

BAB II

LANDASAN TEORI

Penyusunan Laporan Tugas Akhir ini diperlukan informasi terhadap sekolah, tinjauan pustaka dan landasan teori sebagai landasan dalam membuat Laporan Tugas Akhir.

2.1 Kajian pustaka

Dalam mengembangkan *website* SMA Muhammadiyah 5 Karanganyar dalam proses pendaftaran siswa baru yang dilakukan secara *online* agar dapat memudahkan calon siswa baru dalam melakukan pendaftaran ke SMA Muhammadiyah 5 Karanganyar dengan lebih mudah dan lebih efektif.

Adapun dalam mengembangkan *website* SMA Muhammadiyah 5 Karanganyar ada beberapa penelitian yang terkait dalam pembuatan penerimaan siswa baru secara *online* menjadi dasar penelitian ini dilakukan sebagai berikut;

1. Penelitian dari Almustaqim dan Mulyono (2017) dengan judul Analisis Dan Pengembangan Sistem Informasi Penerimaan Mahasiswa Baru Berbasis *Web* Pada Pascasarjana Universitas Jambi membahas tentang Sistem informasi penerimaan mahasiswa baru berbasis *web* pada Pascasarjana Universitas Jambi masih terdapat beberapa kekurangan, yaitu tidak adanya akses untuk login ke sistem oleh calon mahasiswa sehingga tidak dapat memperbaiki biodata, mencetak formulir berulang kali dan tidak adanya laporan untuk admin maupun prodi tentang berapa banyak calon mahasiswa yang terdaftar. Tujuan penelitian ini adalah untuk memudahkan calon mahasiswa dalam memperbaiki data diri dan memudahkan admin maupun prodi dalam melihat banyaknya calon mahasiswa yang terdaftar. Metode yang digunakan yaitu *waterfall*. Dari hasil pengujian modul -modul berjalan dengan baik dan masalah-masalah yang sering terjadi pada sistem yang lama sudah dapat teratasi.
2. Penelitian dari Cahyana, Prasetyo, dan Himawan (2013) dengan judul aplikasi penerimaan siswa baru berbasis *web* (SMK Negeri 3 Yogyakarta) yang membahas tentang Permasalahan yang terjadi pada masa pendaftaran adalah masih

digunakannya formulir, yang menyebabkan antrian panjang, sehingga calon mahasiswa sering kerepotan, selain nilai tertinggi dan terendah UAS sebagai acuan diterimanya dari calon siswa baru hanya bisa dilihat dengan mengunjungi langsung. Jadi disini dibuat Aplikasi Pendaftaran Berbasis *Web* untuk memecahkan masalah di atas. Berdasarkan latar belakang masalah diatas, Maka dibangun sistem Penerimaan Seleksi Berbasis *Web* dengan menggunakan pengembangan perangkat lunak *sekuensial linier* (air terjun), yang memiliki beberapa tahapan, termasuk persyaratan perangkat lunak tahap analisis, tahap desain, tahap pembuatan kode, tahap pengujian, dan tahap pemeliharaan. sehingga dihasilkan aplikasi memiliki keunggulan dalam pengolahan data. Dalam aplikasi ini juga dapat memetakan sekolah, tempat pembuatan peta menggunakan Google Maps untuk menghasilkan peta yang memiliki lokasi sesuai dengan lokasi dari kondisi aslinya.

3. Penelitian dari Sholihin dan Mujilawati (2014) dengan judul Sistem Penerimaan Peserta Didik Baru Berbasis *Web (Online)* Di SMK Muhammadiyah 7 Kedungpring Lamongan membahas tentang Dunia pendidikan sekarang sudah merambah ke internet. Banyak sekolah yang menerima peserta didik baru dari internet, khususnya sekolah yang memang peminatnya tidak hanya berasal dari dalam kota, tetapi juga dari luar kota/negara. Dengan internet, jarak ribuan kilometer akan dapat dijangkau dengan sekejap mata. Berdasarkan hal tersebut, maka dilakukan rancang bangun sistem informasi PPDB *Online* di SMK Muhammadiyah 7 Kedungpring Lamongan. Sistem informasi yang dihasilkan dari laporan ini digunakan untuk menerima peserta didik baru secara *online* dalam tahun ajaran baru.

