

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian sebelumnya dilakukan oleh Ahmad Lubis Ghozali, Kurnia Adi Cahyanto, Widayanti tahun 2020 [5] membahas tentang perancangan dan implementasi sistem informasi berbasis web. Sistem tersebut digunakan untuk mempermudah proses pendaftaran anggota NU secara online melalui web. Tujuannya yaitu agar data anggota dapat terpusat, dan memudahkan pengurus dalam mengelola data anggota setiap cabangnya. Metodologi yang digunakan dalam sistem ini adalah metodologi *waterfall* sistem ini dibangun dengan menggunakan bahasa pemrograman PHP. Pengujian yang dilakukan pada penelitian ini menggunakan metode *black-box testing* untuk menemukan kesalahan fungsi pada program.

Pada penelitian selanjutnya dilakukan oleh Muhammad Fajri Ramadhan dan Lusi Fajarita 2018 [6] dengan judul Pembuatan Aplikasi Sistem Informasi Administrasi Surat Menyurat Studi Kasus: Kesekretariatan Badan Eksekutif Mahasiswa Universitas Budi Luhur. Sistem ini digunakan untuk pencatatan administrasi sangat diperlukan, karena untuk membantu para pengurus untuk melakukan pengecekan jikalau terjadi masalah dalam administrasi. Aplikasi yang dibuat adalah berbasis desktop dengan bertujuan untuk dapat mempercepat pembuatan laporan dan operasional yang sedang berjalan agar menjadi lebih efektif dan lebih efisien.

Penelitian lainnya juga dilakukan Sandy Putra Effendi dan Elfi Tasrif 2019 [2] mengenai perancangan digitalisasi dalam pelayanan administrasi akademik. Sistem ini menunjukkan bahwa pentingnya digitalisasi dalam pelayanan administrasi. Tujuan perancangan aplikasi ini adalah untuk meningkatkan aspek layanan administrasi jurusan dan memberikan kepuasan terhadap mahasiswa di perguruan tinggi khususnya jurusan Teknik Elektronika Fakultas Teknik Universitas Negeri Padang. Metode yang digunakan pada perancangan aplikasi ini adalah metode *waterfall* yang terdiri dari analisis kebutuhan, perancangan, implementasi dan pengujian. Hasil akhir dari penelitian ini yaitu aplikasi berbasis android.

Penelitian lainnya dilakukan oleh Budi Supriadi dan Lukman Bachtiar 2018 [7]. Sistem ini digunakan untuk mengelola surat. Metode-metode Pengembangan Sistem Perangkat Lunak untuk menganalisis sistem ini serta sebuah web editor Adobe Dreamweaver CS6, Bahasa pemrograman Javascript, CSS, HTML, PHP dan MySQL. Selain itu juga memudahkan sekretaris dalam mengelola surat, mengkategorikan surat, setting jabatan dan pegawai, membuat laporan surat masuk dan keluar. Hasil akhir dari penelitian ini yaitu aplikasi berbasis website.

Perbedaan dari penelitian yang dilakukan adalah dalam membuat Sistem Informasi Digitalisasi Administrasi Organisasi Pelajar Nahdlatul Ulama yaitu adanya fitur monitoring kegiatan dan keaktifan setiap Pimpinan Ranting. Sistem ini menggunakan metode pengembangan dengan model *prototype*. Model *prototype* digunakan agar pengguna dapat melihat dan melakukan eksperimen sejak awal proses pengembangan. Sistem ini dikembangkan menggunakan bahasa pemrograman PHP dan MYSQL sebagai perancangan *database* dan *management* data dengan *framework laravel*. Hasil sistem ini berupa aplikasi berbasis *website* yang mudah digunakan oleh siapa saja dan dimana saja.

