

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

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

1. Pada jurnal **Agusta**, 2019, yang berjudul “**Aplikasi *Forecasting* Penjualan Dengan Metode *Single Exponential Smoothing* (Studi Kasus : Optik Nusantara)**”. Tujuan dari penelitian ini yaitu untuk memprediksi penjualan lensa kaca mata di bulan berikutnya serta Metode *Single Exponential Smoothing* dapat diterapkan dalam mengelola data-data penjualan pada bulan sebelumnya guna meramalkan penjualan di bulan berikutnya
2. Pada jurnal **Ery Setiyawan Jullev Atmadji**, 2016, yang berjudul “**Analisa Dan Penerapan Metode *Single Exponential Smoothing* Dan *Multi Agent Sistem* Pada Prediksi Penjualan**” Perangkat lunak berbasis intelligent agent ini dirancang menggunakan metodologi *Prometheus*, dan dikembangkan melalui *platform JADE* dengan menggunakan bahasa pemrograman *Java*. Pengembangan model berfokus pada bagaimana agen dapat membantu dalam melakukan peramalan terhadap kebutuhan stocking barang yang kemungkinan akan dipesan. Konsep yang diterapkan pada agen berdasarkan metode *Single Exponential Smoothing*, dengan menggunakan histori barang selama 2 tahun terakhir. Evaluasi dilakukan terhadap 24 data histori data permintaan selama 2

tahun. Metode perbaikan data dengan menggunakan MSE dengan hasil paling optimal antara nilai α 0.3 hingga 0.6.

3. Pada jurnal **Salman Alfarisi**, 2017, yang berjudul “**Sistem Prediksi Penjualan Gamis Toko Qitaz Menggunakan Metode *Single Exponential Smoothing***”. Tujuan dari penelitian ini yaitu digunakan untuk peramalan dalam jangka pendek biasanya satu bulan kedepan. Data dari hasil perhitungannya berfluktuasi disekitar nilai *mean* (rata-rata) yang tetap tanpa trend dan pertumbuhan. Dengan dibuatnya sistem aplikasi dari *Single Exponential Smoothing*, diharapkan pemilik toko Qitaz akan lebih mudah untuk menyimpan data penjualan gamisnya, menghitung nilai peramalan untuk periode berikutnya dan membuat laporan hasil perhitungan peramalan secara cepat dan akurat. Sehingga tujuan dan manfaat dari penelitian ini yaitu pendapatan optimal dan efisiensi biaya dapat tercapai.

II.2. Uraian Teoritis

II.2.1. Sistem

Ada dua kelompok pendekatan dalam mendefinisikan sistem. Ada yang menekankan pada prosedurnya dan ada yang menekankan pada komponen atau elemennya, diantaranya : Pendapat pertama menekankan sistem pada komponennya. “Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu”. Pendapat kedua menekankan sistem pada prosedurnya. “Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang

saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. [3].

Sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Berdasarkan pendapat para ahli tersebut maka dapat disimpulkan sistem adalah kumpulan elemen-elemen yang saling bekerja sama dan berinteraksi untuk memproses masukan kemudian saling berhubungan untuk mencapai suatu sasaran tertentu. [4].

Sebuah sistem terdiri dari berbagai unsur yang saling melengkapi dalam mencapai tujuan dan sasaran. Unsur-unsur yang terdapat dalam sistem itulah yang disebut dengan subsistem. Subsistem-subsistem tersebut harus saling berhubungan dan berinteraksi melalui komunikasi yang relevan sehingga sistem dapat bekerja secara efektif dan efisien. [5].

Suatu sistem mempunyai karakteristik atau sifat yang tertentu, yaitu mempunyai komponen-komponen (*component*), batas sistem (*boundary*), lingkungan luar sistem (*environment*), penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolah (*proses*), sasaran (*objectives*), ataupun tujuan (*goal*). Adapun Karakteristik dari sistem adalah [6] :

a. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem.

b. Batasan Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisah-pisahkan.

c. Lingkungan Luar Sistem (*Environtment*)

Lingkungan luar sistem adalah bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut.

d. Penghubung Sistem (*Interface*)

Penghubung sistem atau *interface* adalah media yang menghubungkan sistem dengan subsistem yang lain.

e. Masukan Sistem (*Input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*).

f. Keluaran Sistem (*Output*)

Keluaran ini merupakan masukan bagi subsistem yang lain. Seperti contoh sistem informasi, keluaran yang dihasilkan adalah informasi, di mana informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang merupakan *input* bagi subsistem lain.

g. Pengolah Sistem (*Process*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

h. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik.

