

## **BAB II**

### **TINJAUAN PUSTAKA**

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

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

Berdasarkan penelitian yang dilakukan oleh Akmal Nasution (2018) dengan judul “Forecasting Produksi Karet Menggunakan Metode Weighted Moving Average” Proses perhitungan peramalan secara manual menggunakan metode Weighted Moving Average tentunya harus menguasai keahlian khusus terutama dibidang matematis. Untuk itu peneliti bermaksud membangun sebuah rancangan aplikasi peramalan sehingga dapat digunakan oleh semua orang dengan mudah dan cepat. Teknik peramalan ini dapat diterapkan diberbagai bidang pekerjaan, termasuk dalam peramalan produksi karet. Dengan memperoleh data peramalan produksi karet dimasa mendatang tentunya dapat memberikan gambaran untuk langkah-langkah kerja kedepannya, sehingga dapat meningkatkan produktivitas kerja perusahaan. Adapun hasil penelitian yang dilakukan adalah mempermudah perusahaan dalam menentukan peramalan produksi karet pada periode berikutnya.

Berdasarkan penelitian Dewi Putu (2018) dengan judul “Sistem Informasi Peramalan Persediaan Barang Menggunakan Metode *Weighted Moving Average*” Terdapat kendala yang dialami oleh perusahaan antara lain permintaan dengan persediaan tidak seimbang dan pencatatan persediaan barang juga masih manual dengan tulisan tangan. Penentuan persediaan barang itu sendiri masih dilakukan

secara manual dengan tulisan tangan jadi untuk menentukan berapa jumlah persediaan barang yang akan disediakan perusahaan harus terlebih dahulu membandingkan jumlah barang yang keluar dengan perbandingan data sebelumnya juga data pengeluaran barang yang baru terjadi. Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk merancang dan membangun Sistem Informasi yang dapat membantu dalam pengolahan data barang dan peramalan persediaan barang untuk bulan berikutnya.

Berdasarkan penelitian yang dilakukan oleh Alfian Nurlifa (2017) dengan judul “Sistem Peramalan Jumlah Penjualan Menggunakan Metode *Moving Average* Pada Rumah Jilbab Zaky” Bentuk laporan yang masih manual membuat pemilik menjadi kesulitan dalam melihat perkembangan usaha, akibatnya pemilik dapat melakukan kesalahan dalam pengambilan keputusan terutama untuk penambahan atau pengurangan stok barang. Tujuan dari penelitian ini adalah untuk memanfaatkan informasi yang ada semaksimal mungkin dengan menggunakan sistem informasi manajemen. Dalam sistem informasi manajemen ini akan mengelola data untuk dijadikan informasi yang lebih mudah dilihat seperti menggunakan grafik. Sistem informasi manajemen ini disertai peramalan untuk membantu pengguna terutama manajer dalam pengambilan keputusan.

Berdasarkan penelitian yang dilakukan oleh Eko Sulistiono (2016) dengan judul “Sistem Prediksi Penjualan Dengan Single Moving Everage Pada Distro Mega Busana I” Transaksi penjualan mencapai seratus kali dalam sehari dengan sekali transaksi dapat berisi lima macam perlengkapan fashion bahkan lebih, sehingga sering mengalami kekurangan stok barang. Hal ini dikarenakan Distro

Mega Busana II tidak memiliki sebuah sistem yang dapat memperkirakan jumlah persediaan yang harus dilakukan. Sistem Informasi Prediksi Penjualan menggunakan Metode Single Moving Average digunakan untuk menyelesaikan permasalahan tersebut. Sistem informasi ini menghasilkan prediksi yang dapat memprediksi jumlah penjualan pada periode yang akan datang dan jumlah ini dapat digunakan sebagai acuan dalam pemesanan barang pada supplier. Implementasi Sistem informasi prediksi penjualan dapat digunakan untuk melakukan prediksi penjualan per bulan dan dapat digunakan untuk melakukan pemesanan barang

## **II.2. Uraian Teoritis**

### **II.2.1. Sistem Informasi**

Menurut Imade Budi Adnyana (2016) bahwa 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.