2.2 Landasan Teori

Dalam menunjang penelitian ini, maka diperlukannya teori-teori mendasar. Teori-teori yang digunakan dalam penelitian ini adalah:

2.2.1 Sistem

Sistem adalah suatu jaringan kerja dari prosedur - prosedur yang saling berhubungan, berkumpul bersama - sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu [7]. Sistem informasi merupakan salah satu faktor yang mempengaruhi keberhasilan suatu organisasi. Dikarenakan sistem informasi yang akan terus berkembang, maka sebuah organisasi sangat memerlukan perencanaan strategis dalam sistem informasi [8]. Suatu sistem dibuat untuk menyelesaikan sesuatu sasaran dalam organisasi digunakan dengan memperhatikan kebutuhan-kebutuhan yang ada oleh organisasi.

2.2.2 Digitalisasi

Digitalisasi adalah sebuah istilah atau terminologi yang digunakan untuk menjelaskan sebuah proses peralihan media. Peralihan yang dimaksud adalah peralihan dari tercetak, video, maupun audio menjadi sebuah digital [2]. Digitalisasi juga dapat diartikan sebagai proses konversi dari media/informasi yang tercetak, ditulis dan/atau digambar ke dalam

bentuk atau format digital [3]. Tujuan melakukan digitalisasi adalah agar bisa membuat sebuah arsip dokumen dalam bentuk digital. Digitalisasi arsip dapat dilakukan dengan cara mengalihmediakan arsip dari konvensional menjadi arsip digital. Digitalisasi arsip dapat menjadi solusi atas kelemahan arsip konvensional. Manfaat digitalisasi arsip diantaranya untuk menghindari kerusakan arsip atau kehilangan arsip akibat bencana alam.

2.2.3 Administrasi

Administrasi merujuk pada kegiatan atau usaha untuk membantu, melayani, mengarahkan atau mengatur semua kegiatan di dalam mencapai suatu tujuan [2]. Sistem informasi administrasi adalah kumpulan dari komponen-komponen atau elemen-elemen yang saling berhubungan untuk melakukan proses pencatatan, pengaturan, pengalokasian suatu kegiatan untuk mencapai tujuan tertentu dengan menggunakan sarana perlengkapan dan peralatan yang ada [7].

2.2.4 Pelajar Nahdlatul Ulama

Pelajar adalah individu atau sekelompok orang yang ikut dalam kegiatan belajar untuk mendapatkan ilmu pengetahuan. Setiap pelajar perlu mempunyai kualiti disiplin ilmu yang relevan, akhlak dan penetapan minda yang sesuai, tingkah laku dan beradab untuk memajukan diri bagi membolehkan mereka memberikan sumbangan kepada keharmonian dan kemajuan keluarga, masyarakat, negara dan komuniti global [9]. Nahdlatul Ulama merupakan organisasi masyarakat yang besar dan cakupannya luas [5]. Pelajar Nahdlatul Ulama yaitu para pelajar yang masuk dalam struktural organisasi Nahdlatul Ulama yang biasa disebut dengan IPNU IPPNU. Organisasi IPNU IPPNU adalah organisasi yang merupakan bagian dari badan otonom Nahdlatul Ulama yang memiliki susunan kepengurusan masing-masing disetiap periodenya [10]. Badan otonom Nahdlatul Ulama yang paling muda adalah IPNU IPPNU dibandingkan badan otonom lainnya karena rentan usianya itu adalah usia remaja.

2.2.5 PHP

PHP atau singkatan dari *Personal Home Page* merupakan bahasa skrip yang tertanam dalam HTML untuk dieksekusi bersifat server side. PHP bersifat *server-side* maka user tidak akan melihat kode-kode khusus terkait password atau akun yang merupakan data- data penting server sehingga server web lebih terjamin dalam hal keamanan [4].