II.2.2. Informasi

Di dalam pengolahan sistem pada akhirnya menghasilkan suatu informasi, untuk itu pendefenisian informasi diperlukan untuk menunjang berhasilnya pengembangan sistem yang akan dirancang. Informasi adalah data yang dapat diolah yang lebih berguna dan berarti bagi yang menerimanya.

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya. Sumber informasi adalah data. Data kenyataannya yang menggambarkan suatu kejadian – kejadian dan kesatuan nyata. Kejadian – kejadian (*event*) adalah kejadian yang terjadi pada saat tertentu. [7].

II.2.3. Sistem Informasi

Sistem informasi merupakan sistem yang disusun secara sistematis dan teratur dari jaringan-jaringan aliran informasi yang menghubungkan setiap bagian dari suatu sistem, sehingga memungkinkan diadakannya komunikasi antar bagian atau satuan fungsional. Berikut pengertian sistem informasi menurut para ahli adalah :

Sistem informasi merupakan sistem yang disusun secara sistematis dan teratur dari jaringan-jaringan aliran informasi yang menghubungkan setiap bagian

dari suatu sistem, sehingga memungkinkan diadakannya komunikasi antar bagian atau satuan fungsional. [8].

Sistem informasi ini terdiri dari kata sistem dan informasi. Sistem berasal dari bahasa Latin (*systema*) dan bahasa Yunani (*sustema*) adalah suatu kesatuan yang terdiri komponen atau elemen yang saling bekerja sama atau yang dihubungkan dengan cara-cara tertentu sehingga membentuk satu kesatuan untuk melaksanakan suatu fungsi guna mencapai suatu tujuan. Sedangkan kata informasi berasal dari kata Perancis kuno *informacion* (tahun 1387) yang diambil dari bahasa Latin *informationem* yang berarti “garis besar, konsep, ide”. Informasi merupakan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. [9].

II.2.4. Peramalan

Peramalan merupakan suatu keputusan tentang kemungkinan masa yang akan datang yang didasarkan fakta-fakta ekonomi sekarang dan sejarah masa lalu. Sebelum melakukan peramalan harus diketahui lebih dulu persoalan dalam pengambilan keputusan. Dengan adanya peramalan manajemen dapat segera menyiapkan langkah paling baik dalam mengatasi permasalahan dan dapat bersiap-siap memanfaatkan situasi jika terjadi perkembangan yang lebih baik dalam peramalan. Peramalan yang menjadi titik tolak bagi tindakan perusahaan adalah ramalan penjualan. Pengertian ramalan penjualan adalah proyeksi teknis permintaan konsumen yang potensial pada masa yang akan datang dengan menggunakan berbagai pendekatan. Hal ini sangat berhubungan dengan kegiatan

produksi. Dalam proses produksi kegiatan yang dilaksanakan dalam peramalan adalah untuk menentukan jumlah permintaan konsumen terhadap suatu produk yang dilakukan pada awal proses perencanaan dan untuk kepentingan proses pengendalian produksi. Oleh sebab itu perangkat lunak sangat dibutuhkan, untuk mendukung peramalan permintaan konsumen untuk mempercepat, mempermudah dan menghindari hitungan dengan tangan yang kemungkinan terjadi kesalahan. [10].

