

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Tinjauan Pustaka**

Penelitian mengenai Pengembangan Sistem Penanganan Keluhan Berbasis *Web* (Studi Kasus Fakultas Teknik Universitas Muhammadiyah Yogyakarta) sebelumnya telah dikaji oleh Saputra (2012). Namun, dari penelitian tersebut masih diperlukan pembaharuan dalam sistem dengan menambahkan beberapa fungsi pada Sistem Penanganan Keluhan Berbasis *Web* tersebut dengan mengkaji konsep baru. Atas dasar tersebut, peneliti mengkaji beberapa penelitian yang memiliki teori maupun konsep yang berkaitan dengan penelitian yang akan dilakukan oleh peneliti. Berikut ini beberapa penelitian terkait yang dipaparkan oleh peneliti, yaitu :

Pertama, dalam penelitian Yohanitas & Prayitno (2014) yang berjudul “*Pengelolaan Pengaduan Masyarakat Kota Bekasi (Bekasi City Public Complaints Management)*”, membahas mengenai pola yang diterapkan oleh kota Bekasi atas pengelolaan pengaduan yang diajukan masyarakatnya serta memaparkan beberapa contoh yang ditempuh dengan tujuan pengelolaan pengaduan kota Bekasi dapat menjadi pembanding yang baik terhadap instansi lain. Pengelolaan Pengaduan Masyarakat di kota Bekasi menggunakan metode deskriptif eksploratif terhadap peraturan yang telah ada. Metode tersebut digunakan untuk menjelaskan secara lugas dan terstruktur terhadap pengaduan yang disampaikan masyarakat kota Bekasi. Untuk merealisasikan metode tersebut, kota Bekasi telah mengemasnya dalam sebuah sistem pelayanan penyediaan informasi dan pengaduan masyarakat menggunakan sarana teknologi website dan sms center. Untuk menunjang keberhasilan pelayanannya, kota Bekasi telah memberikan aturan yang jelas mengenai pemberian informasi pengaduan masyarakat dan melakukan transparansi data pengaduan masyarakat serta manajemen yang sederhana.

Dari hasil penelitian yang dilakukan oleh Yohanitas & Prayitno (2014) dengan penelitian yang dilakukan oleh peneliti yaitu sama-sama membahas

pengelolaan pengaduan dengan menggunakan sarana teknologi website. Namun untuk penelitian Yohanitas & Prayitno (2014) diterapkan untuk Kota Bekasi, sedangkan peneliti menerapkannya pada Website Pengaduan Keluhan Universitas Muhammadiyah Yogyakarta. Terdapat juga kesamaan penelitian yang dilakukan oleh Yohanitas & Prayitno (2014) dengan peneliti yaitu sama-sama bertujuan untuk membuat pengguna (masyarakat atau mahasiswa) dapat menjelaskan secara lugas dan terstruktur terhadap pengaduan keluhan yang disampaikan.

Kedua, dalam penelitian Lestari, dkk. (2015) yang berjudul “*Pengelolaan Pengaduan Pelayanan Publik Berbasis E-Government (Studi Kasus Pengelolaan Pengaduan Rakyat Online Denpasar Pada Dinas Komunikasi Dan Informatika Kota Denpasar Tahun 2014)*”. Penelitian tersebut bertujuan untuk mengetahui Pengelolaan Pengaduan Rakyat *Online* (PRO) dengan penerapannya di beberapa SKPD (Satuan Kerja Perangkat Daerah) dan sejauh mana *feedback* yang diperoleh masyarakat dari adanya PRO tersebut. Dalam penelitian Lestari, dkk. (2015) konteks penerapan *E-Government* dalam pengelolaan Pengaduan Rakyat *Online* (PRO) Denpasar termasuk dalam lingkup *Government to Citizens* (Pemerintah ke Masyarakat), karena Pemerintah Kota Denpasar dalam hal ini Dinas Komunikasi dan Informatika membangun dan menerapkan aplikasi teknologi informasi berupa Pengaduan Rakyat *Online* Denpasar yaitu aplikasi berbasis website yang melibatkan partisipasi publik dan bersifat dua arah, yang digunakan sebagai alat bantu untuk melakukan monitoring dan verifikasi pencapaian program pembangunan maupun pengaduan masyarakat terkait pelaksanaan program pembangunan dan pelayan publik di Kota Denpasar.

Adanya PRO Denpasar memberikan dampak positif sehingga masyarakat tidak perlu kebingungan lagi untuk mengadu dan tidak perlu datang secara langsung ke Instansi terkait karena bisa diakses dari rumah. Namun, hasil dari penelitian ini pada praktiknya dapat disimpulkan belum optimal. Hal ini dikarenakan kurangnya koordinasi dan persamaan persepsi mengenai standar pelayanan dan prosedur dalam pengelolaan Pengaduan Rakyat *Online* (PRO)

Denpasar, serta masih minimnya umpan balik (*feedback*) dari masyarakat atas respon terhadap kepuasan masyarakat terkait kualitas pelayanan yang telah diberikan oleh instansi yang bersangkutan.

