

BAB II

TINJAUAN PUSTAKA

II.1. Peneliti Terkait

Adapun Penelitian terkait ini, peneliti akan membandingkan hasil penelitiannya dengan dua jurnal yaitu :

1. Menurut Radiant Victor Imbar (2013) dengan judul “Aplikasi Peramalan Stok Barang Menggunakan Metode *Double Exponential Smoothing*” Tujuan dari penelitian ini adalah : Membuat Sistem Informasi yang dapat proses pembelian dan penjualan barang secara terkomputerisasi, pengaturan stok barang, harga (satuan, grosir, dan beli) dari setiap barang yang ada dan pengecekan barang yang harus di retur dengan menerapkan aplikasi PHP Dan MySql.
2. Menurut Cahyarizki Adi Utama, 2016, yang Berjudul “Pengembangan Sistem Informasi Stok Barang Dengan Peramalan Menggunakan Metode *Double Exponential Smoothing* (Studi Kasus : PT. Tomah Jaya Elektrikal)” Dan apabila ketersediaan barang terlalu banyak mengakibatkan kerugian dalam biaya perawatan bahkan barang mengalami penurunan kualitas dikarenakan terlalu lama tersimpan. Untuk meminimalkan dan mengatasi permasalahan tersebut perlu adanya suatu system terkomputerisasi yang dapat meramalkan kebutuhan stok barang secara optimal. Dan dapat memenuhi kebutuhan barang pada saat pelayanan. Metode yang digunakan adalah *Double Exponential Smoothing* karena Metode ini tergolong dalam metode time

series (runtut waktu) yang mempergunakan data masa lalu untuk memprediksi sesuatu di masa yang akan datang. Data yang akan digunakan juga menunjukkan adanya trend. Sistem ini dapat meramalkan kebutuhan stok barang untuk 1 periode / 1 bulan kedepan agar stok barang optimal. Dalam peramalan menggunakan metode ini nilai alpha (α) sangat berpengaruh terhadap kesalahan / *error* dari peramalan.

3. Menurut Nendang Kacikal, 2015, yang berjudul “Sistem Peramalan Dan Monitoring Persediaan Obat Di Rspg Cisarua Bogor Dengan Menggunakan Metode Single Exponential Smoothing Dan Reorder Point” Berdasarkan hasil pengujian, dapat disimpulkan bahwa metode Single Exponential Smoothing cocok untuk digunakan karena rata-rata akurasi peramalan mencapai 93,4%, selain itu sistem yang dibangun membantu Kepala Instalasi Farmasi dalam menentukan jumlah obat yang harus disediakan dan membantu bagian gudang dalam memantau persediaan stok obat.

II.2. Uraian Teoritis

II.2.1. Sistem Informasi

Sistem informasi adalah suatu kumpulan dari komponen – komponen dalam perusahaan atau organisasi yang berhubungan dengan proses penciptaan dan pengaliran informasi. System informasi dapat juga didefinisikan sebagai suatu system yang menerima sumber data sebagai *input* dan mengolahnya menjadi produk informasi sebagai *output*. System informasi merupakan suatu system yang terdiri dari beberapa subsistem atau komponen *hardware*, *software* dan

brainware, data dan prosedur untuk menjalankan *input*, proses, *output*, penyimpanan dan pengontrolan yang mengubah sumber data menjadi informasi. Atau dapat juga didefinisikan sebagai suatu system di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung oprasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan – laporan yang diperlukan (Marimin ; 2012 : 18).