Sistem informasi PPDB *Online* ini dirancang dengan menggunakan *case tool* Power Designer 15.1 dan dibangun menggunakan bahasa pemrograman PHP dan database MySQL. Sistem informasi dapat diakses oleh dua entitas, yaitu panitia dan calon peserta didik baru. Panitia PPDB melakukan validasi berkas pendaftaran dan menghasilkan laporan peserta didik baru yang selanjutnya diumumkan ke calon siswa baru dan Bagian Kesiswaan. Sistem Penerimaan Peserta Didik Baru *Online* ini dibuat dengan *interface* yang memberikan keleluasaan bagi panitia PSB dalam mengolah data peserta didik baru dan

siswa baru yang diterima di SMK Muhammadiyah 7 Kedungpring Lamongan. Informasi-informasi yang berhubungan dengan calon peserta didik baru, baik yang gagal masuk SMK Muhammadiyah 7 Kedungpring Lamongan maupun yang diterima, disimpan dalam suatu database. Sehingga data akan lebih mudah diakses daripada dalam bentuk *hardcopy*

4. Penelitian dari Prasetyo, Susanti, dan Triyanto (2017) dengan judul Adopsi Cloud Computing Untuk Aplikasi Cbt Penerimaan Siswa Baru Sma/Sederajat yang membahas tentang Sistem seleksi penerimaan siswa baru yang diadakan di sekolah-sekolah terutama untuk SMA/ sederajat sebagian sudah mulai beralih menggunakan sistem computer based test (CBT). Dengan adanya aplikasi ini diharapkan proses pelaksanaan Penerimaan Peserta Didik Baru dapat terlaksana dengan baik dan efektif. aplikasi yang dibangun menggunakan konsep cloud computing yaitu gaya komputasi di mana sumber daya komputasi mudah diakses dan langsung dapat digunakan.

5. Penelitian dari santoso dan Nurwati (2016) dengan judul Rancangan Sistem Informasi Pendaftaran Siswa Baru membahas tentang Informasi dibutuhkan untuk semua lini organisasi, baik untuk mempermudah manajemen dalam mengambil keputusan maupun untuk operasional organisasi tersebut. Untuk mempermudah mengorganisir data yang merupakan bagian dari informasi diperlukan suatu sistem yang dapat menampung dan menyediakan setiap saat informasi diperlukan.