Dari hasil penelitian Lestari, dkk. (2015) dengan peneliti terdapat juga beberapa kesamaan yaitu sama-sama menggunakan media website dalam mengelola semua laporan pengaduan atau keluhan, mempermudah pengguna (masyarakat atau mahasiswa) dalam menyampaikan aspirasi mereka berupa keluhan maupun saran, serta memiliki tujuan untuk memonitor jalannya proses pengelolaan pengaduan tersebut serta memverifikasi pencapaian yang diberikan oleh pengguna (masyarakat atau mahasiswa) terhadap tindak lanjut atas pengaduan atau keluhan yang telah dilakukan oleh instansi terkait. Namun, dalam hal ini penelitian yang dilakukan Lestari, dkk. (2015) masih memiliki kendala dalam proses verifikasi pencapaian atas pengaduan atau keluhan karena umpan balik masyarakat Denpasar yang masih kurang.

Ketiga, dalam penelitian Prasetya, dkk. (2013) yang berjudul “*Analisis Pengelolaan Pengaduan Masyarakat Dalam Rangka Pelayanan Publik (Studi Pada Dinas Komunikasi dan Informatika Kota Malang)*”. Dalam penelitian tersebut, proses pengaduan dilakukan masyarakat melalui *website* yang ada di Pemerintah Kota Malang atau melalui *Telpon* ke instansi yang berkaitan, serta masyarakat juga bisa langsung datang ke instansi yang terkait untuk melakukan pengaduan. Selain itu, masyarakat juga bisa mengecek sampai dimana proses pengaduan mereka telah ditindak lanjuti oleh pemerintah yaitu dengan tiket atau kode yang telah didapatkan masyarakat saat melakukan pengaduan melalui *website* yang telah disediakan pemerintah. Namun, terdapat faktor penghambat birokrasi pemerintah di Kota Malang yaitu rumitnya proses tindak lanjut pengaduan tersebut dan tidak adanya petugas informasi yang bertugas untuk mempermudah kinerja antar SKPD (Satuan Kerja Perangkat Daerah) berikutnya.

Berdasarkan uraian penelitian Prasetya, dkk. (2013), terdapat beberapa kesamaan dengan penelitian yang dilakukan oleh peneliti yaitu pengelolaan pengaduan sama-sama menggunakan media *website*, serta memiliki fungsi yang

sama yang akan diimplementasikan peneliti juga yaitu proses tindak lanjut yang dilakukan oleh instansi tertentu bisa dilihat oleh statusnya oleh masyarakat yang dalam sisi peneliti yaitu instansi sebagai prodi atau fakultas dan masyarakat sebagai mahasiswanya. Namun terdapat juga perbedaan antara penelitian Prasetya, dkk. (2013) dengan peneliti yaitu tidak adanya hambatan birokrasi dalam proses penanganan pengaduan berupa keluhan yang disampaikan oleh mahasiswa.

Dari ketiga penelitian sebelumnya, dapat disimpulkan bahwa fungsi pembangunan sistem penanganan keluhan bertujuan untuk mempermudah pengguna (masyarakat atau mahasiswa) dalam menyampaikan aspirasinya dengan lugas dan dapat ditindak lanjuti oleh pihak yang bersangkutan dengan pengelolaan yang terpantau dan terstruktur.

Berdasarkan kesimpulan dari ketiga penelitian diatas, maka untuk melengkapi kekurangan penelitian sebelumnya peneliti menambahkan beberapa fitur-fitur sebagai berikut :

1. Sistem dapat menghasilkan keluaran (*output*) data pengaduan berupa data berbentuk *pdf* yang dapat langsung dikelola, untuk mempermudah fakultas atau prodi dalam menindaklanjuti data laporan pengaduan yang telah dibuat mahasiswa.
2. Sistem dapat memberikan respon terhadap data pengaduan yang telah dibuat mahasiswa dengan mengubah status dari data pengaduan tersebut sebagai status terkini terhadap data pengaduan tersebut.
3. Mahasiswa dapat mengetahui status terkini terhadap data pengaduan yang telah dibuat sebelumnya.

