

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Penelitian Terdahulu**

Adapun penelitian terkait yang akan digunakan sebagai sumber acuan yang relevan dan terkini yaitu:

1. Berdasarkan penelitian yang dilakukan oleh Heru Setiawan (2017) dengan judul “Perancangan Sistem Informasi Manajemen Proyek : Sistem Informasi Kontraktor” Perancangan sistem informasi manajemen proyek diharapkan dapat menjadi salah satu acuan dan sumber solusi untuk permasalahan tersebut. Oleh karena itu, penelitian ini dirancang dengan menggunakan metode *Waterfall* yang terdiri dari beberapa tahap yaitu : tahap analisa data, tahap perencanaan, tahap implementasi dan tahap pengujian. Konsep perancangan menggunakan metode *object oriented* dengan menggunakan UML (*Unified Modelling Language*), yang terdiri dari : *Use Case Diagram*, *Activity Diagram* dan *Sequence Diagram*. Hasil dari penelitian ini adalah berupa rancangan sistem informasi manajemen proyek yang disesuaikan dengan kebutuhan prosedur serta pemberkasan yang dibutuhkan.
2. Berdasarkan penelitian yang dilakukan oleh Gino Gabriel (2016) dengan judul “Rancang Bangun Sistem Informasi Manajemen Proyek Konstruksi Berbasis Web Pada Ud Surya Agung” Saat ini sistem pengerjaan proyek mulai dari perhitungan rencana anggaran biaya, penjadwal kerja, dan pencatatan setiap informasi pembayaran pada UD. Surya Agung masih

menggunakan cara manual berupa pencatatan pada *microsoft word*. Selain itu untuk memperoleh informasi pengerjaan proyek di luar kota maupun di luar pulau hanya bisa melalui telepon atau langsung meninjau ke lokasi langsung. Berdasarkan latar belakang permasalahan pada UD. Surya Agung, dibuatlah sebuah sistem informasi manajemen proyek berbasis *web* yang dapat mencatat setiap informasi dalam sebuah proyek untuk mengatasi permasalahan tersebut.

3. Berdasarkan penelitian oleh penulis bertujuan untuk membangun dan merancang sistem yang baru yaitu mengenai Manajemen Proyek Kontraktor Pada CV. Buana Pilar Jaya Mandiri. Sistem ini bertujuan untuk membantu perusahaan dalam mengatasi masalah biaya yang dikeluarkan selama proses proyek kontraktor terjadi, sehingga laporan proyek dapat diperoleh dengan cepat tanpa membutuhkan waktu yang cukup lama. Sistem yang akan dirancang menggunakan aplikasi PHP dan Mysql.

## **II.2. Sistem Informasi**

Sistem informasi adalah sekumpulan komponen-komponen sistem yang berada didalam suatu ruang lingkup organisasi, saling berinteraksi untuk menghasilkan sebuah informasi yang bertujuan untuk pihak manajemen tertentu dan untuk mencapai tujuan tertentu. Faktor-faktor yang menentukan kehandalan dari suatu sistem informasi atau informasi dapat dikatakan baik jika memenuhi kriteria-kriteria sebagai berikut :

a) Keunggulan (*Usefulness*)

Yaitu suatu sistem yang harus dapat menghasilkan informasi yang tepat dan relevan untuk mengambil keputusan manajemen dan personil operasi dalam organisasi.

b) Ekonomis

Kemampuan sistem yang mempengaruhi sistem harus bernilai manfaat minimal, sebesar biayanya.

c) Keandalan (*Reliability*)

Keluaran dari sistem harus mempunyai tingkat ketelitian tinggi dan sistem tersebut harus beroperasi secara efektif.

d) Pelayanan (*Customer Service*)

Yakni suatu sistem memberikan pelayanan yang baik dan efisien kepada para pengguna sistem pada saat berhubungan dengan organisasi.

e) Kapasitas (*Capacity*)

Setiap sistem harus mempunyai kapasitas yang memadai untuk menangani setiap periode sesuai yang dibutuhkan.

f) Sederhana dalam kemudahan (*Simplicity*)

Sistem tersebut lebih sederhana (umum) sehingga struktur dan operasinya dapat dengan mudah dimengerti dan prosedur mudah diikuti.

g) Fleksibel (*Flexibility*)

Sistem informasi ini harus dapat digunakan dalam kondisi yang bagaimana yang diinginkan oleh organisasi tersebut atau pengguna tertentu.