II.2.5. Metode *Single Exponential Smoothing*

Penghalusan *exponential* adalah teknik peramalan rata-rata bergerak dengan pembobotan dimana data diberi bobot oleh sebuah fungsi *exponential*. Penghalusan *exponential* merupakan metode peramalan rata-rata bergerak dengan pembobotan canggih, namun masih mudah digunakan. Metode ini sangat sedikit pencatatan data masa lalu. Rumus penghalusan *exponential* dapat ditunjukkan sebagai berikut :

$$F_{t+1} = \alpha X_t + (1-\alpha) F_{t-1} \dots\dots\dots(1)$$

Keterangan :

F_{t+1} = Ramalan untuk periode ke $t+1$

X_t = Nilai riil periode ke t

α = Bobot yang menunjukkan konstanta penghalus ($0 < \alpha < 1$)

F_{t-1} = Ramalan untuk periode ke $t-1$

Metode ini membutuhkan nilai *alpha* (α) sebagai nilai parameter pemulusan. Bobot nilai α lebih tinggi diberikan kepada data yang lebih baru,

sehingga nilai parameter α yang sesuai akan memberikan ramalan yang optimal dengan nilai kesalahan (*error*) terkecil. Untuk mendapatkan nilai α yang tepat pada umumnya dilakukan dengan *trial and error* (coba-coba) untuk menentukan nilai kesalahan terendah. Nilai α dilakukan dengan membandingkan menggunakan interval pemulusan antar $0 < \alpha < 1$, yaitu α (0,1 sampai dengan 0,9). Metode ini hanya mampu memberikan ramalan satu periode ke depan dan cocok untuk data yang mengandung unsur *stationer*. Karena jika diterapkan pada serial data yang memiliki *trend* yang konsisten, ramalan yang dibuat akan selalu berada dibelakang trend. Selain itu, metode eksponensial ini juga memberikan bobot yang relatif lebih tinggi pada nilai pengamatan terbaru dibanding nilai-nilai periode sebelumnya.[11].

II.2.8. Database

Konsep *database* Pangkalan data atau basis data (bahasa Inggris: *database*), atau sering pula dieja basisdata, 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.

Database merupakan suatu tempat untuk menyimpan data. Pada sebuah *database* bisa terdapat satu atau lebih tabel dan *query*. Operasi yang biasanya dilakukan *database* berhubungan erat dengan pengaksesan tabel atau *query*.

Selain itu juga dalam mengambil data dari *server* lain akan mengalami penurunan performa. Tetap dengan menggunakan terdistribusi, bisa dengan cepat melakukan akses untuk data pada *database server* yang didistribusikan. Sedangkan untuk tersentralisasi, karena databasenya hanya satu dan terpusat (misalnya di *head office*) maka seluruh *client* dari manapun akan mengambil data tersebut dari satu *database*. Dengan demikian data yang diambil tidak akan bermasalah dalam hal konsistensi karena berada dalam satu sumber, tetapi akan membutuhkan *hardware* yang jauh lebih besar dan *bandwidth* yang lebih tinggi. Hal ini dikarenakan server tersebut berfungsi untuk menampung penggunaan *connection* yang sangat banyak. [12].

II.2.6. Visual Studio 2010

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi *console*, aplikasi *Windows*, ataupun aplikasi *Web*. *Visual Studio* mencakup kompiler, *Software Development Kit* (SDK), *Integrated Development Environment* (IDE), dan dokumentasi (umumnya berupa *MSDN Library*). Kompiler yang dimasukkan ke dalam paket *Visual Studio* antara lain *Visual C++*, *Visual C#*, *Visual Basic*, *Visual Basic .NET*, *Visual InterDev*, *Visual J++*, *Visual J#*, *Visual Fox Pro*, dan *Visual Source Safe*.

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas *windows*)

ataupun *managed code* (dalam bentuk *Microsoft Intermediate Language* di atas *.NET Framework*). Selain itu, *Visual Studio* juga dapat digunakan untuk mengembangkan aplikasi *Silverlight*, aplikasi *Windows Mobile* (yang berjalan di atas *.NET Compact Framework*).

Visual Studio sebelumnya versi *Visual Studio 9.0.21022.08*, atau dikenal dengan sebutan *Microsoft Visual Studio 2008* yang diluncurkan pada 19 November 2007, yang ditujukan untuk *platform Microsoft .NET Framework 3.5*. Versi sebelumnya, *Visual Studio 2005* ditujukan untuk *platform .NET Framework 2.0 dan 3.0*. *Visual Studio 2003* ditujukan untuk *.NET Framework 1.1*, dan *Visual Studio 2002* ditujukan untuk *.NET Framework 1.0*. Versi-versi tersebut di atas kini dikenal dengan sebutan *Visual Studio .NET*, karena memang membutuhkan *Microsoft .NET Framework*. Sementara itu, sebelum muncul *Visual Studio .NET*, terdapat *Microsoft Visual Studio 6.0 (VS1998)*. [13].