Dalam penelitian yang berjudul Pengembangan Sistem Penanganan Keluhan Berbasis Web (Studi Kasus : Fakultas Teknik Universitas Muhammadiyah Yogyakarta) ini, sistem yang dibangun guna untuk menampung dan mengelola seluruh aspirasi mahasiswa berupa pengaduan atau keluhan maupun saran dan kritik pada Fakultas Teknik Universitas

Muhammadiyah Yogyakarta. Selain itu, sistem ini dibangun bertujuan untuk memberikan kemudahan dalam proses pengolahan data pengaduan dan verifikasi tindakan dari prodi atau fakultas terhadap pengaduan yang diberikan oleh mahasiswa. Sistem ini dibangun dengan menggunakan *Framework Codeigniter* yang memudahkan peneliti membuat aplikasi web menjadi cepat dan mudah dibandingkan harus membuatnya dari awal. Keunggulannya adalah karena performa yang cepat dalam mengeksekusi data yang dibuat, dan mudah digunakan bagi programmer yang pemula. Selain itu, sistem ini juga dibuat menggunakan *Framework Front-End Bootstrap, Library JQuery* dan penggunaan *MySql* yang mempermudah pembuatan sistem ini.

Melalui sistem ini, mahasiswa bisa menyampaikan aspirasi mereka berupa keluhan atau pengaduan, kritik dan saran. Selain itu, mahasiswa juga bisa menentukan kepada siapa aspirasi mereka ditujukan. Untuk bisa menggunakan website ini, mahasiswa akan mendapatkan akses *Nim* dan *Password* mereka yang telah ditetapkan oleh Fakultas Teknik Universitas Muhammadiyah Yogyakarta.

## **2.2. Landasan Teori**

### **2.2.1. Pengertian Sistem**

Sistem adalah sekumpulan subsistem yang saling berhubungan, berkumpul bersama, saling bekerja sama untuk mencapai tujuan tertentu. Suatu sistem dapat terdiri dari sub-sistem. Subsistem tersebut dapat pula terdiri dari beberapa subsistem yang lebih kecil. (Bahra , Al, 2005)

Secara lugas, sistem dapat diterjemahkan sebagai sekumpulan atau himpunan dari bagian atau komponen yang terorganisir yang saling berinteraksi satu sama lainnya. Sistem adalah kelompok dari lebih komponen atau subsistem yang saling berhubungan yang berfungsi dengan tujuan yang sama. (James A, Hall, 2007:6) Sistem terdiri dari bagian atau komponen seperti pengolahan (*pocessing*), serta keluaran (*output*). (Scoot, 1996)

Dari definisi diatas, dapat dirinci lebih lanjut pengertian dari sistem secara umum, yaitu:

1. Setiap sistem terdiri dari unsur-unsur atau komponen. (Putra dan Febrian, 2017)
2. Unsur-unsur atau komponen tersebut saling berkaitan satu dengan yang lainnya. (Putra dan Febrian, 2017)
3. Unsur sistem tersebut bekerja sama untuk mencapai tujuan sistem tersebut dan suatu sistem merupakan bagian dari sistem lain yang lebih besar. (Putra dan Febrian, 2017)

a. Karakteristik Sistem

Suatu sistem mempunyai beberapa karakteristik atau sifat-sifat tertentu yang dapat membedakan suatu sistem dengan sistem lainnya. Berikut adalah karakteristik suatu sistem (Jogiyanto, 1995):

1. Komponen Sistem (*Component System*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi atau saling bekerjasama untuk membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagianbagian dari sistem.

2. Batasan Sistem (*Boundary*)

Batasan sistem (boundary) merupakan daerah yang membatasi antara satu sistem dengan sistem lainnya atau dengan lingkungan luarnya. Batas suatu sistem menunjukkan ruang lingkup (scope) dari sistem tersebut.

### 3. Lingkungan Luar Sistem (*Environment*)

Lingkungan luar sistem merupakan apapun yang berada diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan merugikan sistem. Lingkungan luar yang menguntungkan merupakan kekuatan untuk sistem yang harus tetap dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan sistem harus segera dikendalikan dan dicegah agar tidak mengganggu kelangsungan sistem.

### 4. Penghubung Sistem (*Interface*)

Penghubung sistem merupakan media penghubung antara satu sub sistem dengan yang lainnya dan berintegrasi membentuk satu kesatuan. Melalui penghubung ini memungkinkan sumber- sumber daya mengalir dari satu sub sistem ke sub sistem lainnya.

### 5. Masukan Sistem (*Input*)

Masukan sistem atau input adalah energi yang dimasukan kedalam sistem. Masukan tersebut dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk mendapatkan keluaran.

### 6. Keluaran Sistem (*Output*)

Keluaran sistem (*output*) merupakan hasil dari energi yang diolah dan diklasifikasikan menjadi

keluaran yang bermanfaat. Keluaran dapat berupa masukan untuk subsistem lain.

#### 7. Pengolah Sistem (*Process*)

Merupakan suatu bagian pengolahan yang akan merubah masukan menjadi keluaran yang diinginkan.

#### 8. Sasaran Sistem (*Object*)

Sasaran sistem merupakan tujuan yang akan dicapai oleh suatu sistem, jika sistem tidak memiliki sasaran maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan masukan yang di butuhkan sistem dan keluaran yang dihasilkan sistem. Suatu sistem bisa dikatakan berhasil apabila mengena pada sasaran dan tujuan.