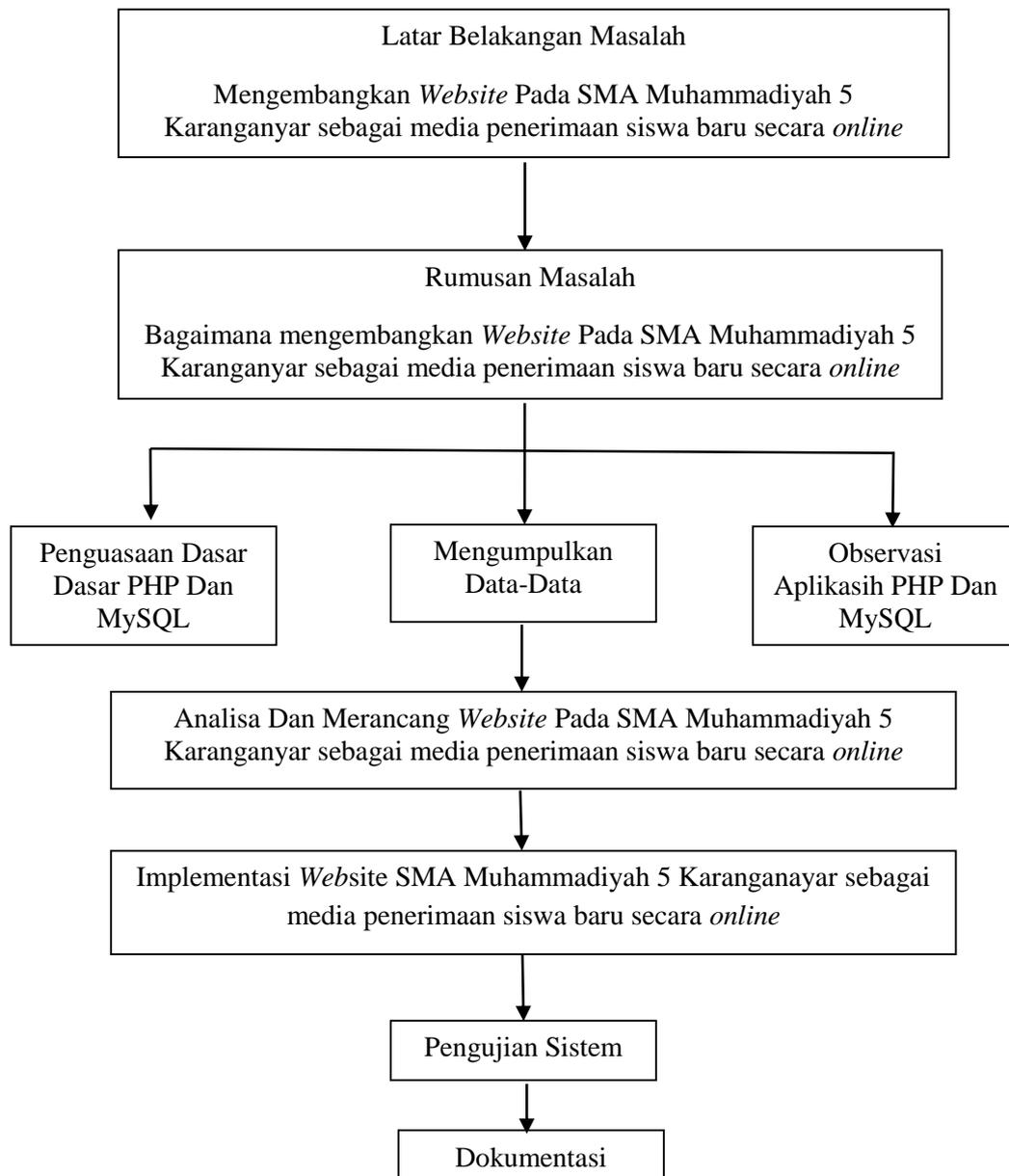
Kesimpulan dari beberapa penelitian di atas bawah dapat disimpulkan dalam Tabel 2.1 Keunggulan dan kelemahan dari sistem penerimaan siswa baru.

Tabel 2.1 Keunggulan dan Kelemahan Dari Sistem Penerimaan Siswa Baru

No	Tempat Penelitian	Keunggulan	Kelemahan
1	Pascasarjana Universitas Jambi	Dapat memudahkan calon mahasiswa dalam memperbaiki data diri dan memudahkan admin maupun prodi dalam melihat banyaknya calon mahasiswa yang terdaftar melalui websitenya tersebut	Identifikasi permasalahan sistem yang dibangun adalah tidak adanya fitur untuk memberitahu informasi kepada calon mahasiswa melalui sms karena tidak semua calon mahasiswa memiliki akses internet, apalagi yang berada di sebuah desa yang jarang terjangkau internet
2	SMK Negeri 3 Yogyakarta	aplikasi memiliki keunggulan dalam pengolahan data. Dalam aplikasi ini juga dapat memetakan sekolah, tempat pembuatan peta menggunakan Google Maps untuk menghasilkan peta yang memiliki lokasi sesuai dengan lokasi dari kondisi aslinya	Pada bagian tampilannya pada sistem masih begitu sederhana . keamanan data masih begitu kurang.
3	SMK Muhammadiyah 7 Kedungpring Lamongan	Sistem informasi PPDB <i>Online</i> ini dirancang dengan menggunakan case tool Power Designer 15.1 dan dibangun menggunakan bahasa pemrograman PHP dan database MySQL	tampilan yang diberikan oleh sistem ini masih kurang menarik dan tak semua siswa dapat melakukan pendaftaran secara online karena kendala jaringan di daerahnya.
4	SMA Negeri 2 Kudus dan SMK Negeri 1 Kudus	Sistem seleksi penerimaan siswa baru yang diadakan di sekolah-sekolah terutama untuk SMA/ sederajat sebagian sudah mulai beralih menggunakan sistem computer based test (CBT)	Kemungkinan service performance yang masih kurang konsisten dari provider.
5	Taman Pendidikan Alqur'an SAQUBI Bintaro	sistem informasi yang mampu menampung kegiatan proses bisnis pendaftaran siswa baru bagi Taman Pendidikan Al Qur'an SAQUBI.	Sistem tersebut belum dapat diterapkan karena masyarakat masih belum paham terhadap sistem yang akan dibuat dan kekurangannya terhadap pengetahuan internet.