Menurut Imade Budi Adnyana (2016) bahwa 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.2.2. Peramalan**

Menurut Widiyarini (2016) bahwa Peramalan (*Forecasting*) adalah proses untuk memperkirakan besar kebutuhan di masa datang yang meliputi kebutuhan dalam ukuran kuantitas, kualitas, waktu dan lokasi yang dibutuhkan dalam rangka memenuhi permintaan barang ataupun jasa. Peramalan tidak terlalu dibutuhkan dalam kondisi permintaan pasar yang stabil karena permintaannya relatif kecil. Sebaliknya, akan sangat dibutuhkan bila kondisi permintaan pasar bersifat kompleks. Metode serial waktu sangat tepat untuk meramalkan permintaan yang memiliki pola permintaan dimasa lalunya cukup konsisten dalam periode waktu yang lama, sehingga pola tersebut dapat diharapkan tetap berlanjut pada permintaan di masa yang akan datang.

### **II.2.3. Metode Moving Average**

Menurut Alfian Nurlifa (2017) bahwa Metode moving average digunakan jika data masa lalu merupakan data yang tidak memiliki unsur trend atau

faktor musiman. Moving average forecasting banyak digunakan untuk menentukan trend dari suatu deret waktu. Tujuan utama dari penggunaan rata-rata bergerak adalah untuk menghilangkan atau mengurangi acakan (randomness) dalam deret waktu. Tujuan ini dapat dicapai dengan merata-ratakan beberapa nilai data bersama-sama, dengan cara mana kesalahan-kesalahan positif dan negatif yang mungkin terjadi dapat dikeluarkan atau dihilangkan.

Menurut Alfian Nurlifa (2017) Untuk mendapatkan nilai dari moving average sebelumnya ditentukan terlebih dahulu jumlah periode (T). Setelah ditentukan jumlah periode yang akan digunakan dalam observasi pada setiap rata-rata atau MA (T) dapat dihitung nilai rata-ratanya. Secara aljabar, rata-rata bergerak (MA) dapat dituliskan sebagai berikut :

$$WMA = (\sum (Dt * bobot)) / (\sum bobot) \quad \dots\dots\dots(2)$$

**Keterangan :**

Dt : data aktual pada periode t

bobot : bobot yang diberikan untuk setiap bulan

**II.2.4. Database**

Menurut Neni Purwati dan Hendra Kurniawan (2015 : 50).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. 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

### II.2.5. Normalisasi

Menurut Dwi Puspitasari (2016) bahwa Normalisasi adalah proses pembentukan struktur basis data sehingga sebagian besar ambiguity bisa dihilangkan. Normalisasi merupakan sebuah teknik dalam logical desain sebuah basis data relasional yang mengelompokkan atribut dari suatu tabel sehingga membentuk struktur tabel yang normal. Adapun kriteria tabel dikatakan normal adalah ketika tidak ada kerangkapan data (redudansi data). Tujuan dari normalisasi adalah untuk :

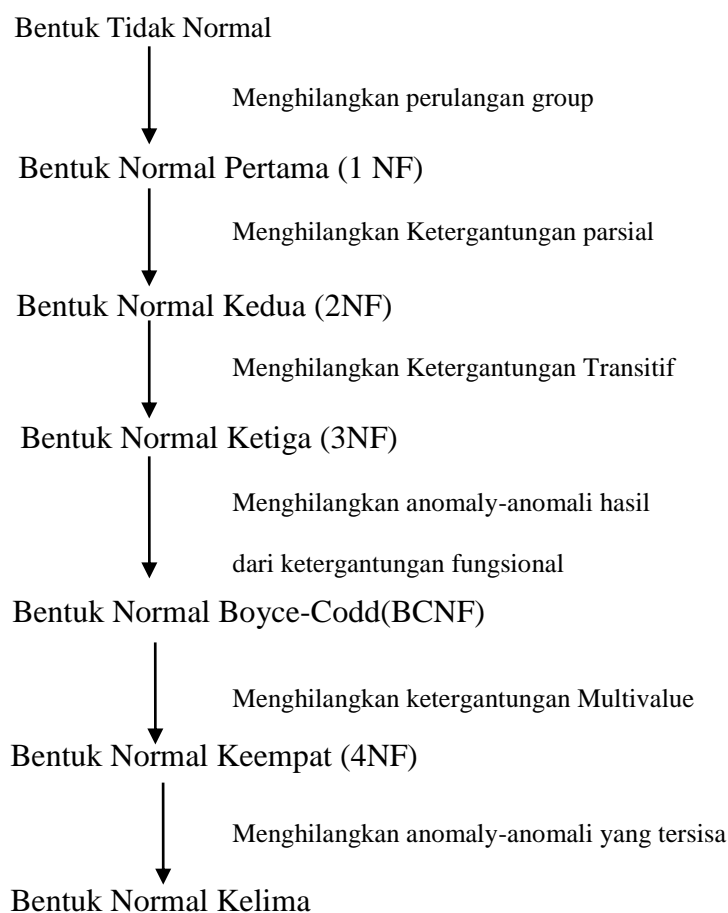
1. Untuk menghilangkan kerangkapan data sehingga meminimumkan pemakaian *storage* yang dipakai oleh *base relations* (file)
2. Untuk mengurangi kompleksitas
3. Untuk mempermudah pemodifikasian data