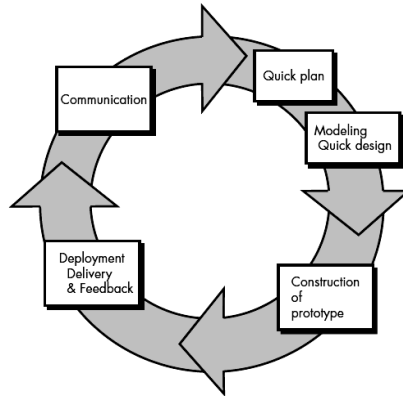
2.2.6 Rekayasa Perangkat Lunak

Pengertian rekayasa adalah pemakaian yang dilakukan untuk menyelesaikan masalah praktis dari tidak ada menjadi ada. Perangkat lunak adalah program komputer yang dieksekusi memberikan fungsi dan unjuk kerja seperti yang diinginkan. Perangkat lunak diartikan sebagai struktur data yang memungkinkan program memanipulasi informasi secara proporsional. Perangkat lunak juga dapat diartikan sebagai dokumen yang menggambarkan operasi dan kegunaan program [11]. Jadi dapat disimpulkan bahwa rekayasa perangkat lunak merupakan pemakaian sekelompok item atau objek yang membentuk konfigurasi di dalamnya yang terdapat program, data atau informasi, dan dokumen.

Perangkat lunak yang sangat umum digunakan salah satunya adalah *website*. *Website* adalah kumpulan dari halaman-halaman situs, yang biasanya terangkum dalam sebuah domain atau subdomain, yang tempatnya berada di dalam *World Wide Web (WWW)* di Internet [5]. Penyebaran informasi melalui *website* sangat cepat dan mencakup area yang luas serta tidak dibatasi oleh jarak dan waktu. Oleh sebab itu, *website* merupakan sarana penting untuk mendapatkan dan mengelola informasi. Berikut ini metode dan *tools* pembangunan rekayasa perangkat lunak yang digunakan:

A. Metode Pengembangan Sistem

Pengembangan sistem yang digunakan dalam penelitian ini yaitu metode *prototype*. *Prototype* adalah sebuah metode pengembangan software yang banyak digunakan pengembang agar dapat saling berinteraksi dengan pelanggan selama proses pembuatan sistem [4]. *Prototype* didefinisikan sebagai alat yang memberikan idebagi pembuat maupun pemakai potensial tentang cara sistem berfungsi dalam bentuk lengkapnya, dan proses untuk menghasilkan sebuah *prototype* disebut *prototyping* [12]. Metode *prototype* digunakan dimaksudkan untuk mendapatkan representasi dari pemodelan aplikasi yang akan dibuat. Rancangan aplikasi awal mulanya berbentuk *mockup* selanjutnya akan dievaluasi oleh pengguna. Setelah *mockup* dievaluasi pengguna tahap selanjutnya *mockup* menjadi bahan rujukan bagi pengembang software untuk merancang aplikasi.



Gambar 2.1 Metode Prototype

Berdasarkan model *prototype* yang telah digambarkan diatas, maka dapat diuraikan pembahasan disetiap tahapannya sebagai berikut:

1. *Communication* / Komunikasi
Pada tahap awal, penulis melakukan komunikasi atau pertemuan dengan para *stakeholder* untuk menentukan kebutuhan dari sistem yang saat itu diketahui dan untuk menggambarkan sistem yang akan dibuat.
2. *Quick Plan* / Perencanaan Secara Cepat
Dalam tahapan ini, penulis melakukan penentuan sumberdaya, spesifikasi untuk pengembangan berdasarkan kebutuhan sistem dan tujuan berdasarkan pada hasil komunikasi yang dilakukan agar pengembangan dapat sesuai dengan yang diharapkan.
3. *Modeling Quick Design* / Model Rancangan Cepat
Pada tahap ini memodelkan perencanaan dengan menggunakan tools UML yaitu *Usecase* untuk mendefinisikan fungsi dari sistem. Dalam tahap ini, prototype yang dibangun dengan sistem rancangan sementara kemudian di evaluasi oleh pengguna apakah sudah sesuai dengan yang diinginkan atau masih perlu untuk di evaluasi kembali. Setelah sistem dianggap sesuai dengan apa yang diharapkan, langkah berikutnya yaitu pembuatan aplikasi (pengkodean).