2.2 Kerangka Pemikiran

Berikut ini adalah tahapan kerangka pemikiran yang akan dijalankan oleh penulis dalam pengembangan *website* SMA Muhammadiyah 5 Karanganyar seperti pada Gambar 2.1.



Gambar 2.1 Kerangka Pemikiran

2.3 Landasan Teori

2.3.1 Informasi

Informasi merupakan data-data masukan yang dapat dijadikan dasar untuk membuat keputusan. Data merupakan bentuk yang masih mentah yang belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut (Susena, 2013).

Sedangkan menurut Kadir (2003) dalam bukunya menerangkan bahwa, informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan berarti bagi yang menerimanya dan bermanfaat bagi pengambilan keputusan saat atau saat mendatang.

2.3.2 Sistem Informasi

Sistem Informasi adalah sekumpulan sub sistem yang saling berhubungan, berkumpul bersama-sama antara bagian satu dengan lainnya dengan cara-cara tertentu untuk melakukan fungsi pengolahan data, menerima masuk (*input*) berupa data-data, kemudian mengolahnya (*processing*) dan menghasilkan keluaran (*output*) berupa informasi sebagai dasar pengambilan keputusan yang berguna dan mempunyai nilai nyata (Sutanta, 2011).

Menurut Mulyanto (2009) “Sistem Informasi merupakan suatu komponen yang terdiri dari manusia, teknologi informasi dan prosedur kerja memproses, menyimpan, menganalisa dan menyelesaikan informasi untuk mencapai tujuan”.

2.3.3 Website

Web adalah fasilitas *hypertext* untuk menampilkan data berupa teks, gambar, bunyi, dan animasi dan data multimedia lainnya, yang di antara data tersebut saling berhubungan satu sama lainnya (Anonim, 2011).

Website (situs *web*) adalah merupakan alamat (URL) yang berfungsi sebagai tempat penyimpanan data dan informasi dengan berdasarkan topik tertentu (Sutarman, 2007).

Website (Situs *Web*) merupakan kumpulan dari halaman-halaman *web* yang berhubungan dengan file-file lain yang terkait. Dalam sebuah *website* terdapat suatu

halaman yang dikenal dengan sebutan *home page*. *Home page* adalah sebuah halaman yang pertama kali dilihat ketika seseorang mengunjungi *website*. Dari *home page*, pengunjung dapat mengklik *hyperlink* untuk pindah ke halaman lain yang terdapat dalam *website* tersebut (Jhonsen, 2004).

2.3.4 PHP

PHP atau kependekan dari Hypertext Preprocessor adalah salah satu bahasa pemrograman open source yang sangat cocok atau dikhususkan untuk pengembangan *Web* dan dapat di tanamkan pada sebuah skrip HTML. Bahasa PHP dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti C, Java, dan Perl serta mudah untuk dipelajari (A.M & Virgi, 2011).