Gambaran proses normalisasi adalah :

1. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat, kemudian

2. Apabila tabel yang diuji belum memenuhi persyaratan tertentu, maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal.

Tahapan dalam normalisasi dimulai dari tahap paling ringan (*1NF*) hingga paling ketat (*5NF*). Biasanya hanya sampai pada tingkat *3NF* atau *BCNF* karena sudah cukup memadai untuk menghasilkan tabel-tabel yang berkualitas baik. Urutan tahapan normalisasi tampak seperti gambar II.3.



**Gambar II.1. Tahapan pada Normalisasi**  
(Sumber : Dwi Puspitasari, dkk ; 2016)

Adapun aturan dalam normalisasi adalah suatu tabel dikatakan baik (efisien) atau normal jika memenuhi 3 kriteria sbb:

1. Jika ada dekomposisi (penguraian) tabel, maka dekomposisinya harus dijamin aman (*Lossless-Join Decomposition*). Artinya, setelah tabel tersebut diuraikan / didekomposisi menjadi tabel-tabel baru, tabel-tabel baru tersebut bisa menghasilkan tabel semula dengan sama persis.
2. Terpeliharanya ketergantungan fungsional pada saat perubahan data (*Dependency Preservation*).
3. Tidak melanggar Boyce-Codd Normal Form (BCNF)

Jika kriteria ketiga (BCNF) tidak dapat terpenuhi, maka paling tidak tabel tersebut tidak melanggar Bentuk Normal tahap ketiga (3rd Normal Form / 3NF). Pada penelitian ini formula yang dibuat sampai memenuhi bentuk normal ke 3 yaitu 3NF.

#### **II.2.6. Visual Studio 2010**

Menurut Rolly Yesputra (2017) bahwa Visual studio merupakan suatu perangkat lunak yang dapat digunakan untuk pengembangan berbagai macam aplikasi yang memiliki berbagai macam tipe antara lain aplikasi *desktop* (*Windows Form, CommandLine (Console)*), *Aplikasi Web, Windows Mobile (Paket PC)*. Visual Studio 2010 memiliki lebih dari satu kompiler, SDK (*Software Development Kit*), dan Dokumentasi Tutorial (*MSDN Library*). Kompiler yang dimasukkan kedalam Visual Studio 2010 antara lain Visual Basic, Visual C#, Visual C++, Visual InterDev, Visual J++, Visual F#, dan Visual Source Safe, dan banyak yang lainnya. Dan semua itu sudah terpaket dan diperuntukkan kedalam platform .Net Framework 4.0 atau versi yang lebih tinggi. Visual studio ini dapat

digunakan untuk membuat aplikasi yang berbasis desktop yang merupakan platform windows, namun juga dapat dijalankan dalam bentuk Microsoft Intermediate Language diatas .Net Framework. Selain itu Visual Studio juga dapat digunakan untuk membuat aplikasi yang dapat dijalankan diatas windows mobile yang berjalan diatas .Net Compact Framework. Visual Studio 2010 terbagi menjadi beberapa tipe diantaranya :