### **2.2.2. Pengertian Informasi**

Informasi adalah sekumpulan data yang telah diolah, dibentuk atau dimanipulasi sesuai dengan kebutuhan tertentu. (Amsyah, Drs Zulfikar, 2003:2) Informasi artinya data telah dibentuk menjadi suatu yang mempunyai arti dan berguna bagi manusia. (Laudon, Kenneth C, 2008:16, Laudon, Jane P, 2008:16)

### **2.2.3. Pengertian Sistem Informasi**

Secara luas, sistem informasi dapat diartikan sebagai sebuah sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak tertentu atas laporan yang diperlukan tersebut. Sistem informasi didefinisikan sebagai sekumpulan komponen yang saling berhubungan, mengumpulkan atau mendapatkan, memproses, menyimpan

dan mendistribusikan informasi untuk menunjang pengambilan keputusan dan pengawasan dalam suatu organisasi. (Laudon, Kenneth C, 2008:15, Laudon, Jane P, 2008:15)

#### **2.2.4. Pengertian Penanganan Keluhan (*Complaint Handling*)**

Setiap organisasi yang berorientasi pada pelanggan (*customer oriented*) perlu menyediakan kesempatan dan akses yang mudah serta nyaman bagi para pelanggannya guna menyampaikan saran, kritik, pendapat dan keluhan mereka. (Tjiptono, 2005:210)

Keluhan adalah salah satu bagian dari ekspresi negatif yang dihasilkan karena ketidaksesuaian kenyataan dengan keinginan seseorang. (James, 2006:20) Keluhan adalah suatu pernyataan atau ungkapan rasa kurang puas terhadap suatu produk atau layanan, baik secara lisan maupun tertulis dari pelanggan internal maupun eksternal.

Manfaat prosedur penanganan keluhan antara lain:

- a. Tersedia prosedur yang jelas ketika terjadi keluhan.
- b. Menciptakan pemahaman dan keyakinan cara menangani keluhan.
- c. Membantu mengatasi rasa “bersalah” secara pribadi bagi orang yang menangani keluhan.
- d. Menerima keluhan sebagai umpan balik yang berharga, bukan sebagai kritik.
- e. Menghasilkan catatan yang dapat digunakan untuk menganalisa kemungkinan peningkatan layanan.

Disisi lain, keluhan berguna sebagai alat kendali atau evaluasi terhadap pemberian kualitas pelayanan yang selama ini diberikan kepada masyarakat yang dalam penelitian ini adalah mahasiswa. Namun disisi lain, merupakan suatu hal yang penting untuk diperhatikan karena menjadikan keluhan sebagai masalah yang perlu dicari solusinya. Penangan terhadap

munculnya keluhan harus ditanggapi secara rasional karena untuk membedakan jenis keluhan yang berupa masalah serius dan bukan masalah serius.

## 2.3. Teknologi Pengembangan Aplikasi

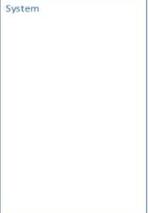
### 2.3.1. *Unified Modeling Language (UML)*

*Unified Modeling Language (UML)* merupakan bahasa pemodelan secara grafis yang digunakan untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan seluruh rancangan aplikasi perangkat lunak. Penggunaan model UML bertujuan untuk mengidentifikasi bagianbagian yang termasuk dalam lingkup sistem di dalam aplikasi. Model UML yang dipakai dalam pengembangan aplikasi Penanganan Keluhan Berbasis Web antara lain adalah Use Case Diagram, Activity Diagram, Class Diagram, ER Diagram. Bagian dari UML antara lain:

#### a. *Use Case Diagram*

*Use Case Diagram* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem yang dibuat. Dapat dikatakan *Use Case Diagram* digunakan untuk mengetahui fungsi yang ada di dalam sistem dan siapa saja yang berhak menggunakan fungsi-fungsi yang dibutuhkan. Simbol-simbol yang digunakan dalam *use case diagram* dapat dilihat pada Tabel 2.1.

Tabel 2.1 Simbol *Use Case Diagram*

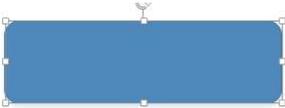
NO	SIMBOL	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.
2		<i>Generalization</i>	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor).
3		<i>Include</i>	Menspesifikasikan bahwa use case sumber secara eksplisit.
4		<i>Extend</i>	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.
5		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.

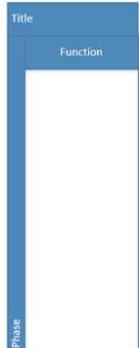
NO	SIMBOL	NAMA	KETERANGAN
7		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

b. *Activity Diagram*

*Activity Diagram* merupakan diagram yang digunakan untuk menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada Tabel 2.2.