4. *Construction of Prototype / Kontruksi Prototype*
Tahapan ini digunakan untuk membangun prototype dan menguji-coba sistem yang dibangun. Proses instalasi dan penyediaan *user support* juga dilakukan agar sistem dapat berjalan dengan sesuai.
5. *Deployment Deliery & Feedback / Penyerahan Dan Memberikan Umpan Balik Terhadap Pengembangan Prototype* kemudian diserahkan kepada para pengguna untuk mengevaluasi *prototype* yang telah dibuat sebelumnya dan memberikan umpan-balik yang akan digunakan untuk memperbaiki spesifikasi kebutuhan. Langkah terjadi saat pengembang melakukan perbaikan terhadap *prototype* tersebut.

B. Metode Pengujian Sistem

Metode pengujian perangkat lunak merupakan hal yang sangat dibutuhkan untuk membuktikan apakah sistem yang diimplementasikan telah memenuhi spesifikasi dan rancangan yang sudah direncanakan sebelumnya. Ada beberapa jenis pengujian perangkat lunak, antara lain [13]:

- 1) Pengujian *white box* adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara prosedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Secara sekilas dapat diambil kesimpulan *white box testing* merupakan petunjuk untuk mendapatkan program yang benar secara 100%,
- 2) *Black-Box Testing* merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetestan pada spesifikasi fungsional program pengujian program dalam penelitian ini yaitu menggunakan pengujian *black-box*. Pengujian *black-bok* berfokus pada persyaratan pengujian perangkat lunak.

Terkait dengan pengujian sistem, pengujian yang digunakan adalah pengujian *black-box*. *Black box testing* cenderung dapat menemukan beberapa hal seperti fungsional yang tidak benar atau tidak ada, kesalahan struktur data, kesalahan akses basis data, kesalahan antar muka, kesalahan performance serta kesalahan inialisasi dan terminasi. Keunggulan menggunakan metode *black-box testing* adalah [14]:




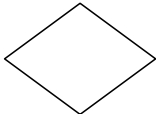

- 1) Penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu;
- 2) Pengujian dilakukan dari sudut pandang pengguna, hal ini membantu untuk mengungkapkan ambiguitas atau inkonsistensi dalam spesifikasi persyaratan;
- 3) Programmer dan tester keduanya saling bergantung satu sama lain.

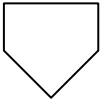
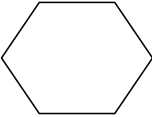
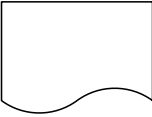
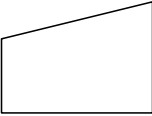
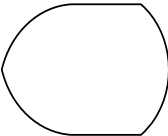
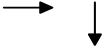
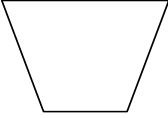
C. *Tools/Alat Bantu Penelitian*

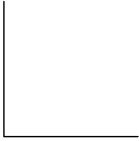

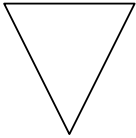
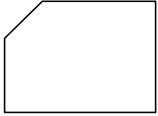
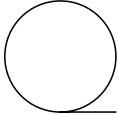
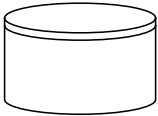
1. *Flowchart System*

Flowchart adalah sebuah diagram yang menjelaskan alur proses dari sebuah program. *Flowchart* merupakan penggambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program. Diagram *flowchart* merupakan sistem standar yang sudah banyak digunakan untuk membangun suatu sistem atau aplikasi [15]. Dalam membangun sebuah program, *flowchart* berperan penting untuk menerjemahkan proses berjalannya sebuah program agar lebih mudah untuk dipahami.