1. Visual Studio 2010 Express Edition yang bisa digunakan secara gratis tanpa memberikan royalti kepada Microsoft Inc.
2. Visual Studio Standard Edition
3. Visual Studio 2010 Professional Edit
4. Visual Studio 2010 Ultimate Edition

### **II.2.7. SQL Server 2008**

Menurut Eka Iswandy (2015 : 73) bahwa SQL (*Structured Query Language*) adalah bahasa non procedural untuk mengakses data pada database relasional. SQL adalah bahasa database yang dipergunakan dalam menyelesaikan permasalahan dalam database serta mempunyai kelebihan dalam mengolah data. Standar SQL mula-mula didefinisikan oleh ISO (*International Standards Organization*) dan ANSI (*the American National Standards Institute*) yang dikenal dengan sebutan SQL86.

Dengan menggunakan SQL, kita dapat melakukan hal-hal berikut:

1. Memodifikasi struktur database .
2. Mengubah, mengisi, menghapus isi database.

3. Mentransfer data antara database yang berbeda. SQL ada yang dikembangkan untuk PC dan ada juga yang dikembangkan untuk dapat mengakomodasi database yang sangat besar.

Beberapa contohnya antara lain:

1. *Microsoft Access*

Digunakan untuk PC, sangat mudah dipakai dimana perintah SQL dapat langsung dimasukkan atau melalui fasilitas yang telah digunakan.

2. *Microsoft Query*

SQL yang dipaket dengan produk lain dari Microsoft Windows, yaitu Microsoft Visual Studio seperti Visual Basic dan Visual C++. Untuk terhubung dengan database lain menggunakan ODBC.

3. *Oracle*

Digunakan untuk perusahaan yang menggunakan database besar.

### **II.2.8. UML (*Unified Modelling Language*)**

Menurut Saipul Anwar, et al (2016 : 75-76) bahwa *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

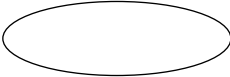



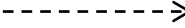

#### **II.2.8.1. Use Case Diagram**

*Use case* diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja

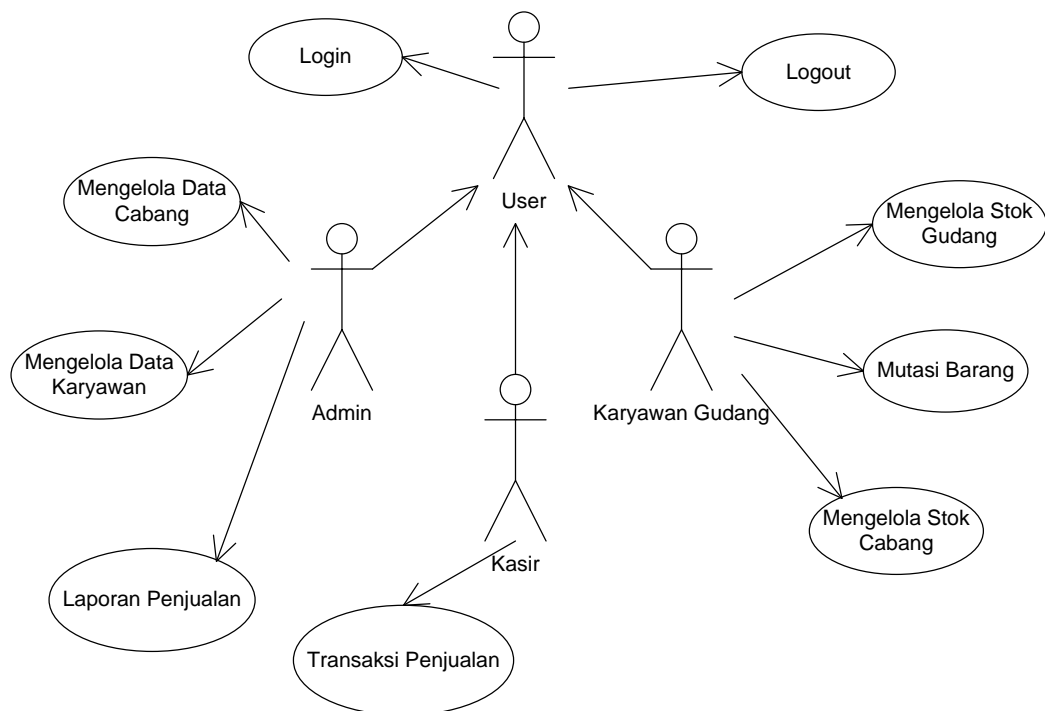
yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Ade Hendini, 2016 : 108).