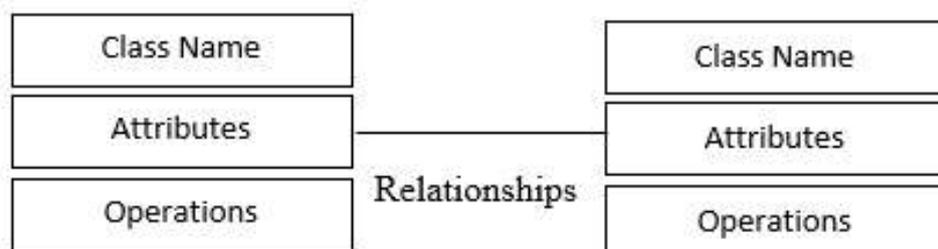
**Tabel 2.2** Simbol *Activity Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1		<i>Start Point</i>	Merupakan awal dalam aktifitas.
2		<i>End Point</i>	Merupakan akhir dalam aktifitas.
3		<i>Activities</i>	Menggambarkan suatu proses atau kegiatan bisnis
4		<i>Decision Point</i>	Menggambarkan pilihan untuk pengambilan keputusan dalam aktifitas.

NO	SIMBOL	NAMA	KETERANGAN
			
5		<i>Swimlane</i>	Digunakan untuk pembagian activity diagram yang menunjukkan siapa yang melakukan aktifitas.

c. *Class Diagram*

*Class Diagram* merupakan diagram yang digunakan untuk menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class Diagram* secara unik meliputi: *Nama Kelas (Class Name)*, *Atribut (Attributes)*, *Operasi(Operations)*, dan *Relasi(Relationships)* dapat juga dilihat pada Gambar 2.1.



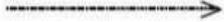
**Gambar 2.1** Struktur *Class Diagram*

Atribut dan operation (metoda) dapat memiliki salah satu sifat berikut:

1. Private, hanya bisa dipanggil dari dlm kelas itu sendiri. Methode atau atribut diawali “-“.
2. Protected, hanya dapat dipanggil oleh class yang bersangkutan dan class turunannya. metode diawali dg tanda “#”.
3. Public, dapat dipanggil dari semua objek. Methode atau atribut diawali tanda “+”

Simbol-simbol yang digunakan dalam *Class Diagram* dapat dilihat pada Tabel 2.3.

**Tabel 2.3** Simbol *Class Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1	 asosiasi / <i>association</i>	<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>Multiplicity</i>
2	 asosiasi berarah / <i>directed association</i>	<i>Directed Association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>Multiplicity</i>
3	 generalisasi	<i>Generalization</i>	Relasi antar kelas dengan generalisasi-spesialisasi (umum khusus)
4	 kebergantungan / <i>dependency</i>	<i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas

NO	SIMBOL	NAMA	KETERANGAN
5	 agregasi / aggregation	Aggregation	Relasi antar kelas dengan makna semua-bagian ( <i>Whole-Part</i> )

- **Relasi Generalisasi** digunakan dalam hubungan antara kelas induk dengan kelas turunan (inherited).
- **Relasi Agregasi** digunakan ketika satu kelas dibentuk (terdiri dari) dari kelas kelas lain.
- **Relationship Multiplicity** menunjukkan jumlah suatu objek yang bisa berhubungan dengan objek lain.

Penggunaan Nilai Kardinalitas yang digunakan dalam *Class Diagram* dapat dilihat pada Gambar 2.2.

Nilai Kardinalitas	Arti	Contoh	
 0..1	Nol atau satu	karyawan	0..1 istri
1	Hanya satu	negara	1 presiden
0..*	Nol atau lebih	karyawan	0..* anak
1..*	Satu atau lebih	bos	1..* bawahan
n	Hanya n (dengan n > 1)	karyawan	n cek up
0..n	Nol sampai n (dengan n > 1)	karyawan	0..n sim
1..n	Satu sampai n (dengan n > 1)	kereta api	1..n gerbong

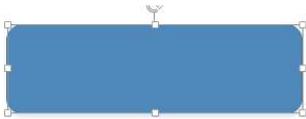
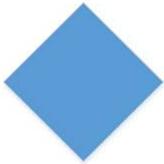
**Gambar 2.2** Simbol Nilai Kardinalitas *Class Diagram*

d. *Entity Relationship Diagram (ERD)*

*Entity Relationship Diagram (ERD)* merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD digunakan untuk memodelkan

struktur data dan hubungan antar data (expresiku, n.d.), untuk menggambarannya digunakan beberapa notasi dan simbol seperti pada Tabel 2.4.