Sistem informasi adalah sejumlah komponen (manusia, komputer, teknologi informasi, dan prosedur kerja), ada sesuatu yang diproses (data menjadi informasi), dan dimaksudkan untuk mencapai suatu sasaran atau tujuan. Penggunaan sistem informasi telah banyak diterapkan diberbagai bidang termasuk dalam bisnis. Salah satu tujuan penerapan sistem informasi dalam bidang bisnis agar dapat meningkatkan keuntungan bisnis dengan menggunakan kemampuan yang didapatkan dari sistem informasi. Ada beberapa kemampuan dari sistem informasi yang dapat mendukung dalam bidang bisnis. Kemampuan tersebut seperti pengurangan biaya, mempercepat pekerjaan, dapat meningkatkan kemudahan dalam pengambilan keputusan, dan peningkatan pelayanan terhadap pelanggan (Alfian Nurlifa : 2017)

II.2.2. Persediaan

Persediaan barang merupakan banyaknya barang yang digunakan untuk memfasilitasi produksi atau untuk memuaskan permintaan konsumen. Persediaan barang meliputi barang yang dibeli dan disimpan untuk dijual kembali. Dalam hal ini, untuk memenuhi persediaan barang sebuah perusahaan dagang melakukan

pembelian dan selanjutnya barang yang dibeli akan dijual kembali kepada konsumen. Persediaan berpengaruh terhadap neraca dan laporan Laba/Rugi dalam neraca perusahaan dagang atau perusahaan Industri, persediaan seringkali merupakan bagian yang sangat besar dari keseluruhan Aktiva lancar yang dimiliki perusahaan ini merupakan bukti betapa pentingnya persediaan barang dagangan untuk kegiatan pembelian dan penjualan dalam operasi perusahaan semacam itu. Pencatatan persediaan yang efektif seringkali merupakan kunci keberhasilan operasi perusahaan disamping pimpinan perusahaan berusaha untuk mempertahankan kuantitas dan jenis persediaan yang cukup untuk memenuhi permintaan konsumen, pimpinan juga harus menjaga keseimbangan persediaan agar tidak terlalu tinggi dan juga tidak terlalu rendah. Persediaan yang terlalu kecil akan menimbulkan kekecewaan konsumen sebaliknya persediaan yang terlalu tinggi akan menyebabkan biaya penyimpanan dan pemeliharaan persediaan akan membengkak .(Ruslan ; 2013 : 38).

II.2.3. *Double Exponential Smoothing*

Metode *Double Exponential Smoothing* merupakan model linear yang dikemukakan oleh Brown. Dalam metode ini dilakukan proses smoothing dua kali. Dasar pemikiran metode pemulusan eksponensial linear dari Brown adalah serupa dengan rata-rata bergerak linear, karena kedua nilai pemulusan tunggal dan ganda ketinggalan dari data yang sebenarnya jika terdapat unsur trend. Perbedaan antara nilai pemulusan tunggal dan ganda dapat ditambahkan dengan nilai pemulusan tunggal dan disesuaikan untuk trend.

Persamaan yang dipakai dalam implementasi pemulusan eksponensial linear satu-parameter. (Cahyarizki Adi Utama :2016)

Adapun rumus dari perhitungan metode *Double Exponential Smoothing* adalah sebagai berikut : (Cahyarizki Adi Utama :2016)

1. Pemulusan Eksponensial Tunggal:

$$S'_t = aX_t + (1 - a) S'_{t-1} \dots\dots\dots(1)$$

2. Pemulusan Eksponensial Ganda:

$$S''_t = aS'_t + (1 - a) S''_{t-1} \dots\dots\dots(2)$$

Dimana S'_t adalah nilai pemulusan eksponensial tunggal dan S''_t adalah nilai pemulusan eksponensial ganda

3. Pemulusan Trend:

$$a_t = S'_t + (S'_t - S''_t) = 2 S'_t - S''_t \dots\dots\dots(3)$$

4. Pemulusan Trend:

$$b_t = \frac{a}{1-a} ((S'_t - S''_t)) \dots\dots\dots(4)$$

5. Ramalan

$$F_t + m = a_t + b_t(m) \dots\dots\dots(5)$$

Dimana m adalah jumlah periode ke muka yang akan diramalkan.