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 mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan 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)



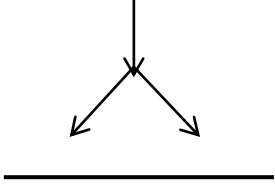
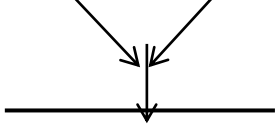
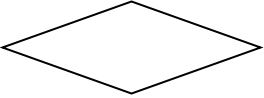
**Gambar II.2. Use Case Diagram**  
(Sumber : Ade Hendini, 2016 : 112)

### II.2.8.2. Activity Diagram

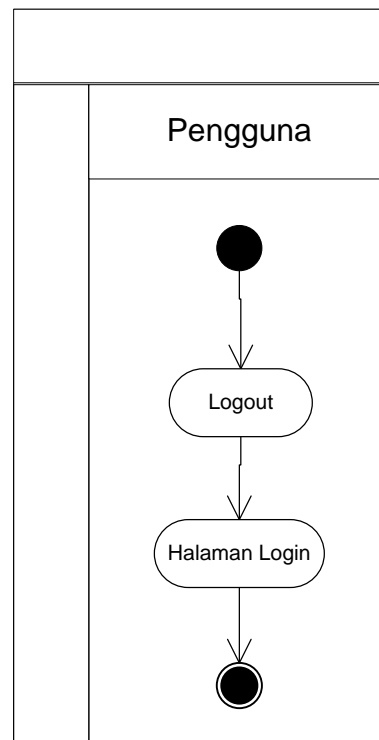
Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.2 dibawah ini:

**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.
▭	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.

	<p><i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel 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>
<div style="border: 2px solid black; padding: 2px; display: inline-block;">New Swimlane</div>	<p><i>Swimlane</i>, pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.</p>

(Sumber : Ade Hendini, 2016 : 109)

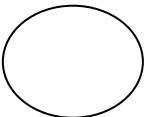
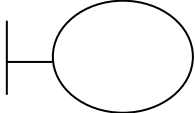
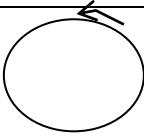
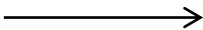
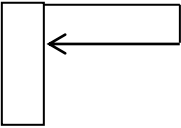




**Gambar II.4. Activity Diagram**  
(Sumber : Ade Hendini, 2016 : 112)

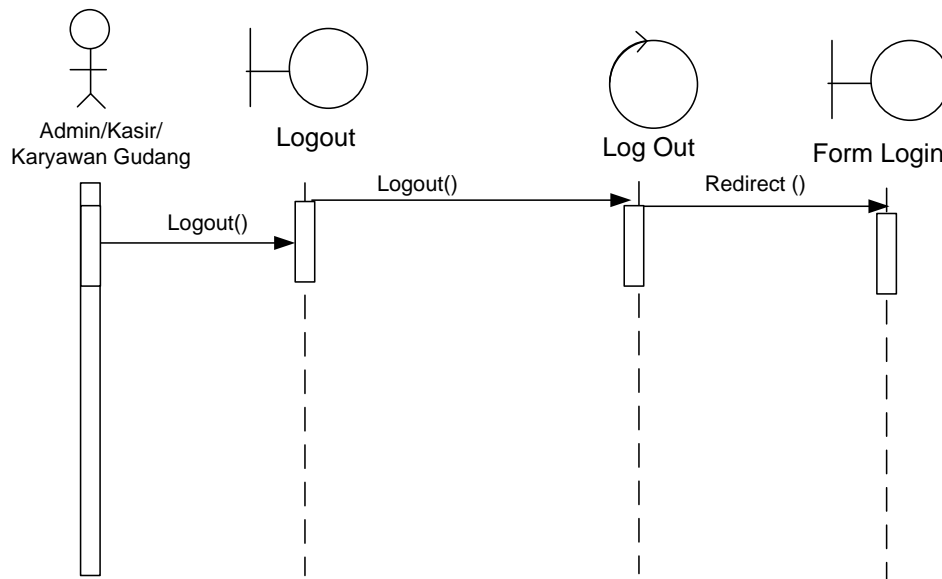
### II.2.8.3. Sequence Diagram

*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> .