Menurut Kadir (2008) menurut dokumen resmi PHP, PHP merupakan singkat dari PHP *Hypertext Preprocessor*. Ia merupakan bahasa berbentuk skrip yang ditempatkan dalam *server* dan diproses di *server*. Hasilnya yang dikirimkan ke klien, tempat pemakai menggunakan *browser*. Secara khusus, PHP dirancang untuk membentuk aplikasi *web* dinamis. Artinya, ia dapat membentuk suatu tampilan berdasarkan permintaan terkini. Misalnya, Anda bisa menampilkan isi *database* ke halaman *web*. Pada prinsipnya PHP mempunyai fungsi yang sama dengan skrip-skrip seperti ASP (*ActiveServer Page*), Cold Fusion, ataupun Perl. Namun, perlu diketahui bahwa PHP sebenarnya dipakai secara *command line*. Artinya, skrip PHP dapat dijalankan tanpa melibatkan *web server* maupun *browser*.

2.3.5 MySQL

MySQL (*My Structure Query Language*) adalah salah satu *DataBase Management System* (DBMS). MySQL berfungsi untuk mengelola database menggunakan bahasa SQL. MySQL bersifat open source sehingga kita bisa menggunakannya secara gratis. Pemrograman PHP juga sangat mendukung/ *support* dengan database MySQL (Anhar, 2010).

Menurut Kadir (2008) MySQL adalah salah satu jenis *database server* yang sangat terkenal. Kepopulerannya disebabkan MySQL menggunakan SQL sebagai

bahasa dasar untuk mengakses *database* nya. MySql termasuk jenis RDBMS (*Relational Database Management System*). Itulah sebabnya, istilah seperti tabel, baris, dan kolom digunakan pada MySQL. Pada MySQL, sebuah *database* mengandung satu atau sejumlah tabel. Selain itu, ia bersifat *Open Source* (tidak perlu membayar untuk menggunakannya) pada pelbagai *platform* (kecuali untuk jenis *Enterprise*, yang bersifat komersial).

2.3.6 XAMPP

XAMPP merupakan tool yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dengan menginstal XAMPP maka tidak perlu lagi melakukan instalasi dan konfigurasi *web server Apache*, PHP dan MySQL secara manual (Terigan, 2011)

XAMPP merupakan merupakan paket php berbasis *open source* yang dikembangkan oleh sebuah komunitas Open Source. Dengan menggunakan XAMPP tidak perlu lagi melakukan penginstalan program yang lain karena semua kebutuhan telah disediakan oleh XAMPP. Beberapa paket yang telah disediakan adalah Apache, MySql, Php, Filezila, dan Phpmyadmin (Nugroho, 2011)

2.3.7 CSS

CSS atau *cascading style sheet* adalah sekumpulan kode pemrograman *web* yang berfungsi untuk mengendalikan beberapa komponen di dalam *web* sehingga menjadi setampak seragam, berstruktur, dan teratur .selain itu, CSS juga dapat diartikan sebagai kumpulan kode yang digunakan untuk mendefinisikan disain dari bahasa *markup*. Salah satu bahasa *markup* adalah HTML. CSS membuat mengubah desain dari text, warna, dan latar belakang dari hampir semua kode tag HTML Komputer (2015).

Cascading Style Sheet(CSS) adalah salah satu bahasa untuk mengatur tampilan dalam *web* yang berfungsi memisahkan antara desain dengan content. CSS dapat ditulis untuk mendefinisikan ulang tag-tag HTML ataupun dengan menambah atribut id dan class pada tag HTML(Team, 2010).

2.3.8 HTML

HTML atau *Hypertext Markup Language* merupakan bahasa pemrograman yang biasa digunakan untuk membuat aplikasi berbasis *web*. Bahas pemrograman ini ditulis dalam berkas format ASCII, supaya dapat menghasilkan tampilan wujud yang terintergrasi. HTML adalah bahasa pemrograman web yang umum dan biasa digunakan serta mudah dipelajari. (Maya, Bowo, Hs, dan Irawan, 2015).