Keterangan :

$F_t + m$ = Nilai ramalan untuk m periode ke depan

M = Jarak periode yang akan diramalkan

X_t = Nilai actual periode ke- t

S'_t = Nilai *Smoothing period* ke- t

a = Konstanta *Smoothing* ($1/n$) (Cahyarizki Adi Utama :2016)

Metode ini dikemukakan oleh *Brown's* untuk mengatasi perbedaan yang muncul antara data actual dan nilai peramalan apabila ada trend pada poltnya. Dasar pemikiran dari pemulisan eksponensial linier dari *Brown's* adalah serupa dengan rata-rata bergerak linier (*Linier Moving Average*), karena kedua nilai pemulisan tunggal dan ganda ketinggalan dari data yang sebenarnya bilamana terdapat unsur trend, perbedaan antara nilai pemulisan tunggal dan ganda ditambahkan kepada nilai pemulisan dan disesuaikan untuk trend. Dan digunakan untuk peramalan dengan cara menentukan besarnya α (*alpha*) secara *trial* dan *error* antara 0 sampai dengan 1, dan dilakukan proses *smoothing* dua kali. Kelebihan dari metode ini yaitu dapat memodelkan *trend* dan tingkat dari suatu deret waktu lebih efisien dibandingkan metode lain, karena memerlukan data yang lebih sedikit, dan menggunakan satu parameter sehingga menjadi lebih sederhana. Kekurangan dari metode ini yaitu metode ini memerlukan optimasi parameter sehingga memerlukan waktu untuk mencari α (*alpha*) yang paling optimal. (Annastasya Lieberty : 2015)

Untuk tahap-tahap dalam menentukan ramalan adalah sebagai berikut :

a) Menentukan Smoothing pertama (S^1_t)

$$S^1_t = \alpha X_t + (1-\alpha) S^1_{t-1},$$

X_t adalah nilai aktual periode ke- t

α adalah parameter smoothing

b) Menentukan Smoothing kedua (S^{2t})

$$S^{2t} = \alpha S^1_t + (1-\alpha) S^{2t-1},$$

α adalah parameter smoothing

c) Menentukan besarnya Konstanta (a_t)

$$a_t = 2S'_t - S''_t$$

d) Menentukan besarnya Slope (b_t)

$$b_t = \frac{\alpha}{1-\alpha}(S'_t - S''_t)$$

α adalah parameter smoothing

e) Menentukan besarnya forecast (S_{t+m})

$$S_{t+m} = a_t + b_t m \text{ (Annastasya Lieberty : 2015)}$$

II.2.4. Basis Data (*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).

Database adalah sekumpulan data mentah yang disusun menurut logika tertentu dan terorganisasi dalam bentuk yang dapat disimpan dan diproses oleh komputer. Contoh database dapat berisi data pegawai, data penjualan, pembayaran dan lain-lain. Data internal dari akunting, keuangan, penjualan dan bidang-bidang bisnis lainnya yang disimpan dalam suatu komputer dan disusun menurut logika tertentu disebut dengan internal database. Database seringkali disimpan dalam suatu perangkat tertentu pada komputer, seperti hard disk, compact disk, dan sebagainya. Hubungan antarsistem database dan sistem software sangat kuat karena sistem database yang dipakai sangat menentukan kemudahan aksesnya data sementara software sendiri memungkinkan peneliti memanipulasi data untuk dianalisis. (Dermawan Wibisono : 2012)

II.2.5. Normalisasi

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 (redundansi data). (Dwi Puspitasari, dkk, 2016)

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.

