

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

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

Berdasarkan penelitian Ruli Utami yang berjudul “Perbandingan Metode *Holt Exponential Smoothing* dan *Winter Exponential Smoothing* Untuk Peramalan Penjualan Souvenir” Dalam penjualan souvenir, seringkali UD. Fajar Jaya mengalami kekosongan stok souvenir ketika pesanan membludak di waktu-waktu tertentu. Hal ini terjadi karena tidak terdapat analisa dan strategi manajemen stok (tidak dapat memprediksi berapa jumlah optimal souvenir yang harus di sediakan) yang di lakukan oleh eksekutif UD. Fajar Jaya. Untuk mengatasi hal tersebut di atas, maka penulis mengusulkan untuk melakukan peramalan terhadap tingkat penjualan souvenir dengan menggunakan metode *Holt* dan *Winter* yang ada pada pengembangan metode *Exponential Smoothing* (ES). Dari penerapan kedua metode tersebut, kemudian dibuat perbandingan efektifitas metode yang di ukur melalui akurasi data aktual dan hasil peramalan dengan cara mengetahui tingkat kesalahan peramalan.[1]

Berdasarkan penelitian Ridho Denanda Putra yang berjudul “Rancang Bangun Aplikasi Peramalan Permintaan Barang Dengan Metode Pemulusan Eksponensial Winter Pada PT. Supramedika Prima” Dihasilkan aplikasi yang dapat meramalkan permintaan barang untuk periode yang akan datang dengan

masukan berupa data penjualan barang pada periode sebelumnya, dengan menggunakan Metode Pemulusan Eksponensial Winter yang secara keseluruhan memiliki tingkat persentase kesalahan peramalan sebesar 13,2%. Aplikasi peramalan permintaan barang dibuat dengan berbasis *desktop* untuk penggunaannya. [2]

Menurut Ni Ketut Dewi Ari Jayanti dengan judul “Penerapan Metode *Triple Exponential Smoothing* pada Sistem Peramalan Penentuan Stok Obat” Untuk memudahkan dalam menggali informasi yang tersimpan dalam data penjualan obat pada periode sebelumnya tersebut maka dibuat sebuah aplikasi yang diimplementasikan dengan bidang ilmu data mining dengan salah satu bagiannya adalah forecasting dengan menggunakan metode smoothing data deret berkala. Metode yang dijadikan dasar pada sistem ini adalah metode triple exponential smoothing dengan pola data deret berkala. Sistem ini membantu manajemen apotek dalam mengelola data penjualan dan menggali informasi untuk menentukan pengadaan stok obat yang tepat dan akurat. Keakuratan maksimal aplikasi ini dalam menentukan perkiraan penjualan obat mencapai 98,15%. [3]

II.2. Landasan Teori

II.2.1. Sistem

Sistem dapat terdiri dari sistem-sistem bagian (*subsystem*). Sebagai misal, sistem komputer dapat terdiri dari subsistem perangkat keras dan subsistem perangkat lunak. Masing-masing subsistem dapat terdiri dari subsistem-subsistem yang lebih kecil atau terdiri dari komponen-komponen. Subsistem perangkat keras (*hardware*) dapat terdiri dari alat masukan, alat pemroses, alat keluaran dan

simpanan luar, dan kemudian subsistem-subsistem tersebut akan berinteraksi sedemikian rupa sehingga dapat mencapai satu kesatuan yang terpadu. Dalam buku Analisa dan Design Sistem Informasi Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis. “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. [4]

II.2.1.1. Karakteristik Sistem

Sistem mempunyai karakteristik atau sifat-sifat tertentu, diantaranya adalah sebagai berikut :

1. Komponen Sistem (*Components*)

Sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

2. Batas Sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batas sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem (*Environments*)

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga merugikan suatu sistem.

4. Penghubung Sistem (*Interface*)

Penghubung merupakan media penghubung antara suatu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari suatu subsistem ke subsistem lainnya.

5. Masukan Sistem (*Input*)

Masukan adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*).

6. Keluaran sistem (*Output*)

Keluaran adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran dapat berupa masukan untuk subsistem yang lain.

7. Pengolah Sistem (*Process*)

Sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolah. Pengolah yang akan merubah masukan menjadi keluaran.