II.2.5. SQL Server

SQL Server merupakan suatu *Relational Database Management Systems* (RDBMS) yang digunakan untuk menyimpan data. Data yang disimpan pada *database* bisa dalam skala kecil maupun besar. Selain itu, penyajiannya merupakan penyajian pada level fisik karena kita akan menyimpan langsung data pada *database* dengan kondisi yang sebenarnya, yaitu disimpan pada tabel apa, kolom mana, dan menggunakan tipe data saat penyimpanan.

Database merupakan suatu tempat untuk menyimpan data. Pada sebuah *database* bisa terdapat satu atau lebih tabel dan *query*. Operasi yang biasanya

dilakukan *database* berhubungan erat dengan pengaksesan tabel atau *query* Pada *SQL Server 2008* terdapat fitur-fitur yang dapat mengembangkan performa dari *database* tersebut. Beberapa fitur tersebut, yaitu :

- 1) *Date Data Type*: Digunakan untuk menyimpan data tanggal saja sehingga akan menghemat *space* pada *server*.
- 2) *Data Compression*: Digunakan untuk melakukan compress data sehingga ukuran data yang disimpan dalam hal *space hardisk* akan lebih kecil.
- 3) *Sparse Column*: Digunakan untuk menyimpan data yang memiliki lebih banyak data *NULL* dengan lebih efisien.
- 4) *Row Constructor*: Digunakan untuk melakukan insert beberapa data sekaligus dengan satu perintah *INSERT*.
- 5) *Table-Valued Parameter*: Digunakan untuk melakukan *parsing array* pada bahasa pemrograman, dimana satu variable diberikan data-data yang akan diproses setelahnya [14].

II.2.9. UML (*Unified Modelling Language*)

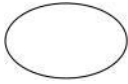
Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.[15]

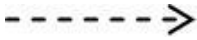
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

II.2.9.1. Use Case Diagram

Use case diagram merupakan pemodelan untuk melakukan (*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. Simbol-simbol yang digunakan dalam *Use Case Diagram* yaitu:

Tabel II.1. Simbol Use Case Diagram

Gambar	Keterangan
	<i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukaran pesan antara unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal nama <i>Use Case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mendefinisikan aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem biasa 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 mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidentifikasi bila aktor berinteraksi secara pasif dengan sistem

	<i>Include</i> , merupakan didalam <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 dipenuhi.

(Sumber :Ade Hendini, 2016)

II.2.9.2. Class Diagram

Class Diagram merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas dalam model desain dari suatu sistem juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Tabel II.2.Simbol Class Diagram

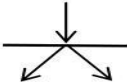
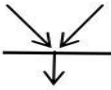
Multiplicity	Penjelasan
1	Satudanhanyasatu
0...*	Bolehtidakadaatau 1 ataulebih
1...*	1 ataulebih
0..1	Bolehtidakadamaksimal 1
n..n	BatasanantaraContoh 2..4 mempunyaiarti minimal 2 maksimum 4

(Sumber :Ade Hendini, 2016)

II.2.9.3. 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.3. Activity Diagram

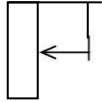
Gambar	Keterangan
	<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End Point</i> , akhir aktifitas
	<i>Activities</i> menggambarkan suatu proses atau kegiatan bisnis.
	<i>Fork</i> (Percabangan) digunakan untuk menunjukan kegiatan yang dilakukan secara parallel atau untuk mengabungkan dua kegiatan parallel menjadi satu.
	<i>Join</i> (Penggabungan) atau rake, digunakan untuk menunjukan adanya dekomposisi.
	<i>Decision points</i> menggambarkan pilihan untuk pengambilan keputusan, true, false
	<i>Swimlane</i> pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa

(Sumber :Ade Hendini, 2016)

II.2.9.4. 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.4. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari system yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal system 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 kelas.
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.

(Sumber :Ade Hendini, 2016)