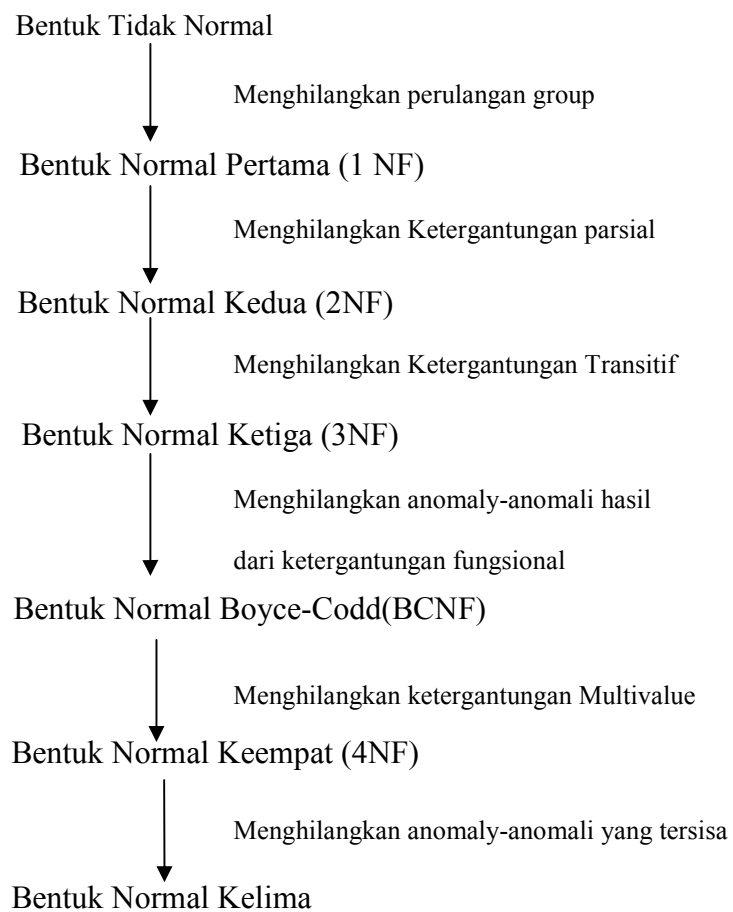
II.2.5.1. Proses Normalisasi

Gambaran proses normalisasi yaitu:

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. Untuk melakukan proses tersebut dibutuhkan beberapa tahapan.

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)*.

II.2.5.2. Bentuk-bentuk Normalisasi

1. Bentuk tidak normal

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan apa adanya sesuai keadaanya.

Tabel II.1 Tabel Tidak Normal

Kode	Kode Suplier	Nama Suplier	Kode Brg	Nama brg	Tgl
004	G01	Gobel Nustra	A01 A02	AC Model 1 AC Model 2	07/10/04
006	A03	Angkasa	B01 A03	Meja, Kursi	09/10/05

2. Bentuk normal tahap pertama (1st Normal Form)

Definisi :

Sebuah table disebut 1NF jika :

- a. Tidak ada baris yang duplikat dalam tabel tersebut.
- b. Masing-masing cell bernilai tunggal

Catatan: Permintaan yang menyatakan tidak ada baris yang duplikat dalam sebuah tabel berarti tabel tersebut memiliki sebuah kunci, meskipun kunci tersebut dibuat dari kombinasi lebih dari satu kolom atau bahkan kunci tersebut merupakan kombinasi dari semua kolom.

Tabel II.2. Tabel Normal 1

Kode	Kode Suplier	Nama Suplier	Kode Brg	Nama brg	Tgl
004	G01	Gobel Nustra	A01	AC Model 1	07/10/04
004	G01	Gobel Nustra	A02	AC Model 2	07/10/04
006	A03	Angkasa	B01	Meja	09/10/05
006	A03	Angkasa	A03	Kursi	09/10/05

3. Bentuk normal tahap kedua (2nd normal form)

Bentuk normal kedua (2NF) terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam *primary key* memiliki ketergantungan fungsional pada primary key secara utuh.

KdFaktur → Tgl. Jtempo.

KodeSup. NamaSup

KdFaktur, Kodebrg → namaBrg, Qty, Harga

4. Bentuk normal tahap ketiga (3rd normal form)

Sebuah tabel dikatakan memenuhi bentuk normal ketiga (3NF), jika untuk setiap ketergantungan fungsional dengan notasi $X \rightarrow A$, dimana A mewakili semua atribut tunggal di dalam tabel yang tidak ada di dalam X, maka :

- a. X haruslah superkey pada tabel tersebut.
- b. Atau A merupakan bagian dari primary key pada tabel tersebut.