**Tabel 2.4** Simbol Entity Relationship Diagram (ERD)

NO	SIMBOL	NAMA	KETERANGAN
1		<i>Entitas</i>	Suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
3		<i>Relasi</i>	Menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.
4		<i>Atribut</i>	Berfungsi mendeskripsikan karakter entitas (atribut yang berfungsi sebagai key diberi garis bawah).
6		<i>Association</i>	Sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut.

## 2.4. Model Pengembangan Aplikasi

### 2.4.1. SDLC (Software Development Life Cycle)

Dalam alur penelitian, metode yang digunakan adalah model SDLC (*Software Development Life Cycle*). SDLC (*Software*

*Development Life Cycle*) merupakan kerangka yang menggambarkan beberapa kegiatan yang dilakukan melalui beberapa tahap dalam pembuatan sebuah *software* (Fatta, 2007). Selain itu, SDLC juga penting untuk proses *maintenance* pada system *software* yang dikembangkan.

Model SDLC yang dipakai dalam pengembangan aplikasi adalah model *Waterfall*. Menurut (Sommerville, 2011), model *Waterfall* adalah sebuah contoh dari proses perencanaan dimana semua proses kegiatan harus terlebih dahulu direncanakan dan dijadwalkan sebelum dikerjakan. *Waterfall Model* atau *Classic Life Cycle* merupakan model yang paling banyak dipakai dalam *Software Engineering (SE)*. Disebut *Waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.

## **2.5. Pengembangan Sistem Berbasis Web**

Pengembangan sistem berbasis *web* adalah aplikasi yang sejak awal dirancang untuk dieksekusi di lingkungan berbasis *web*. Definisi ini mengungkapkan dua aspek penting dari aplikasi ini (Simarmata, 2009) sebagai berikut:

1. Suatu aplikasi *web* dirancang agar dapat berjalan di dalam lingkungan berbasis *web*. Artinya, aspek-aspek *hipermedia* dalam kaitannya dengan *hiperteks* dan multimedia di dalam kombinasi dengan kelola aplikasi tradisional harus diperhitungkan di seluruh hidup aplikasi.
2. Aplikasi *web* adalah suatu aplikasi yang tidak hanya berupa sekumpulan halaman-halaman *web*.

## 2.6. Konsep Basis Data (*Database*)

### 2.6.1. Definisi Basis Data (*Database*)

Basis Data (*Database*) adalah sekumpulan data yang saling berhubungan secara logis dan fungsi data-data tersebut dirancang untuk memenuhi informasi yang dibutuhkan oleh suatu organisasi (Indrajati, 2009). Yang artinya, Basis Data (*Database*) adalah sebuah penyimpanan data yang dalam jumlah banyak atau besar yang digunakan oleh banyak pengguna.

Sistem *Database* memiliki empat komponen penting (Indrajati, 2009) yaitu:

1. **Data** adalah informasi yang disimpan dalam suatu struktur tertentu yang terintegrasi.
2. **Hardware** adalah perangkat keras berupa komputer dengan media penyimpanan sekunder yang digunakan untuk menyimpan data.
3. **Software** adalah perangkat lunak yang digunakan untuk melakukan pengelolaan data. Perangkat lunak ini sering disebut sebagai Database Management System (DBMS) yaitu sekumpulan komponen untuk menetapkan, membangun, dan menggerakkan suatu database.
4. **User** adalah orang yang menggunakan data yang tersimpan dan dikelola. User dapat berupa seorang yang mengelola database tersebut, yang disebut dengan database administrator (dba), bisa juga end user yang mengambil hasil dari pengelolaan data base melalui bahasa query.

### 2.6.2. Definisi *MySQL*

*MySQL* (*My Structure Query Language*) adalah sebuah program pembuat *database* yang bersifat *open source* (Bunafit

Nugroho, 2004). Dengan bahasa perintah terstruktur yang distandarisasikan untuk semua program pengakses database seperti Oracle, Postgres, SQL, SQL Server, dll.

Dalam penggunaannya SQL dikategorikan menjadi tiga sub perintah, yaitu DDL (*Data Definition Language*), DML (*Data Manipulation Language*) dan DCL (*Data Control Language*).

#### 1. Pengertian DDL (*Data Definition Language*)

Pengertian dari DDL (*Data Definition Language*) adalah perintah SQL yang digunakan untuk mendefinisikan, menciptakan, dan menghapus sebuah database ataupun tabel. Yang termasuk dalam perintah DDL meliputi *CREATE*, *DROP*, dan *ALTER* (Emma Utami& Sukrisno, 2008).

- a. *Create* : berfungsi untuk membuat data, termasuk diantaranya membuat database baru, tabel baru, tampilan baru dan kolom.
- b. *Alter* : berfungsi untuk mengubah struktur tabel yang telah dibuat. Tugasnya mencakup mengganti nama tabel, menambah kolom, mengubah kolom, menghapus kolom, maupun memberikan atribut pada kolom.
- c. *Drop* : berfungsi untuk menghapus database dan tabel.