## h) Komponen Sistem Informasi

Istilah dalam komponen sistem informasi adalah blok bangunan (*building block*) yang dapat di bagi menjadi enam blok yaitu :

### a. Blok masukan (*Input block*)

Blok *input* merupakan data–data yang masuk ke dalam sistem informasi, yang dapat berupa *document-document* dasar yang dapat diolah menjadi suatu informasi tertentu.

### b. Blok model (*Model block*)

Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan mengolah data *input* untuk menghasilkan suatu informasi yang dibutuhkan.

### c. Blok keluaran (*Output block*)

Merupakan informasi yang menghasilkan sekumpulan data yang nantinya akan disimpan berupa data cetak laporan.

### d. Blok teknologi (*Technologi block*)

Blok teknologi merupakan penunjang utama dalam berlangsungnya sistem informasi. Yang memiliki beberapa komponen yaitu diantaranya alat memasukkan data (*input device*), alat untuk menyimpan dan mengakses data (*storege device*), alat untuk menghasilkan dan mengirimkan keluaran (*output divice*) dan alat untuk membentuk pengendalian sistem secara keseluruhan (*control device*). Teknologi informasi terdiri dari 3 (tiga) bagian utama, yaitu teknisi (*humanware* atau *braiware*), perangkat lunak (*software*), dan perangkat keras (*hardware*).

e. Blok basis data (*Database block*)

Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Data perlu di simpan dan perlu di organisasi sedemikian rupa, supaya informasi yang dihasilkan berkualitas.

f. Blok kendali (*Control block*)

Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat di cegah bila terlanjur terjadi. (I Made Budi Adnyana, 2016)

### **II.3. Manajemen Proyek**

Manajemen proyek adalah usaha kegiatan untuk meraih sasaran yang telah didefinisikan dan ditentukan dengan jelas seefisien dan seefektif mungkin. Dalam rangka meraih sasaran-sasaran yang telah disepakati, diperlukan sumber-sumber daya (*resources*) termasuk sumber daya manusia yang merupakan kunci dari segalanya. Tujuan utama dari manajemen proyek adalah agar proyek dapat dilaksanakan dengan efisien, tepat waktu, dan mencapai hasil yang diinginkan. Sering terjadi pada sebuah proyek yang berlarut pada pekerjaannya sehingga pada akhirnya harus mengalami penjadwalan ulang. Oleh karena itu, peran perencanaan dalam suatu proyek sangat penting, segala sesuatu harus dimulai dari rencana dan harus disepakati bersama antara para *stakeholder* yang terlibat pada proyek. (Heru Setiawan : 2017)

## II.4. Database

Pangkalan data atau basis data (*database*) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (*query*) basis data disebut sistem manajemen basis data (*database management system*, DBMS). Sistem basis data dipelajari dalam ilmu informasi. (Neni Purwati dan Hendra Kurniawan, 2015 : 50).

Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya: penjelasan ini disebut skema. Model yang umum digunakan sekarang adalah model relasional, yang mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom. Model yang lain seperti model hierarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel (Neni Purwati dan Hendra Kurniawan, 2015 : 50).

## II.5. Normalisasi

Normalisasi adalah proses penyusunan table-table yang tidak redundan (*double*) yang dapat menyebabkan anomali pada saat terjadi manipulasi data seperti tambah, ubah dan hapus. Tujuan dari normalisasi adalah:

- a. Untuk menghilangkan kerangkapan data.
- b. Untuk mengurangi, kompleksitas.

c. Untuk mempermudah pemodifikasian data. (Herlina Trisnawati : 2016)

Normalisasi adalah suatu proses untuk membuat data yang tidak normal menjadi data yang normal. Bentuk data yang tidak normal / data mentah biasa disebut juga *unnormalized form*. Masing – masing level normalisasi mempunyai aturan tersendiri.