5. Bentuk Normal Tahap Keempat dan Kelima

Penerapan aturan normalisasi sampai bentuk normal ketiga sudah memadai untuk menghasilkan tabel berkualitas baik. Namun demikian, terdapat pula bentuk normal keempat (4NF) dan kelima (5NF). Bentuk Normal keempat berkaitan dengan sifat ketergantungan banyak nilai (*multivalued dependency*) pada suatu tabel yang merupakan pengembangan dari ketergantungan fungsional. Adapun bentuk normal tahap kelima merupakan nama lain dari *Project Join Normal Form* (PJNF).

6. Boyce Code Normal Form (BCNF)

- a. Memenuhi 1st NF
- b. Relasi harus bergantung fungsi pada atribut superkey

KdFaktur → Tgl, Jtempo, KodeSup

KdSup → NamaSup

KdFaktur, Kodebrg → Qty, Harga

KodeBrg → NamaBrg (Kusrini : 2012)

7. Bentuk Normal Kelima

Sebuah tabel berada pada bentuk normal kelima (5NF) jika ia tidak dapat mempunyai dekomposisi lossless menjadi sejumlah tabel lebih kecil. Empat bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima berdasarkan pada konsep ketergantungan gabungan (*join dependence*) (Kusrini : 2012).

Tabel II.3. Tabel Bentuk Normal Kelima (5NF)

Peg#	Pry#	Ahli
1211	11	Perancangan
1211	28	Pemrograman

II.2.6. SQL Server 2008

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 didefenisikan oleh ISO (*International Standards Organization*) dan ANSI (*the*

American National Standards Institute) yang dikenal dengan sebutan SQL86. (Eka Iswandy : 2015 : 73)

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.(Eka Iswandy : 2015 : 73)

II.2.7. Visual Basic

Visual Basic 2010 adalah inkarnasi dari bahasa *visual basic* yang sangat populer dan telah dilengkapi dengan fitur serta fungsi yang setara dengan bahasa

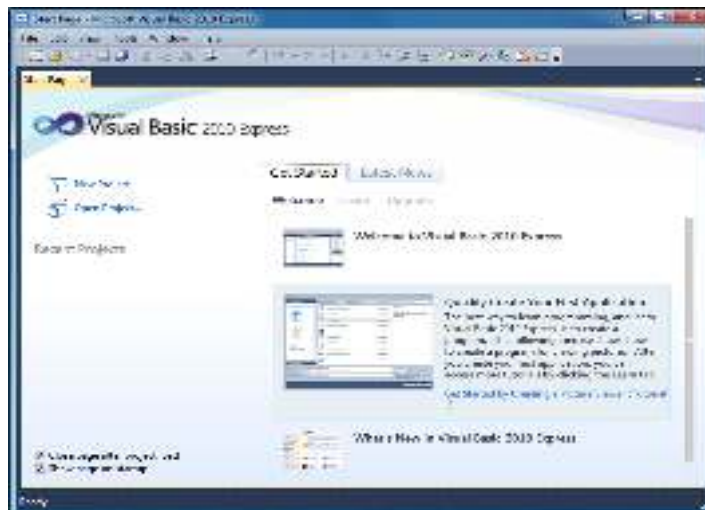
tingkat lainnya seperti C++. Anda dapat menggunakan *visual basic 2010* untuk membuat aplikasi *windows, mobile, web, dan office* atau kode yang telah ditulis oleh orang lain dan kemudian dimasukkan ke program lainnya. *Visual basic* menyediakan berbagai *tools* dan fitur canggih yang memungkinkan dapat menulis kode, menguji dan menjalankan program tunggal atau terkadang serangkaian program yang terkait dengan satu aplikasi. (Christopher Lee:2014:1)