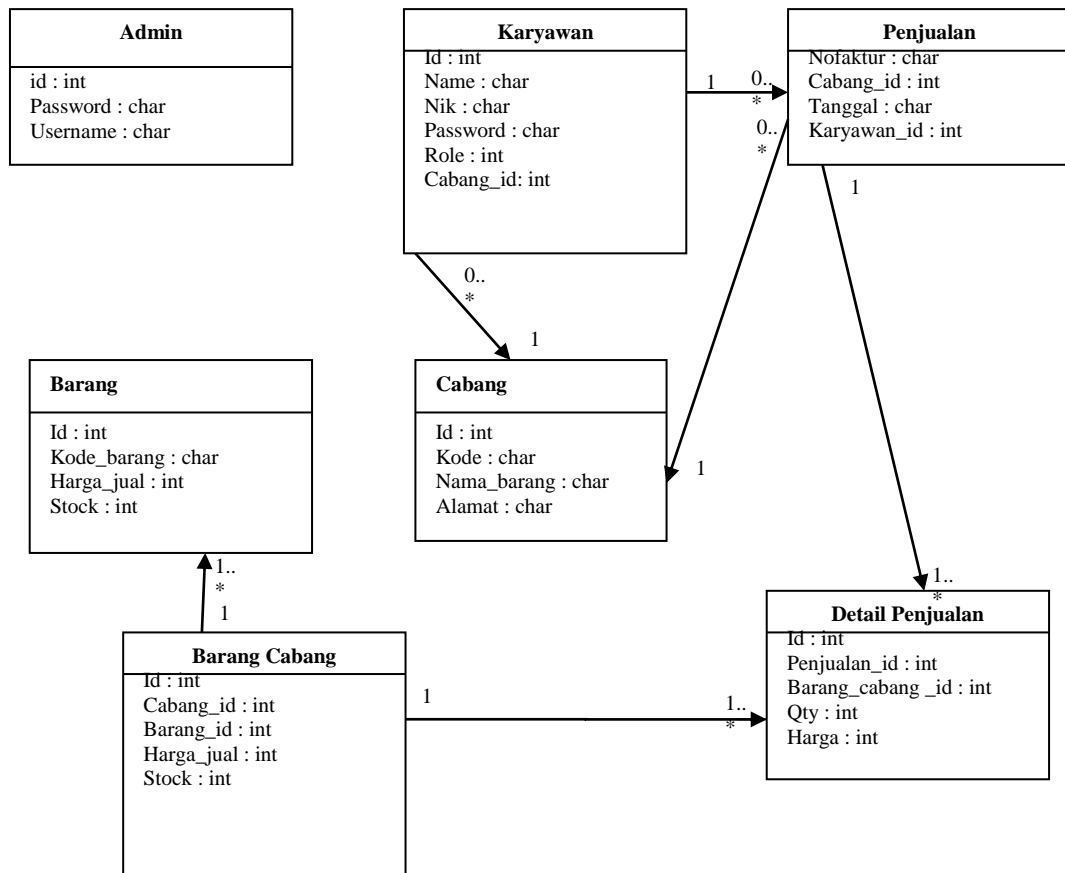
(Sumber : Ade Hendini, 2016 : 110)



**Gambar II.5. Sequence Diagram**  
(Sumber : Ade Hendini, 2016 : 114)

#### II.2.8.4. 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 juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class* diagram secara khas meliputi : Kelas (*Class*), Relasi *Associations*, *Generalization* 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.6. Class Diagram**  
 (Sumber : Ade Hendini, 2016 : 115)