1. *First Normal Form*

Suatu tabel dikatakan dalam keadaan *first normal form* (1NF) jika :

- a. Tidak ada perulangan record data dalam tabel.
- b. Setiap sel memiliki satu nilai saja. Artinya tidak ada perulangan group dan array.
- c. Data yang diinputkan memiliki tipe data yang sama dengan tipe data kolom dalam tabel.

2. *Second Normal Form*

Suatu tabel dikatakan dalam keadaan *Second Normal Form* (2NF) jika tabel tersebut sudah dalam keadaan *First Normal Form* (1NF) dan jika semua atribut yang bukan kunci tabel, baik *primary key* maupun *foreign key* tergantung pada semua kunci dalam tabel.

3. *Third Normal Form*

Suatu tabel dikatakan dalam keadaan *third normal form* (3NF) jika tabel tersebut sudah dalam keadaan *second normal form* (2NF) dan jika tidak terdapat ketergantungan yang transitif. Artinya, data-data yang mungkin diisi berulang-ulang dapat dibuat sebuah tabel baru.

4. *Boyce-Codd Normal Form (BCNF)*

Tabel dikatakan dalam keadaan *boyce-codd normal form* (BCNF) jika tabel tersebut dalam keadaan *third normal form* (3NF) dan setiap determinan adalah kunci kandidat.

5. *Fourth Normal Form (4NF)*

Suatu tabel dikatakan dalam keadaan *fourth normal form* (4NF) jika tabel tersebut dalam keadaan *boyce-codd normal form* (BCNF) dan jika tidak terdapat ketergantungan nilai ganda.

6. *Fiveth Normal Form (5NF)*

Tabel dikatakan dalam keadaan *Fiveth Normal Form* (5NF) jika tabel tersebut dalam keadaan *fourth normal form* (4NF) dan jika setiap ketergantungan dalam join ada pada tabel sudah konsekuen dengan kunci kandidat pada tabel tersebut (Ema Utami ; 2012 : 73-76)

## II.6. PHP

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah web dan bisa digunakan pada dokumen HTML. PHP dirancang untuk dapat bekerja sama dengan database server dan dibuat sedemikian rupa sehingga pembuatan dokumen HTML yang dapat mengakses database menjadi begitu mudah. Tujuan dari bahasa scripting ini adalah untuk membuat aplikasi dimana aplikasi tersebut yang dibangun oleh PHP pada umumnya akan

memberikan hasil pada web browser, tetapi proses secara keseluruhan dijalankan di server. (Saipul Anwar : 2016)

## II.7. MySql

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Kehandalan suatu sistem basisdata (DBMS) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, MySQL mendukung operasi basisdata transaksional maupun operasi basisdata non-transaksional. Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basisdata kompetitor lainnya. Namun pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi blogging berbasis web (wordpress), CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk

bisnis sangat disarankan untuk menggunakan modus basisdata transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional.

MySQL pada awalnya diciptakan pada tahun 1979, oleh Michael "Monty" Widenius, seorang programmer komputer asal Swedia. Monty mengembangkan sebuah sistem database sederhana yang dinamakan UNIREG yang menggunakan koneksi low-level ISAM database engine dengan indexing. Pada saat itu Monty bekerja pada perusahaan bernama TcX di Swedia. TcX pada tahun 1994 mulai mengembangkan aplikasi berbasis web, dan berencana menggunakan UNIREG sebagai sistem database. Namun sayangnya, UNIREG dianggap tidak cocok untuk database yang dinamis seperti web. (Herpendi : 2016)

## **II.8. UML (*Unified Modelling Language*)**

*Unified Modelling Language* (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak sebuah sistem. UML lebih mengedepankan penggunaan diagram untuk menggambarkan aspek dari sistem, karena tergolong bahasa visual yang lebih mudah dan lebih cepat dipahami dibandingkan dengan bahasa pemrograman. *Unified Modelling Language* (UML) biasa digunakan untuk :