Tabel 2.1 *Simbol Flowchart System*

No	Simbol	Nama	Keterangan
1.		<i>Terminal</i>	Awal atau akhir sebuah program
2.		<i>Output/Input</i>	Proses input atau output
3.		Proses	Proses operasional didalam komputer
4.		<i>Decision</i>	Suatu kondisi tertentu mengarah pada dua pilihan kemungkinan ya/tidak.
5.		<i>Connector</i>	Penghubung antar proses pada halaman yang sama

6.		<i>Offline Connector</i>	Penghubung antar proses pada halaman yang lain
7.		<i>Predefined Proses</i>	Simbol yang menunjukkan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
8.		<i>Document</i>	Simbol yang menunjukkan untuk mencetak laporan ke printer.
9.		<i>Manual input</i>	Simbol yang menunjukkan proses input yang dilakukan secara manual menggunakan online keyboard.
10.		<i>Display</i>	Simbol yang menunjukkan adanya <i>output</i> yang digunakan berupa layar (video, computer).
11.		<i>Flow Direction</i>	Simbol yang berfungsi untuk menyatakan jalannya alur sebuah proses
12.		<i>Manual</i>	Simbol yang menunjukkan proses diluar komputer atau kegiatan tanpa penggunaan komputer

13.		<i>Communication Link</i>	Simbol yang berfungsi untuk menyatakan bahwa adanya transisi suatu data/informasi dari satu lokasi ke lokasi lainnya
14.		<i>Keying Operation</i>	Simbol yang berfungsi untuk menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai keyboard
15.		Offline Storage	Simbol yang menunjukkan bahwa data dalam Simbol ini akan disimpan ke suatu media tertentu.
16.		<i>Punched Card</i>	Simbol untuk menyatakan input berasal dari kartu atau output ditulis ke kartu.
17.		<i>Magnetic-tape unit</i>	Simbol untuk menyatakan input berasal dari pita magnetic atau output disimpan ke pita magnetic.
18.		<i>Disk Storage</i>	Simbol untuk menyatakan input berasal dari disk atau input disimpan ke disk.



2. UML

Unified Modeling Language (UML) adalah sekumpulan aturan yang digunakan untuk menspesifikasikan atau mendeskripsikan sebuah sistem perangkat lunak berdasarkan obyek [7]. Pemodelan (modeling) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami [5]. Ada macam-macam diagram dalam Unified Modeling Diagram (UML) yaitu:

a. Usecase Diagram

Usecase adalah sebuah kegiatan atau interaksi yang saling berkaitan antara aktor dan sistem. *Usecase diagram* merupakan interaksi yang dilakukan oleh actor, untuk mengangkat suatu peristiwa atau kebutuhan fungsional dalam sistem yang bersangkutan [16]. Atau secara umum, dapat diartikan sebagai sebuah teknik yang dimanfaatkan untuk pengembangan perangkat lunak (software), guna mengetahui kebutuhan fungsional dari sistem tersebut [6].

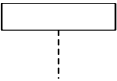
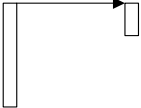

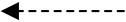

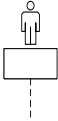
Tabel 2.2 Use Case Diagram

No	Gambar	Nama	Keterangan
1.		<i>Actor</i>	Menspesifikasikan himpunan tokoh yang pengguna mainkan saat berhubungan menggunakan <i>use case</i> .
2.	-- <<include>> -->	<i>Include</i>	<i>Use case</i> merupakan sumber secara eksplisit.
3.	-- <<extend>> -->	<i>Extend</i>	<i>Use case</i> memperluas perilaku sumber pada titik yang diberikan.
4.	————	<i>Association</i>	Berguna untuk menghubungkan objek yang satu dengan objek yang lain
5.		<i>Use Case</i>	Deskripsi berdasarkan urutan aksi-aksi yang ditampilkan sistem yang membuat suatu output terukur bagi sebuah aktor

b. Sequence Diagram

Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi diantara obyek-obyek tersebut[7]. *Sequence diagram* menggambarkan kelakuan/prilaku objek pada use case dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antara objek[17].