#### 2. Pengertian DML (*Data Manipulation Language*)

Data Manipulation Language (*DML*) merupakan bahasa untuk memanipulasi/ mengubah isi tabel. Perintah yang digunakan diantaranya adalah (Emma Utami & Sukrisno, 2008)

- a. *Insert* : berfungsi untuk menyisipkan atau memasukkan data baru ke dalam tabel. Penggunaannya setelah database dan tabel dibuat.

- b. *Update* : berfungsi untuk mengubah atau memperbaharui data lama menjadi data baru.
- c. *Delete* : berfungsi untuk menghapus data dari tabel. Pada saat penghapusan data, perintah yang telah dijalankan tidak dapat digagalkan, sehingga data yang telah terhapus tidak dapat dikembalikan lagi.

### 3. Pengertian DCL (*Data Control Language*)

*Data Control Language* (DCL) merupakan bahasa yang berhubungan dengan pengendalian akses ke data base. Perintah DCL diantaranya adalah (Emma Utami & Sukrisno, 2008) :

- a. *Grand* : berfungsi untuk memberikan hak / izin akses oleh *administrator* (pemilik utama) server kepada user (pengguna biasa). Hak akses tersebut berupa hak membuat (*CREATE*), mengambil (*SELECT*), menghapus (*DELETE*), mengubah (*UPDATE*) dan hak khusus berkenaan dengan sistem *database*.
- b. *Revoke* : perintah *revoke* memiliki kegunaan terbalik dengan *GRAND*, yaitu untuk menghilangkan atau mencabut hak akses yang telah diberikan kepada user oleh *administrator*.

## 2.7. Bahasa Pemrograman dan *Framework*

### 2.7.1. HTML (*Hypertext Markup Language*)

HTML (*Hypertext Markup Language*) adalah bahasa pemrograman yang dirancang untuk memungkinkan pembuatan situs web. Situs Web ini kemudian dapat dilihat oleh orang lain yang terhubung ke jaringan internet. HTML sendiri merupakan bahasa pemrograman yang fleksibel, karena dapat menampung bahasa

pemrograman lain, seperti JAVA, Visual Basic, Bahasa C, dan lain-lain.

Untuk membuat sebuah situs web, maka diperlukan *design* menggunakan HTML. Untuk membuat halaman situs web menggunakan HTML maka diperlukan beberapa perintah yang dibuat dalam sebuah file dokumen dengan ekstensi *.html*. Perintah untuk membuat halaman dalam bentuk HTML terdapat pada Tabel 2.5.

**Tabel 2.5** Perintah pada HTML

<i>Tag</i>	<i>Fungsi Tag / Lokasi Tag</i>
<code>&lt;html&gt;&lt;/html&gt;</code>	Diawal dan diakhir code (start dan stop)
<code>&lt;head&gt;&lt;/head&gt;</code>	Letakan <code>&lt;title&gt;</code> dan <code>&lt;/title&gt;</code> disekitarnya
<code>&lt;title&gt;&lt;/title&gt;</code>	Letakan judul/Teks disekitarnya
<code>&lt;body&gt;&lt;/body&gt;</code>	Setelah meletakkan <code>&lt;title&gt;</code> dan <code>&lt;/title&gt;&lt;/head&gt;</code> letakan <code>&lt;body&gt;</code> dan <code>&lt;/body&gt;</code>
<code>&lt;h1&gt;&lt;/h1&gt;</code> <code>&lt;h2&gt;&lt;/h2&gt;</code> sampai <code>&lt;h6&gt;&lt;/h6&gt;</code>	Membuat tulisan dengan format header
<code>&lt;b&gt;&lt;/b&gt;</code>	Tulisan bold (tebal)
<code>&lt;i&gt;&lt;/i&gt;</code>	Tulisan italic (miring)
<code>&lt;p&gt;&lt;/p&gt;</code>	Tulisan Paragraf-break
<code>&lt;br&gt;</code>	Membuat baris baru (break)
<code>&lt;hr&gt;</code>	Membuat garis horizontal
<code>&lt;a&gt;&lt;/a&gt;</code>	Membuat hyperlink text
<code>&lt;img src=""file"&gt;</code>	Meletakkan gambar

### 2.7.2. PHP (Hypertext Processor)

PHP (*Hypertext Preprocessor*) merupakan bahasa standar yang digunakan dalam dunia website yang mempunyai bahasa

program berbentuk script yang disisipkan pada halaman HTML dan diletakkan didalam server web (Bunafit Nugroho, 2004). PHP (*Hypertext Preprocessor*) adalah script pemrograman yang terletak dan dieksekusi di server. Salah satunya adalah untuk menerima, mengelola, dan menampilkan data dari dan ke seluruh situs. Data akan diolah ke sebuah database server (pemrograman database yang terletak di sisi server) untuk memudahkan hasilnya ditampilkan di browser sebuah situs (Madcoms, 2011). PHP memungkinkan untuk membuat sebuah halaman situs web yang bersifat dinamis.