8. Sasaran Sistem (*Objectives*)

Sistem mempunyai tujuan atau sasaran yang akan menentukan masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem [5]

II.2.2. Informasi

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. (1) Akurat (*Accurate*) Berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. (2) Tepat waktu

(*Timelines*) Berarti informasi yang datang pada si penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai logika karena informasi merupakan landasan dalam pengambilan keputusan. (3) Relevan (*Relevance*) Berarti informasi tersebut bermanfaat bagi pemakainya. [6]

II.2.3. 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) Kehandalan (*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 dibagi 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. [7]

II.2.4. Peramalan

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. [8]

II.2.5. Metode *Winter Exponential Smoothing*

Model *Winter Exponential Smoothing (Triple Exponential Smoothing Multiplicative)* ini digunakan untuk meramalkan data dengan pola musiman. Berbeda dengan model *Holt Double Exponential Smoothing*, model *Triple Exponential Smoothing Multiplicative* menggunakan tiga parameter untuk mencapai nilai peramalan yaitu α , β , dan μ . Langkah untuk melakukan peramalan terdiri dari 3 bagian yaitu proses pemulusan (A_t), proses estimasi *trend* (T_t), dan proses estimasi musiman (Y_{t+p}). Rumus yang digunakan untuk menghitung pemulusan adalah sebagai berikut:

$$A_t = \alpha \frac{Y_t}{S_{t-L}} + (1-\alpha)(A_{t-1} + T_{t-1}) \dots \dots \dots (1)$$

$$T_t = \beta (A_t - A_{t-1}) + (1-\beta)T_{t-1} \dots \dots \dots (2)$$

$$S_t = \mu \frac{Y_t}{A_t} + (1-\mu)S_{t-L} \dots \dots \dots (3)$$

Dengan ketentuan:

- At : Nilai pemulusan ke – t
- α : Parameter pemulusan untuk data
- β : Parameter pemulusan untuk estimasi *trend*
- Yt : Data Aktual ke – t
- Tt : Estimasi *trend* ke – t
- p : Jumlah periode yang akan diramalkan
- St : Nilai data ramalan

Sama halnya dengan model *Holt Double Exponential Smoothing*, nilai A1 dapat diasumsikan sama dengan nilai data aktual yang pertama yaitu Y1. Sedangkan nilai T1 dapat diasumsikan bernilai 0 (karena nilai *trend* yang diperoleh dari periode yang lalu tidak ada), nilai estimasi musiman awal (S1) diasumsikan dengan nilai 1(menghilangkan pengaruh musiman pada data aktual) Rumus untuk menentukan nilai ramalannya adalah sebagai berikut:

$$Y_{t+p} = (A_t + T_{t+p})S_{t-L+p} \dots \dots \dots (4)$$

Dengan ketentuan:

- μ : Parameter pemulusan untuk estimasi musiman ($0 < \mu < 1$)
- St: Estimasi musiman ke – t
- L : Panjangnya musim. [9]

II.2.6. 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. 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 [10]

II.2.7. Normalisasi

Normalisasi adalah suatu teknik untuk mengorganisasikan data kedalam tabel-tabel untuk memenuhi kebutuhan pemakai didalam suatu organisasi. Tujuan dari normalisasi adalah :

- a. Untuk menghilangkan kerangkapan data.
- b. Untuk mengurangi kompleksitas.
- c. Untuk mempermudah pemodifikasian data.

Proses normalisasi antara lain :

- a. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu kebeberapa tingkat.
- b. Apabila tabel yang diuji belum memenuhi persyaratan tertentu maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi data yang optimal.

Bentuk-bentuk dari normalisasi adalah :

- a. Bentuk tidak normal (*unformalized form*)

Bentuk ini merupakan bentuk data yang direkam, tidak ada keharusan untuk mengikuti suatu format tertentu, dapat saja data tidak lengkap atau terduplikasi.

- b. Bentuk normal pertama (1NF atau *first normal form*)

Bentuk normal pertama mempunyai ciri-ciri yaitu setiap data dibentuk dalam *flat file* (file dasar) dan data dibentuk dalam satu record demi satu record. Tidak ada set atribut yang berulang-ulang atau atribut yang bernilai ganda.

- c. Bentuk normal kedua (2NF atau *second normal form*)

Bentuk normal kedua mempunyai syarat yaitu bentuk data telah memenuhi kriteria bentuk normal pertama, atribut bukan kunci haruslah bergantung secara fungsi pada kunci utama, atau *primary key*, sehingga untuk bentuk normal kedua haruslah sudah ditentukan kunci-kunci *field*. Kunci *field* harus unik dan dapat mewakili atribut lain yang menjadi anggotanya.

d. Bentuk normal ketiga (3NF atau *three normal form*)

Untuk menjadi bentuk normal ketiga maka relasi haruslah dalam bentuk normal kedua dan sama atribut bukan primer tidak punya hubungan yang transi, dengan kata lain setiap atribut bukan kunci haruslah bergantung pada *primary key* secara menyeluruh.[11]