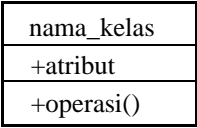



Tabel 2.3 *Sequence Diagram*

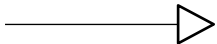


No	Gambar	Nama	Keterangan
1.		<i>LifeLine</i>	Objek antarmuka yang saling berhubungan.
2.		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat berita-berita tentang aktivitas yang terjadi.
3.		<i>Message (Return)</i>	Penggambaran hasil dari pengiriman pesan.
4.		<i>Message (return)</i>	Menyatakan arah kembali pesan.
5.		<i>Activation</i>	Menyatakan objek dalam keadaan aktif.
6.		<i>Actor</i>	Menspesifikasikan himpunan peran pengguna ketika berinteraksi dengan sistem.

c. **Class Diagram**

Class diagram merupakan penjelasan visual dari struktur database yang diinginkan dibuat, yang terdiri dari *class-class* untuk membangun sebuah sistem [16].

Tabel 2.4 *Class Diagram*

No	Bentuk Simbol	Nama Simbol	Fungsi Simbol
1.		Kelas	Kelas pada struktur sistem
2.		Antarmuka/ <i>interface</i>	Sama dengan konsep interface dalam pemrograman berorientasiobjek
3.		Asosiasi/ <i>association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya jugadisertai dengan multiplicity
4.		Asosiasi/ berarah/ directed association	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya jugadisertai dengan multiplicity

5.		Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.		Kebergantungan/dependency	Relasi antar kelas dengan kebergantungan antar kelas
7.		Agregasi/aggregation	Relasi antar kelas dengan makna semua bagian (whole-part)

2.2.7 Pemrograman Berbasis Objek

Pemrograman berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis [18]. Dalam Pemrograman Berorientasi Objek (PBO), terdapat beberapa konsep penting yang membentuk dasar dari pemahaman dan pengembangan perangkat lunak. Beberapa konsep tersebut termasuk:

1. Class (Kelas):
Class adalah kerangka dasar atau blueprint yang mendefinisikan struktur, atribut (data), dan metode (operasi) yang dapat dimiliki oleh objek. Class memberikan panduan tentang bagaimana objek dari kelas tersebut akan diciptakan dan berinteraksi.
2. Object (Objek):
Objek merupakan instance (instansi) konkret dari suatu kelas. Objek memiliki atribut yang sesuai dengan definisi kelasnya dan dapat melakukan operasi atau metode yang didefinisikan dalam kelas tersebut.
3. Inheritance (Pewarisan):
Pewarisan adalah konsep yang memungkinkan pembuatan kelas baru berdasarkan kelas yang sudah ada (superclass atau parent

class). Kelas yang baru ini akan mewarisi atribut dan metode dari kelas yang lebih umum. Ini membantu dalam pembuatan hierarki kelas yang memungkinkan penggunaan kembali kode yang ada.

4. *Encapsulation* (Enkapsulasi):
Enkapsulasi adalah konsep untuk mengatur akses ke atribut dan metode dalam sebuah kelas. Dengan enkapsulasi, atribut tertentu bisa diakses hanya oleh metode yang sesuai, sehingga mencegah manipulasi langsung dari luar kelas dan memastikan integritas data.
5. *Abstraction* (Abstraksi):
Abstraksi mengacu pada kemampuan untuk menyembunyikan detail kompleks dari suatu objek dan hanya menyoroti fitur yang penting. Dalam konteks PBO, abstraksi membantu dalam pembuatan kelas yang menggambarkan karakteristik umum, sementara rincian implementasinya disembunyikan.
6. *Polymorphism* (Polimorfisme):
Polimorfisme mengacu pada kemampuan objek yang berbeda untuk merespons metode dengan nama yang sama. Ini memungkinkan berbagai objek untuk memiliki perilaku yang berbeda dalam situasi yang sama.
7. *Method* (Metode):
Metode adalah fungsi atau operasi yang dapat dilakukan oleh objek dalam sebuah kelas. Metode menggambarkan apa yang objek tersebut bisa lakukan dan bagaimana ia berinteraksi dengan data lain atau objek lain.
8. *Instance Variable* (Variabel Instansi):
Variabel instansi adalah variabel yang dideklarasikan di dalam suatu kelas dan menyimpan nilai yang unik untuk setiap objek yang diciptakan dari kelas tersebut.