### **2.7.3. Framework Codeigniter**

Codeigniter merupakan sebuah framework open source yang menggunakan bahasa pemrograman *PHP* dengan menerapkan metode MVC (Model-View-Controller). Yang mana masing-masing dari MVC tersebut memiliki peran masing-masing.

### **2.7.4. MySql (Database)**

MySql merupakan open source program yang digunakan sebagai database yang juga merupakan salah satu software database server yang paling sering digunakan. MySql bisa dijalankan pada berbagai platform seperti windows dan linux (Madcoms, 2011).

### **2.7.5. JQuery dan Javascript**

JQuery merupakan *library* JavaScript menyederhanakan dokumen HTML, membuat *event*, *animasi*, dan *interaksi Ajax* guna mengembangkan web yang dapat diakses lebih cepat. JQuery dibuat untuk mengubah penulisan JavaScript. Dengan JQuery, tampilan website akan menjadi menarik, interaktif dan *load* akses website yang juga menjadi lebih cepat. Sedangkan Javascript digunakan untuk memanipulasi DOM (*Document Object Model*).

### 2.7.6. CSS (Cascading Stylesheets)

CSS (*Cascading Stylesheets*) berfungsi untuk mengatur *style* element yang ada pada halaman website, mulai dari format teks sampai dengan layout halamannya. Penggunaan dari CSS adalah untuk memperoleh konsistensi *style* pada element tertentu.

### 2.7.7. BOOTSTRAP (Framework Front-End)

BOOTSTRAP merupakan kerangka kerja *front-end* secara gratis untuk pengembangan *web* yang lebih cepat dan mudah. BOOTSTRAP mencakup template desain yang berbasis *HTML* dan *CSS* untuk *tipografi*, bentuk *tombol*, *tabel*, *navigasi*, *modals*, *korsel gambar*, *plugin JavaScript opsional* dan masih banyak lagi. BOOTSTRAP juga memungkinkan untuk membuat desain *responsive* dengan mudah dan cepat.

## 2.8. Perangkat Lunak yang Digunakan

### 2.8.1. XAMPP

XAMPP merupakan perangkat lunak bebas yang didukung oleh banyak sistem informasi, yang merupakan kompilasi dari beberapa program. XAMPP adalah gabungan dari *Apache*, *PHP*, dan *MySql* yang memberikan interface yang lebih mudah dan praktis (Mubarok, 2011). XAMPP digunakan sebagai *server* yang mampu berdiri sendiri (*localhost*) dan penerjemahannya menggunakan bahasa pemrograman. XAMPP memberi kebebasan untuk dikembangkan lebih lanjut.

### 2.8.2. VSCode (Visual Studio Code)

VSCode (*Visual Studio Code*) merupakan alat pengembangan lintas platform yang ringan dan cepat dengan fitur tipe-IDE untuk membangun aplikasi web. VSCode berkerja dengan *Node* dan *ASP.NET v5*.

VSCode juga merupakan editor kode efisien dengan dukungan untuk operasi pengembangan seperti *debugging*, *task running* dan *version control*. VSCode bertujuan untuk menyediakan alat yang dibutuhkan pengembang untuk siklus cepat dalam membangun kode-debug dan memberikan alur kerja yang lebih kompleks ke IDE berfitur lengkap.

### 2.8.3. PhpMyAdmin

PhpMyAdmin merupakan sebuah software gratis yang ditulis dalam *PHP*, yang bertujuan untuk menangani administrasi *MySQL* dengan melalui sebuah Web. PhpMyAdmin didukung oleh berbagai operasi pada *MySQL* dan *MariaDB*. Pengoperasian yang sering dilakukan yaitu mengelola basis data, kolom, tabel, relasi, dll yang bersangkutan yang dapat digunakan dengan menggunakan perintah *SQL*.

## 2.9. Pengujian Perangkat Lunak

### 2.9.1. Black Box Testing

Dalam pengujian perangkat lunak ada dua jenis model pengujian yaitu *white box testing* dan *black box testing*. Dari kedua metode itu, yang digunakan pada skripsi adalah *black box testing* dengan alasan peneliti hanya menguji apakah fungsionalitas dalam aplikasi sudah sesuai dengan yang diharapkan.

Menurut (Roger S, 2002), *black box testing* berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan engineer untuk memperoleh input yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program. *Black box testing* berusaha untuk menemukan kesalahan dalam kategori berikut:

1. Fungsi yang tidak benar atau fungsi yang hilang.
2. Kesalahan antarmuka.
3. Kesalahan dalam struktur data atau akses database eksternal.
4. Kesalahan kinerja.
5. Kesalahan inisialisasi dan pemutusan kesalahan.