Tabel II.4. Jendela Aplikasi Visual Studio 2010

No	Bagian	Keterangan
1	<i>Title Bar</i>	Menampilkan nama aplikasi yang sedang terbuka.
2	<i>Menu Bar</i>	Menampilkan daftar perintah yang memungkinkan anda dapat menulis, mengedit, menyimpan, mencetak, menguji dan menjalankan program <i>visual basic</i> .
3	<i>Standard Toolbar</i>	Berisi Tombol yang menjalankan perintah yang sering digunakan seperti <i>open project, new project, save, cut, copy, paste, dan undo</i>
4	<i>Toolbox</i>	Berisi komponen NET yang dapat anda gunakan untuk mengembangkan antarmuka pengguna grafis untuk program visual studio 2010.
5	<i>Area Kerja Utama</i>	Menampilkan item yang sedang di kerjakan
6	<i>Solution Explorer</i>	Menampilkan elemen dari visual basic solution, yaitu nama yang diberikan kepada program <i>visual basic</i> dan item lainnya yang dihasilkan oleh <i>visual basic 2010</i> sehingga program akan mengeksekusi dengan benar.
7	<i>Properties Window</i>	Setiap Objek dalam program <i>visual basic</i> memiliki seperangkat karakteristik yang disebut sifat-sifat objek.

(Sumber : Christopher Lee ; 2014 : 2)

Untuk melihat tampilan visual basic 2010 dapat dilihat pada gambar II.1. sebagai berikut :



Gambar II.2. Tampilan Utama Visual Basic 2010
(Sumber : Christopher Lee ; 2014 ; 2)

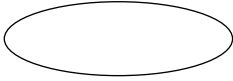
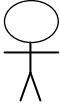
II.2.8. UML (*Unified Modeling Language*)



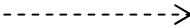
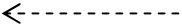
Menurut Gata (2013 : 4) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi 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 berdasarkan UML adalah sebagai berikut :

1. *Use case Diagram*

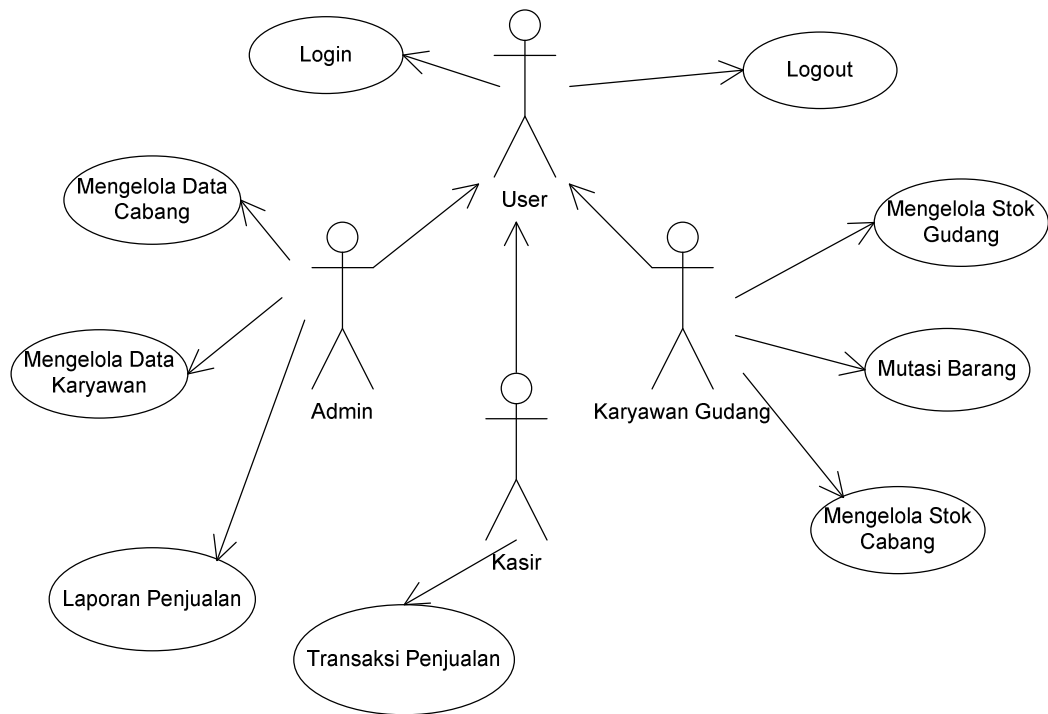
Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram, yaitu :