2.3.9 Bootstrap

Bootstrap sebuah alat bantu untuk membuat sebuah tampilan halaman *website* yang dapat mempercepat pekerjaan seorang pengembang *website* ataupun pendesain halaman *website*. Sesuai namanya, *website* yang dibuat oleh alat bantu ini memiliki tampilan halaman yang sama atau mirip dengan tampilan halaman *twitter* atau desainer juga dapat mengubah tampilan halaman *website* sesuai dengan kebutuhan. Tampilan *website* yang dibuat *bootstrap* akan menyesuaikan ukuran layar dari *browser* yang kita gunakan baik *dekstop*, tablet ataupun *mobile device*. Fitur ini bisa diaktifkan ataupun di-*non*-aktifkan sesuai keinginan. Sehingga, membuat web untuk tampilan *dekstop* saja dan apabila di-*reader* oleh *mobile browser* maka tampilan dari *web* yang dibuat sesuai layar. Dengan *bootstrap* kita juga bisa membangun *web* dinamis ataupun statis (Widyantoro, 2014).

2.3.10 Pengertian Media

Media adalah semua bentuk perantara yang digunakan oleh manusia untuk menyampaikan atau menyebar ide, gagasan atau pendapat, sehingga ide, gagasan atau pendapat yang dikemukakan itu sampai kepada penerima yang dituju. (Arsyad, 2002).

2.3.11 Pengertian Siswa

Siswa adalah mereka yang secara khusus diserahkan oleh orang tua untuk mengikuti pembelajaran yang diselenggarakan di sekolah dengan tujuan untuk menjadi manusia yang memiliki pengetahuan, berkepribadian, berpengalaman, berkepribadian, berakhlak dan mandiri. (Mohammad, 2010).

2.3.12 Entity Relationship Diagram (ERD)

Menurut Cannolly dan Begg (2010) *Entity Relation Diagram* digunakan untuk menggambarkan struktur logical *database* dalam bentuk diagram ERD, serta menyediakan cara yang sederhana dan mudah untuk memahami bagian berbagai komponen dalam desain *database*. Komponen antara lain :

1. Relationship

Relationship adalah hubungan antara *entity*. *Entity* merupakan pengikut dari *relationship*. *Relationship* dapat digambarkan dalam bentuk belah ketupat yang mana berisi nama dari relasi tersebut.

Relationship dapat berupa hubungan *one-to-one*, *one-to-many*, atau *many-to-many*.

- a. Hubungan *one-to-one* (1 atau 1..1) Hubungan dimana setiap *entity* yang ada hanya dapat mempunyai maksimal 1 (satu) hubungan dengan entityyang lain.
- b. Hubungan *one-to-many* (1..*) Hubungan dimana setiap *entity* yang dapat mempunyai satu atau lebih dari satu hubungan dengan *entity* yang lain.
- c. Hubungan *many-to-many* (*..*) Hubungan dimana setiap *entity* dapat mempunyai lebih dari satu relasi dengan entitylainnya.

2.3.13 UML

Pada perkembangan teknologi perangkat lunak diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisari agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modelling Language*. (Rosa dan Shalahuddin, 2016)

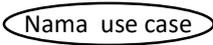
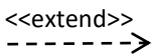
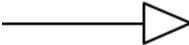
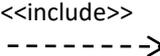
Unified Modelling Language (UML) merupakan bahasa visual untuk pemodelan.

2.3.13.1 Use Case Diagram

Use case atau *diagram use case* merupakan pemodelan untuk melakukan sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi

antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Rosa dan Shalahuddin (2016) Simbol-simbol yang ada pada diagram *use case* dapat dilihat pada Tabel 2.2

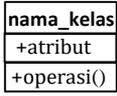
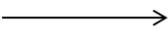
Tabel 2.2 Simbol-simbol *Use Case Diagram*

NO.	SIMBOL	NAMA	KETERANGAN
1.		<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.		<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.
4.		<i>Generalization</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.
5.		<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.

2.3.13.2 *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. Rosa dan Shalahuddin (2016) Simbol-simbol yang ada pada *class diagram* dapat dilihat pada Tabel 2.3.

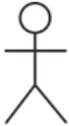
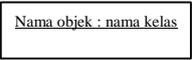
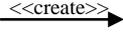
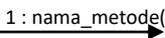
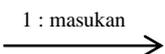
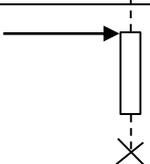
Tabel 2.3 Simbol-simbol *Class Diagram*

NO.	SIMBOL	NAMA	KETERANGAN
1.		<i>Class</i>	Kelas pada struktur sistem
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>).