1. Menggambarkan batasan sistem dan fungsi-fungsi sistem secara umum, dibuat dengan *use case* dan actor.

2. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagrams*.
3. Menggambarkan representasi struktur static sebuah sistem dalam bentuk *class diagram*.
4. Membuat model *behavior* yang menggambarkan kebiasaan atau sifat sebuah sistem dengan *state transition diagrams* UML.
5. Menyatakan arsitektur implementasi fisik menggunakan *component and development diagrams*.
6. Menyampaikan atau memperluas *fungsi* dengan *stereo types*.

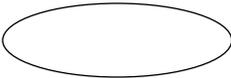
Pemodelan penggunaan UML merupakan metode pemodelan berorientasi objek dan berbasis visual. Karenanya pemodelan objek yang fokus pada pendefinisian struktur statis dan model sistem informasi yang dinamis daripada mendefinisikan data dan model proses yang tujuannya adalah pengembangan tradisional. UML menawarkan diagram yang dikelompokkan menjadi lima perspektif berbeda untuk memodelkan suatu sistem. Seperti satu set *blue print* yang digunakan untuk membangun sebuah rumah (Saipul Anwar, et al., 2016 : 75-76).

### **II.8.1. Use Case Diagram**

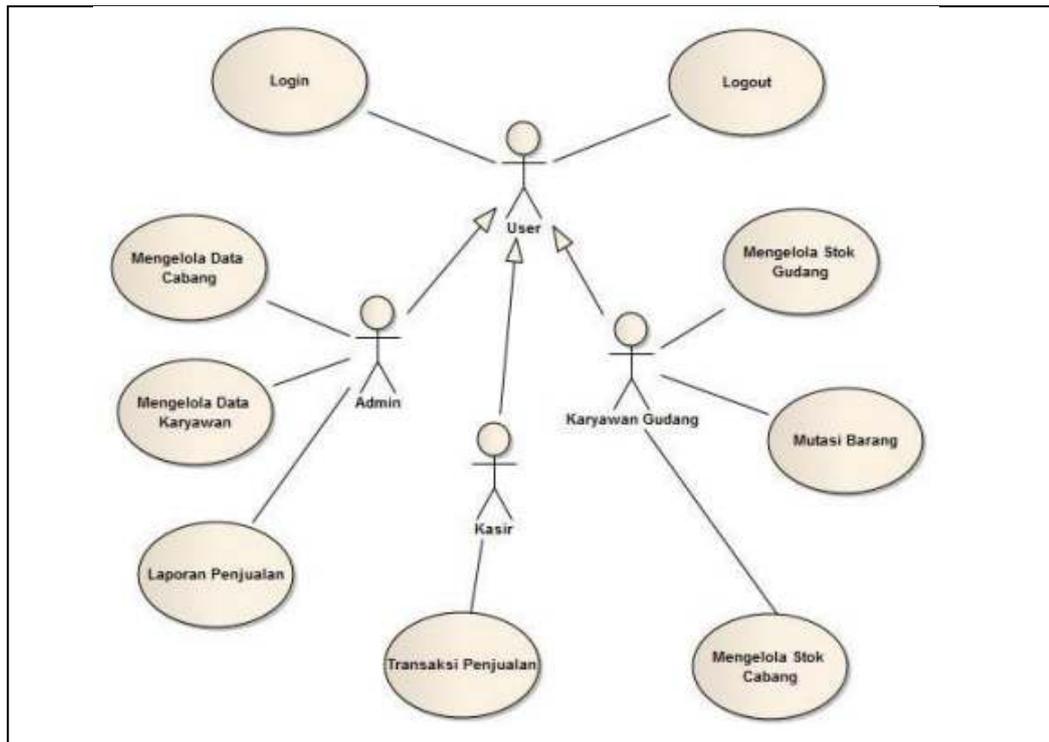
*Use case* diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Ade Hendini, 2016 : 108).

Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.1 dibawah ini :

**Tabel II.1. Simbol Use Case**

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidkasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain ( <i>required</i> ) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Sumber : Ade Hendini, 2016)



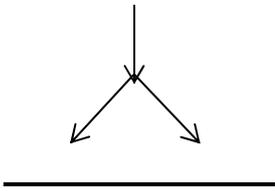
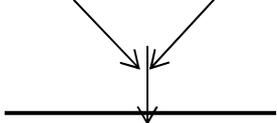
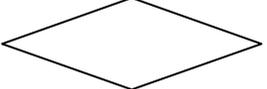
**Gambar II.1. Use Case Diagram**  
(Sumber : Ade Hendini, 2016 : 112)

## II.8.2. Activity Diagram

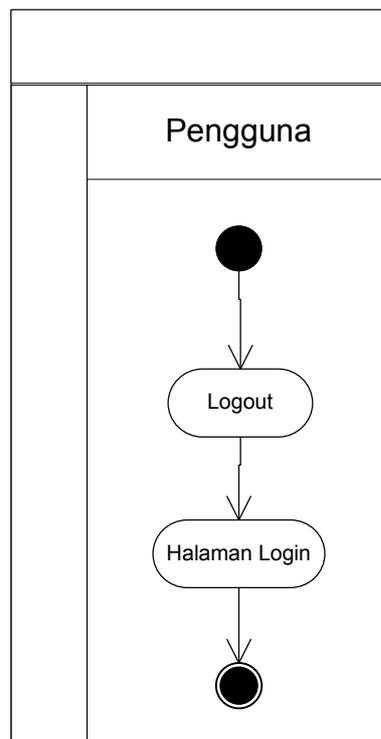
*Activity* diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.2 dibawah ini:

**Tabel II.2. Simbol *Activity Diagram***

Gambar	Keterangan
●	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
◎	<i>End point</i> , akhir aktifitas.

	<p><i>Activites</i>, menggambarkan suatu proses/kegiatan bisnis.</p>
	<p><i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan paralel menjadi satu.</p>
	<p><i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.</p>
	<p><i>Decision Points</i>, menggambarkan pilihan untuk pengambilan keputusan, <i>true</i>, <i>false</i>.</p>
<p>New Swimlane</p>	<p><i>Swimlane</i>, pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.</p>

(Sumber : Ade Hendini, 2016 : 109)

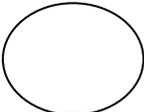
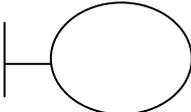
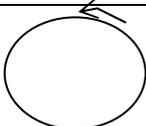
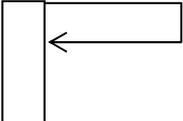


**Gambar II.2. Activity Diagram**  
(Sumber : Ade Hendini, 2016 : 112)

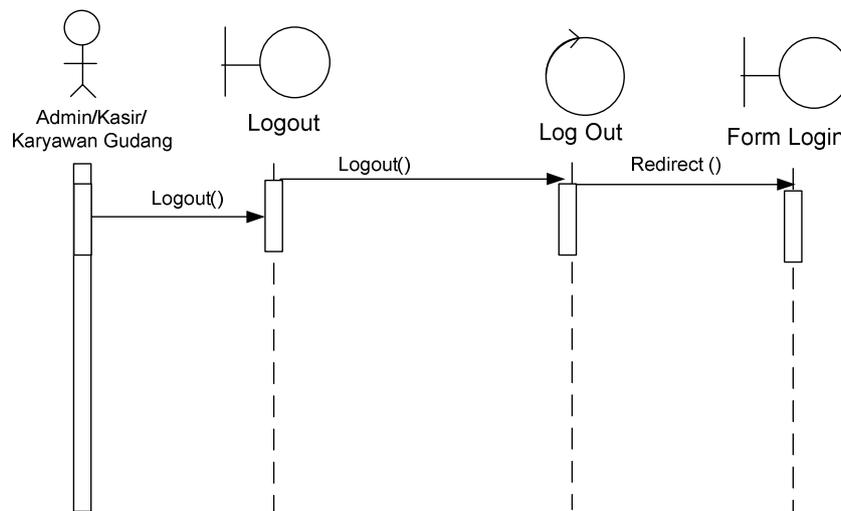
### II.8.3. Sequence Diagram

*Sequence* diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.3 dibawah ini :