II.2.8. PHP (*Hypertext Preprocessor*)

PHP atau kependekan dari *Hypertext Preprocessor* adalah salah satu bahasa pemrograman *open source* yang sangat cocok atau dikhususkan untuk pengembangan *web* dan dapat ditanamkan pada sebuah skripsi HTML. Bahasa PHP dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti C, *Java*, dan *Perl* serta mudah untuk dipelajari. PHP merupakan bahasa *scripting server – side*, dimana pemrosesan datanya dilakukan pada sisi *server*. Sederhananya, *server* yang akan menterjemahkan skrip program, baru kemudian hasilnya akan dikirim kepada *client* yang melakukan permintaan. Adapun pengertian lain PHP adalah *akronim* dari *Hypertext Preprocessor*, yaitu suatu bahasa pemrograman berbasis kode – kode (*script*) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke *web browser* menjadi kode HTML".[12]

II.2.9. MySQL (*My Structure Query Language*)

MySQL (*My Structure Query Language*) merupakan sebuah program pembuat database yang bersifat *open source*, artinya semua orang dapat menggunakannya dan dapat dijalankan pada semua *platform* baik *windows* maupun

linux. *MySQL* juga merupakan sebuah perangkat lunak sistem manajemen basis data *SQL* yang bersifat jaringan sehingga dapat digunakan untuk aplikasi multi *user*. *MySQL* juga sering dikenal dengan nama sistem manajemen *database* relasional. Suatu *database* relasional menyimpan data dalam table yang terpisah. Tabel -table tersebut terhubung oleh suatu relasi terdefinisi yang memungkinkan memperoleh kombinasi data dari beberapa table dalam suatu permintaan. Untuk administrasi *database*, seperti pembuatan *database*, pembuatan tabel, dan sebagainya dapat digunakan aplikasi berbasis web seperti *phpMyAdmin* dengan aplikasi *XAMPP*. [13]

II.2.10. UML (*Unified Modeling Language*)


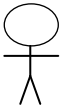


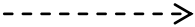
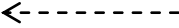
Menurut Gata (2013 : 4) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :[14]

1. *Use case* Diagram

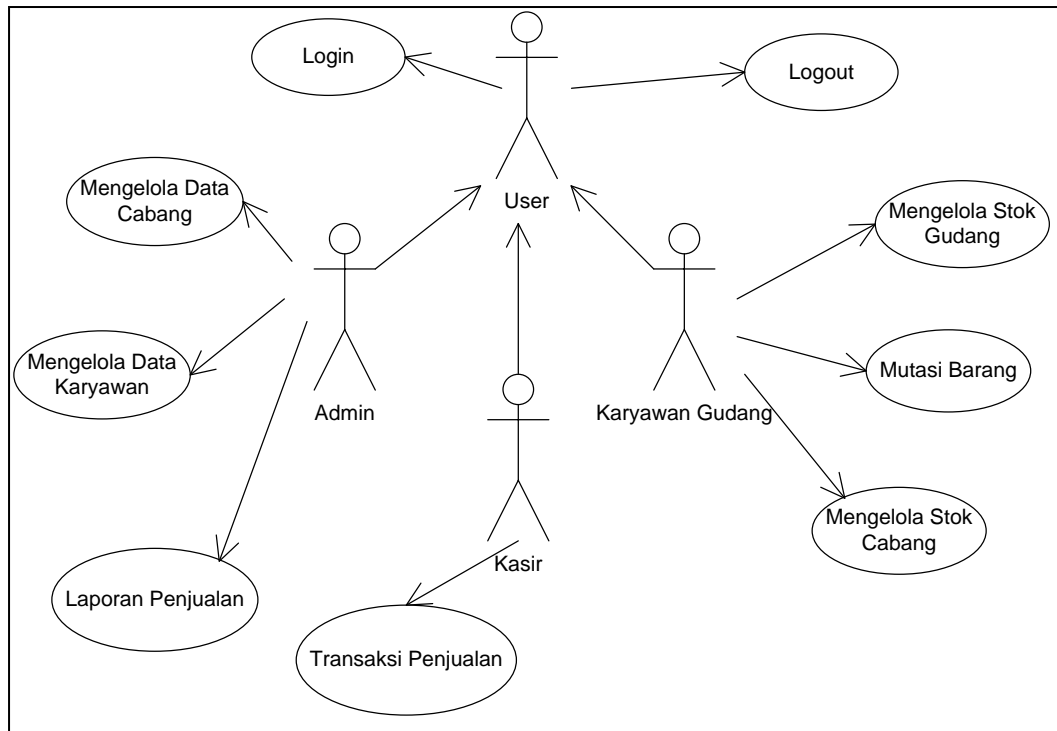
Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara

satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram, yaitu :