Tabel II.5. Simbol *Use Case*

Gambar	Keterangan
	<p><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>.</p>
	<p>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>.</p>

	<p>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.</p>
	<p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.</p>
	<p><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.</p>
	<p><i>Extend</i>, merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.</p>

(Sumber : Gata, 2013 : 4)



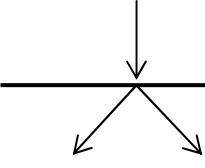
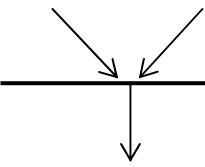
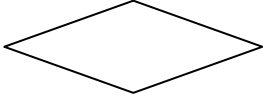
Gambar II.3. Use Case Diagram
(Sumber : Ade Hendini, 2016 : 112)

2. Diagram Aktivitas (*Activity Diagram*)

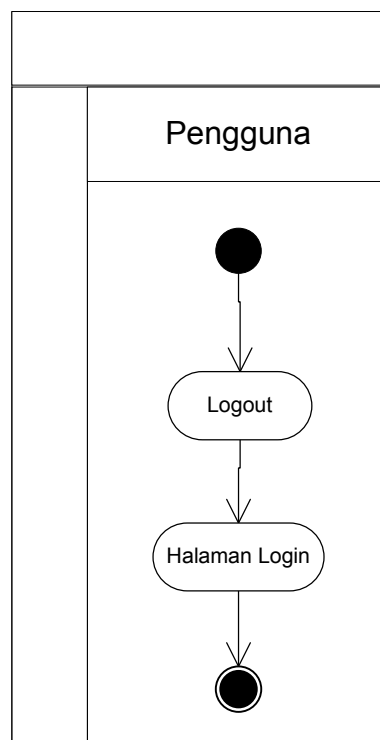
Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu :

Tabel II.6. 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 <i>rake</i>, 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 : Gata, 2013 : 6)

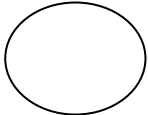
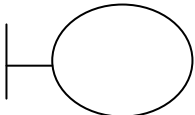
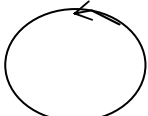

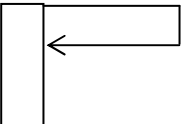



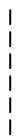
Gambar II.4. Activity Diagram
(Sumber : Ade Hendini, 2016 : 112)

3. Diagram Urutan (*Sequence Diagram*)

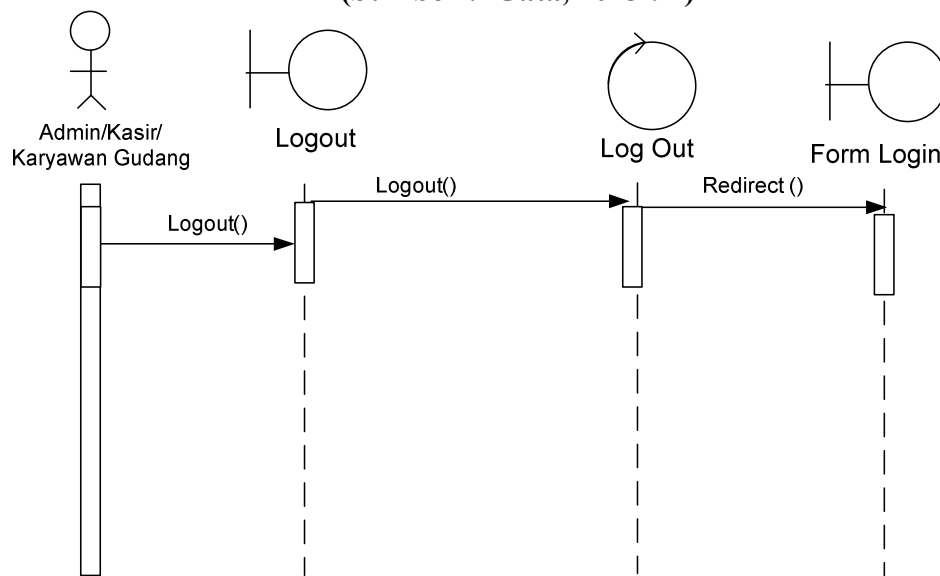
Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

Tabel II.7. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>EntityClass</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.