2.2.8 *Framework Laravel*

Framework adalah komponen pemrograman yang siap re-use (bisa digunakan ulang) kapan saja, sehingga programmer tidak harus membuat skrip yang sama untuk tugas yang sama. Misalkan programmer ingin halaman-halaman web menampilkan data dengan paginasi (paging) halaman, framework telah menyediakan fungsi paging tersebut sedangkan programmer cukup menggunakan fungsi tersebut pada saat coding, tetapi tentu dengan kaidah- kaidah yang ditetapkan oleh masing - masing framework [19].

Laravel adalah sebuah framework web berbasis PHP yang open-source dan tidak berbayar, diciptakan oleh Taylor Otwell dan diperuntukkan untuk pengembangan aplikasi web yang menggunakan pola MVC. Struktur pola MVC pada laravel sedikit berbeda pada struktur pola MVC pada umumnya. Di laravel terdapat routing yang menjembatani antara request dari user dan controller. Jadi controller tidak langsung menerima request tersebut. Laravel merupakan framework yang mudah dipelajari, serta memiliki dokumentasi yang cukup lengkap. Laravel menyediakan fungsi berupa penyimpanan kata sandi, keamanan, pengingat password, mengatur ulang password, enkripsi dan validasi data [20].

2.2.9 Basis Data

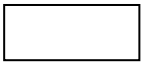
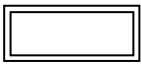
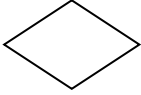
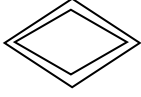
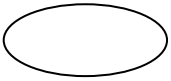
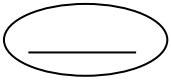
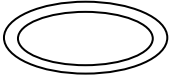
Basis Data adalah kumpulan informasi yang disimpan didalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Basis Data adalah representasi kumpulan fakta yang saling berhubungan disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan [21]. Perangkat lunak yang dapat digunakan mengelola basis data disebut *database management* sistem atau disingkat DBSM. DBMS merupakan antar muka antara pengguna basis data (baik pengguna DBMS langsung maupun aplikasi) dengan data yang disimpan. RDBMS atau *relationship database manajemen* sistem adalah salah satu jenis DBMS yang mendukung hubungan antar tabel salah satunya adalah MySQL [22].

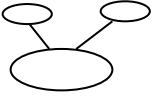

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL [10].

Model data yang digunakan dalam penelitian ini adalah *Entity Relationship Diagram* (ERD). ERD merupakan suatu model data berupa notasi grafis dalam pemodelan data konseptual yang menggambarkan hubungan antara penyimpanan. Model data tersebut merupakan sekumpulan cara, peralatan untuk mendeskripsikan data-data yang berhubungan dengan satu sama lain, semantic serta batasan konsistensi [16]. ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD untuk memodelkan struktur data dan hubungan

antar data, untuk menggambarannya digunakan beberapa notasi dan simbol diantaranya sebagai berikut [2]:

Tabel 2. 5 Simbol ERD

NO	BENTUK SIMBOL	NAMA SIMBOL	FUNGSI SIMBOL
1		<i>Entity</i>	Menunjukkan suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
2		<i>Weak Entity</i>	Menunjukkan keberadaan dari <i>entity</i> yang tergantung dari keberadaan <i>entity</i> lain.
3		<i>Relationship</i>	Menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda
4		<i>Identifying Relationship</i>	Merupakan induk dari <i>Relationship</i>
5		<i>Atribut</i>	Berfungsi mendeskripsikan karakter entitas (atribut yang berfungsi sebagai key diberi garis bawah)
6		<i>Atribut Primary key</i>	<i>Atribut</i> yang menentukan <i>entity</i> secara unik.
7		<i>Atribut Multivalue</i>	Memiliki sekelompok nilai setiap <i>instant entity</i>

8		<i>Atribut Composit</i>	Terdiri dari atribut lebih kecil yang memiliki arti tertentu.
9		Garis	Menunjukkan penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut.

~Halaman Ini Sengaja Dikosongkan~