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 <i>control</i> 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 mengidikasikan 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 : Ade Hendini, 2016)




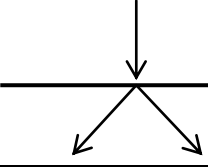
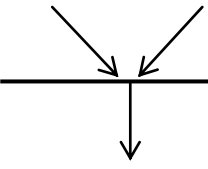
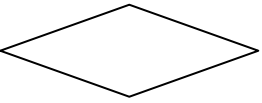
Gambar II.1. Use Case Diagram
 (Sumber : Ade Hendini, 2016 : 112)

2. Diagram Aktivitas (*Activity Diagram*)

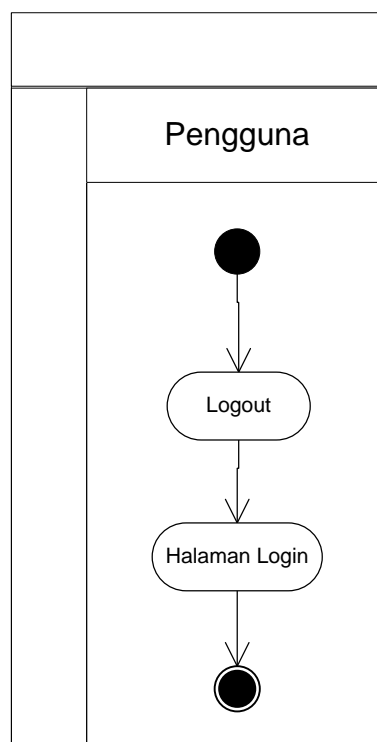
Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu :

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 diagram</i> untuk menunjukkan siapa melakukan apa.</p>

(Sumber : Ade Hendini, 2016)

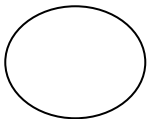
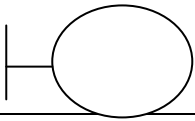
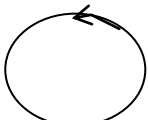

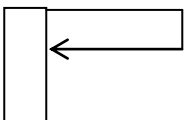




Gambar II.2. Activity Diagram
(Sumber : Ade Hendini, 2016 : 112)

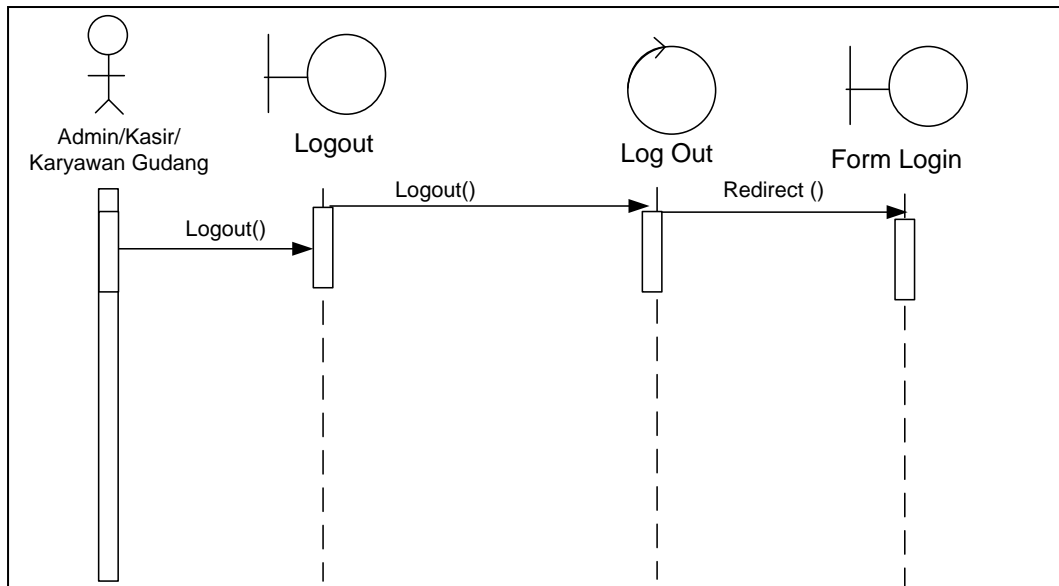
3. Diagram Urutan (*Sequence Diagram*)

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*, yaitu :

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 <i>formentry</i> 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)



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

4. Diagram Kelas (*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 (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

Tabel II.4. Multiplicity Class Diagram

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Ade Hendini, 2016)