunak.

Komponen netbeans terdiri dari *Graphic User Interface (GUI)*, *Navigator Windows*, *Palete Windows Properties Windows*, *Source Area*.

1) *GUI builder*

GUI builder merupakan jendela utama yang didalamnya terdapat komponen untuk merancang *GUI*.

2) *Navigator windows*

Navigator windows merupakan jendela yang menampilkan pohon pewarisan dari semua komponen form yang di buka seperti button, label, menu, timer, dan sebagainya.

3) *Palete windows*

Palete windows adalah jendela yang menampilkan daftar semua komponen *swing* yang dapat dimasukan ke dalam form seperti label, button, menu dan lainnya

4) *Properties windows*

Properties windows merupakan jendela yang dapat di ubah memilih komponen yang akan di pergunakan.

5) *Source area*

Source area merupakan jendela yang di gunakan untuk menambahkan kode program pada pemograman *java*.

2.3.9 Keyword

Menurut Ahmadian, dkk (2017), kosa kata (*keyword*) dalam aturan bahasa pemrograman *java*, tak ubahnya seperti aturan pembentukan kosa kata bahasa indonesia yang bertujuan untuk membentuk suatu kalimat dengan SPOK ataupun bahasa inggris dengan gramer-nya. Dalam bahasa pemrograman *java* ada beberapa keyword yang harus diperhatikan yang tentu saja penulisannya harus dibedakan dengan penulisan kode untuk identifier (misalnya, menyamakan nama identifier nama variabel dengan *keyword*). Kumpulan *keyword java* disajikan pada Tabel 2.1.

Tabel 2.1. Keyword 2.4 Access Modifier8

Abstract	Boolean	Break	Byte	Case	Catch
Char	class	Const	Continue	default	Do
Double	Else	Extends	Final	Finally	Float
For	Goto	If	Implements	Import	Instanceof
Int	Interface	long	Native	New	package
Private	protected	Public	Return	Short	Static
Strictfp	Super	Switch	Synchronized	This	Throw
Throws	Transient	Try	Void	Volatile	While
Assert	Enum				

2.3.10 UML (*Unified Modelling Language*)

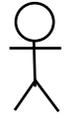
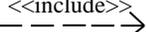
Wahyudi dan luthfi (2011), menyatakan *Unified Modelling Language* atau yang disebut UML adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. UML dapat dipakai membuat model untuk semua jenis aplikasi piranti lunak. Tetapi karena UML juga menggunakan kelas dan operasi dalam konsep dasarnya, maka UML lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C#, atau VB.NET.

Menurut Sukamto dan Shalahuddin (2016) dijelaskan ada 6 (enam) macam diagram dalam *Unified Modeling Language* (UML), yaitu :

2.3.10.1 Use Case Diagram

Use case atau diagram *use case* merupakan pemodelan 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 tersebut. Simbol *use case diagram* dapat dilihat pada Tabel 2.2.

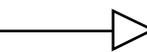
Tabel 2.2 Simbol-simbol *Use Case Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1	Nama <i>Use</i> 	<i>Use case</i>	Fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan kata kerja di awal di awal frase nama <i>use case</i> .
2		<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3		<i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4	<<extend>> 	<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
5		<i>Generalisasi</i> <i>on</i>	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6	<<include>> 	<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

2.3.10.2 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode operasi. Kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak dapat membuat kelas-kelas di dalam program perangkat lunak sesuai dengan perancangan diagram kelas (Sukanto & Shalahuddin, 2016). Simbol *class diagram* dapat dilihat pada Tabel 2.3.

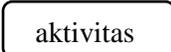
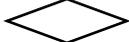
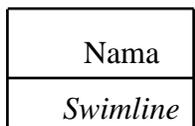
Tabel 2.3 Simbol-simbol *Class Diagram*

No	SIMBOL	NAMA	KETERANGAN			
1	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">Nama kelas</td> </tr> <tr> <td style="padding: 2px;">+ Atribut</td> </tr> <tr> <td style="padding: 2px;">+ Operasi()</td> </tr> </table>	Nama kelas	+ Atribut	+ Operasi()	<i>Class</i>	Kelas pada struktur sistem
Nama kelas						
+ Atribut						
+ Operasi()						
2		<i>Interface</i>	Sama dengan konsep interface dalam pemrograman berorientasi objek.			
3		<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .			
4		<i>Directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain biasanya juga disertai dengan <i>multiplicity</i> .			
5		<i>Generalization</i>	Relasi antar kelas dengan makna generalisasi - spesialisasi (umum - khusus).			
6		<i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.			
7		<i>Aggregation</i>	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).			

2.3.10.3 Activity Diagram

Activity diagram menggambarkan *workflow* atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Sukamto & Shalahuddin, 2016). Simbol *Activity Diagram* ditunjukkan pada Tabel 2.4.