	<p><i>Activation, activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.</p>
	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p>

(Sumber : Gata, 2013 : 7)



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

4. *Class Diagram* (Diagram Kelas)

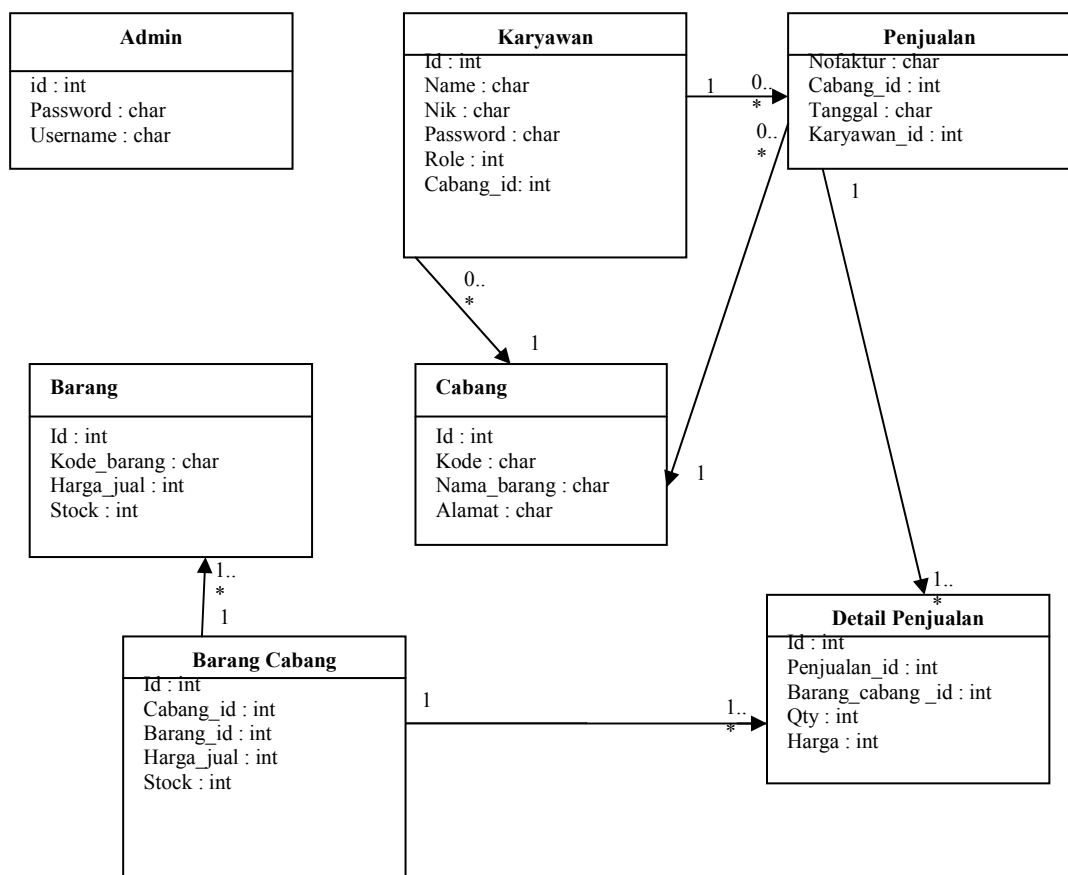
Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* 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.8. 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 : Gata, 2013 : 9)



Gambar II.6. Class Diagram

(Sumber : Ade Hendini, 2016 : 115)