**Tabel II.3. Simbol *Sequence Diagram***

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

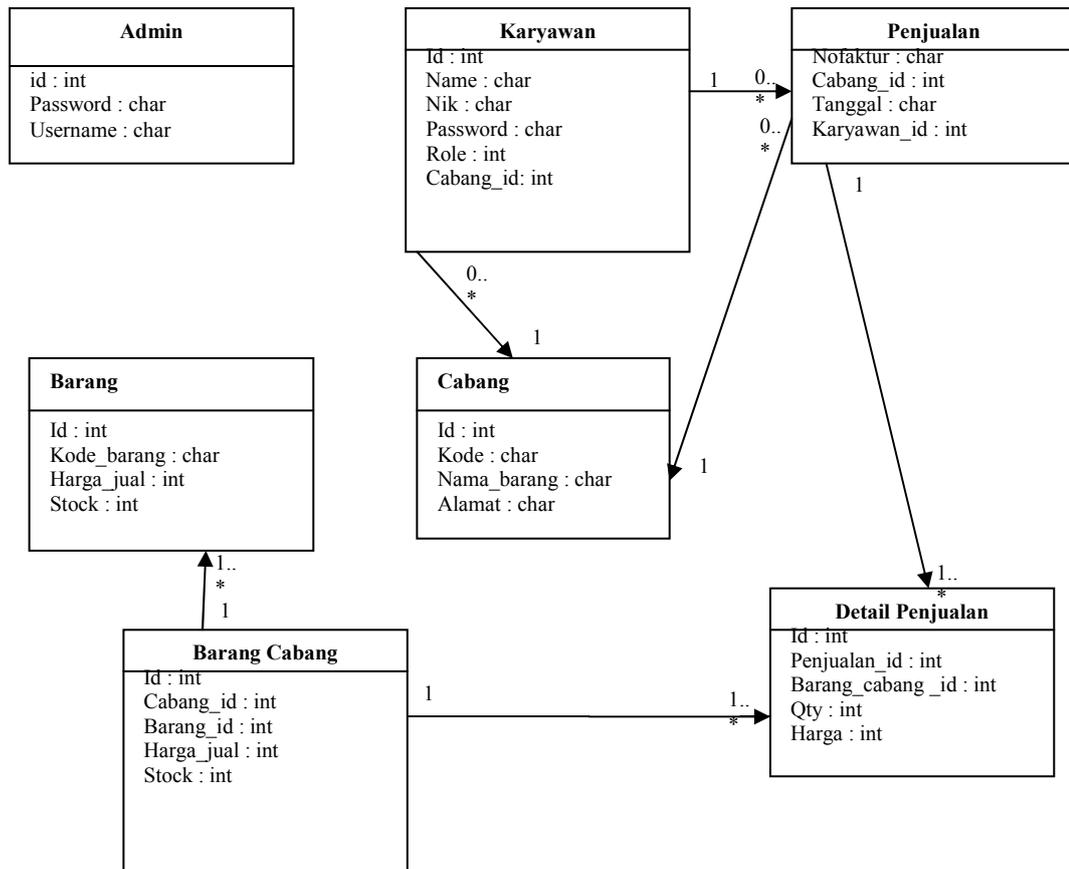
(Sumber : Ade Hendini, 2016 : 110)



**Gambar II.3. Sequence Diagram**  
(Sumber : Ade Hendini, 2016 : 114)

#### II.8.4. Class Diagram

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class* diagram secara khas meliputi : Kelas (*Class*), Relasi *Associations*, *Generalitation* dan *Aggregation*, atribut (*Attributes*), operasi (*operation/ method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality* (Ade Hendini, 2016 : 111).



**Gambar II.4. Class Diagram**  
*(Sumber : Ade Hendini, 2016 : 115)*