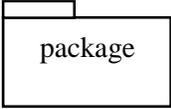
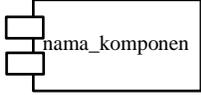
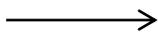
Tabel 2.4 Simbol-simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		Status awal	Status awal aktivitas sistem.
2		Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3		<i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4		<i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5		Status akhir	Status akhir yang dilakukan sistem.
6		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.3.10.4 Component Diagram

Diagram komponen atau *component diagram* dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem (Sukamto & Shalahuddin, 2016). Simbol *component diagram* dilihat pada Tabel 2.5.

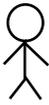
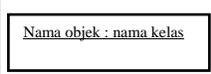
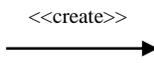
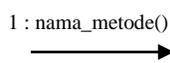
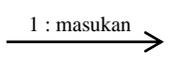
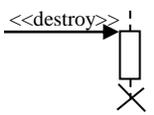
Tabel 2.5 Simbol *Component Diagram*.

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Package</i>	<i>Package</i> merupakan sebuah bungkus dari satu atau lebih komponen
2		<i>Component</i>	Komponen sistem
3		<i>Dependency</i>	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai
4		<i>Interface</i>	Sama dengan konsep <i>interface</i> pada pemrograman berorientasi objek, yaitu sebagai antarmuka komponen agar tidak mengakses langsung komponen
5		<i>Link</i>	Relasi antar komponen

2.3.10.5 Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek (Sukamto & Shalahuddin, 2016). Simbol *sequence diagram* dapat dilihat pada Tabel 2.6.

Tabel 2.6. Tabel *Sequence Diagram*

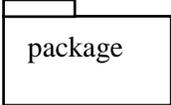
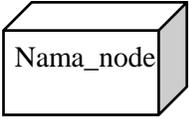
NO	SIMBOL	NAMA	KETERANGAN
1		Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
2		<i>Lifeline</i>	Menyatakan kehidupan suatu objek.
3		Objek	Menyatakan objek yang berinteraksi pesan.
4		Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi.
5		Pesan tipe create	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6		Pesan tipe call	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri.
7		Pesan tipe send	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8		Pesan tipe <i>destroy</i>	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

2.3.10.6 Deployment Diagram

Deployment diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram deployment juga dapat digunakan untuk memodelkan sistem tambahan yang menggambarkan rancangan *device*, *node*, dan *hardware*.

Sistem *client*, sistem terdistribusi murni, rekayasa ulang aplikasi (Sukamto & Shalahuddin, 2016). Simbol *deployment* diagram ditunjukkan pada Tabel 2.7.

Tabel 2.7. Simbol - Simbol *Deployment*

NO	SIMBOL	NAMA	KETERANGAN
1.		<i>Package</i>	<i>Package</i> merupakan sebuah bungkusuan dari satu atau lebih <i>node</i> .
2.		<i>Node</i>	Biasanya mengacu pada perangkat keras (<i>hardware</i>), perangkat lunak yang tidak dibuat sendiri (<i>software</i>), jika didalam <i>node</i> disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelum pada diagram komponen.
3.		<i>Dependency</i>	Kebergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai.
4.		<i>Link</i>	Relasi antar <i>node</i> .

2.3.11 Sistem Basis Data

Pengertian sistem basis data dalam Marlinda (2004) yang dikutip Faizal dan Putri adalah suatu sistem menyusun dan mengelola record-record menggunakan computer untuk menyimpan atau merekam serta memelihara data operasional lengkap sebuah organisasi atau perusahaan sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk proses pengambilan keputusan

2.3.12 MySQL

Menurut Setyorini (2013), MySQL (bisa dibaca dengan mai-es-ki-el atau bisa juga mai-se-kuel) adalah suatu perangkat lunak database relasi (*Relational Database Management System* atau *RDBMS*), seperti halnya ORACLE, Postgresql, Ms SQL,

dan sebagainya. Jangan disalah-artikan MySQL dengan SQL. SQL (singkatan dari *Structured Query Language*) sendiri adalah suatu sintaks perintah-perintah tertentu atau bahasa (pemrograman) yang digunakan untuk mengelola suatu database. Jadi, MySQL adalah dua ‘*mahluk*’ yang berbeda. Mudahnya, MySQL adalah softwarena, dan SQL adalah bahasa perintahnya

2.3.13 XAMPP

Menurut Indrawan (2013), XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (localhost), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (sistem operasi apapun), Apache, MySQL, HP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis.

2.3.14 Black-box Testing

Menurut Sukamto dan Salahuddin (2016), Black-box testing adalah menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian black-box testing harus dibuat dengan kasus benar dan kasus salah

2.3.15 Pengujian Kuesioner

Menurut Sugiono (2017), Kuesioner adalah teknik pengumpulan data yang dilakukan dengan cara memberi seperangkat pertanyaan atau pernyataan tertulis kepada responden untuk dijawabnya. Kuesioner merupakan teknik pengumpulan data yang efisien bila peneliti tahu dengan pasti variabel yang akan diukur dan tahu apa

yang diharapkan dari responden. Selain itu, kuesioner juga cocok digunakan bila jumlah responden cukup besar dan tersebar di wilayah yang luas