2.3.13.3 Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Rosa dan Shalahuddin (2016) Simbol-simbol yang ada pada *sequence diagram* dapat dilihat pada Tabel 2.4.

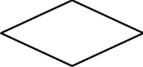
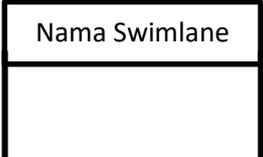
Tabel 2.4 Simbol-simbol *Sequence Diagram*

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> .

2.3.13.4 Activity Diagram

Diagram aktivitas atau *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. Rosa dan Shalahuddin (2016) Simbol-simbol yang ada pada *activity diagram* dapat dilihat pada Tabel 2.5

Tabel 2.5 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.

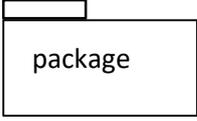
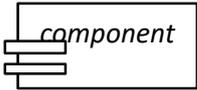
2.3.13.5 *Component Diagram*

Diagram komponen *atau component diagram* dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Rosa dan Shalahuddin (2016) Simbol-simbol yang ada pada *component diagram* dapat dilihat pada Tabel 2.6.

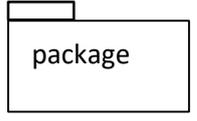
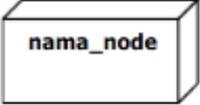
2.3.13.6 *Deployment Diagram*

Deployment diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat digunakan untuk memodelkan sistem tambahan, sistem *client*, sistem terdistribusi murni, rekayasa ulang aplikasi. Rosa dan Shalahuddin (2016) Simbol-simbol yang ada pada *deployment diagram* dapat dilihat pada Tabel 2.7

Tabel 2.6 Simbol *Component Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1.		<i>package</i>	<i>Package</i> merupakan sebuah bungkusan 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
NO	GAMBAR	NAMA	KETERANGAN
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

Tabel 2.7 Simbol-Simbol *Deployment Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1.		<i>Package</i>	<i>Package</i> merupakan sebuah bungkusan 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> .

2.3.14 Pengujian Sistem Dengan *Black-Box*

Pengujian untuk validasi memiliki beberapa pendekatan sebagai berikut:

Black-Box Testing (pengujian kotak hitam)

Yaitu 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 kotak hitam harus dibuat dengan benar dan kasus salah, misalkan untuk kasus proses login maka kasus uji yang dibuat adalah :

- Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
- Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah. (Rosa dan Shalahuddin, 2016)

2.3.15 Pengujian Sistem Dengan Kuisisioner

Kuisisioner adalah hasil dari objektif. Kuisisioner dapat dilakukan kepada banyak orang sekaligus dengan waktu yang lebih singkat. Kelemahan kuisisioner adalah responden cenderung malas untuk mengisi kuisisioner dan sering terjadi kesulitan membuat pertanyaan yang singkat, jelas, dan dipahami. (Tohari, 2014)

Metode penilaian pengujian yang digunakan dengan metode kuantitatif berdasarkan data sampel dari pengguna dan kuisisioner disebarkan kepada beberapa responden dari kalangan mahasiswa, sekolah, dan umum. Kuisisioner terdiri dari beberapa pertanyaan yang memiliki skala pengukuran dari teknik likert. Skala likert merupakan metode yang mengukur sikap dengan menyatakan setuju atau tidak setuju terhadap subyek, obyek atau kejadian tertentu. Data yang diperoleh dapat dianalisa dengan menghitung rata-rata jawaban berdasarkan scoring setiap jawaban dari responden. Analisa dapat dilakukan sebagai berikut:

Jumlah skor ideal untuk seluruh item ditunjukkan Persamaan 1

$$\text{Jumlah skor} = 4 \times \text{jumlah responden} \quad (1)$$

Rata-rata skor dapat dalam presentase menggunakan rumus ditunjukkan Persamaan 2

$$\text{Rata-rata skor} = \frac{\text{jumlah skor}}{\text{jumlah ideal}} \times 100\% \quad (2)$$