

BAB II

TINJAUAN PUSTAKA

II.1. Peneliti Terkait

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

Menurut Etri Pujiati (2017), yang berjudul “Peramalan Dengan Menggunakan Metode *Double Exponential Smoothing* Dari Brown (Studi Kasus: Indeks Harga Konsumen (IHK) Kota Samarinda)”. Bahwa dalam penelitian yang dilakukan oleh Etri berproses dalam peramalan analisis Indeks Harga Konsumen tertinggi untuk 3 bulan yang akan datang yaitu dari bulan Januari, Februari dan Maret 2019 mengalami peningkatan setiap bulannya. Adapun hasil penelitian yang dilakukan adalah mempermudah perusahaan dalam menentukan peramalan harga konsumen pada periode berikutnya.

Menurut Rudi Ariyanto (2017), yang berjudul “Penerapan Metode *Double Exponential Smoothing* Pada Peramalan Produksi Tanaman Pangan”. Pada penelitian oleh Rudi Ariyanto mengalami beberapa kendala yaitu dalam penentuan Jumlah produksi pangan yang tidak menentu menjadi suatu masalah bagi Badan Ketahanan Pangan (BKP) Provinsi Jawa Timur dalam menentukan kebijakan mendatang. Apabila prediksi tidak diketahui maka dalam pembangunan Ketahanan Pangan dipastikan kurang maksimal dan alhasil Indonesia melakukan impor karena belum bisa memenuhi kebutuhan pangan. Diperlukan suatu

peramalan dalam memprediksi produksi tanaman pangan di periode mendatang berdasarkan tahun dengan menggunakan data masa lalu.

Menurut Radiant Victor Imbar (2019) dengan judul “Aplikasi Peramalan Stok Barang Menggunakan Metode *Double Exponential Smoothing*”. Tujuan dari penelitian dari Radiant Victor Imbar 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.

Menurut Agus Purwanto (2017), dengan judul “Teknik Peramalan Dengan *Double Exponential Smoothing* Pada Distributor Gula”. Distributor gula di Denpasar PT. Larasati, dari hasil analisa terhadap data dan informasi di PT. Larasati terdapat keterbatasan dalam menentukan jumlah stok gula yang harus disediakan oleh perusahaan, sehingga perusahaan menentukan jumlah stok dari permintaan pelanggan saja. Oleh karena itu, dibuatkanlah suatu analisis perhitungan dengan sebuah teknik peramalan menggunakan metode *Double Exponential Smoothing*. Dengan metode ini akan menghasilkan sebuah perhitungan untuk persediaan stok gula di masa yang akan datang.

Menurut Nendang Kacikal (2019), yang berjudul “Sistem Peramalan Dan Monitoring Persediaan Obat Di Rspg Cisarua Bogor Dengan Menggunakan Metode *Double Exponential Smoothing* Dan *Reorder Point*”. Berdasarkan hasil pengujian, dapat disimpulkan bahwa metode *Double 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

Definisi sistem adalah kumpulan dari bagian-bagian yang bekerja sama untuk mencapai tujuan yang sama. Definisi sistem adalah sekumpulan objek-objek yang saling berelasi dan berinteraksi serta hubungan antar objek bisa dilihat sebagai suatu kesatuan yang dirancang untuk mencapai suatu tujuan. Sistem adalah penggabungan dari bagian-bagian atau komponen-komponen yang terpisah-pisah dan disatukan menjadi satu rangkaian dan menjadi suatu fungsi yang baru (Aris, 2019).

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 (Deppi Linda, 2019).

II.2.1.1. Karakteristik Sistem

Untuk memahami atau mengembangkan suatu sistem, maka perlu membedakan unsur-unsur dari sistem yang membentuknya. Berikut adalah karakteristik sistem yang dapat membedakan suatu sistem dengan sistem lainnya yaitu:

1. Batasan (*boundary*) adalah penggambaran dari suatu elemen atau unsur mana yang termasuk di dalam sistem dan mana yang di luar sistem.
2. Lingkungan (*environment*) adalah segala sesuatu di luar sistem, lingkungan yang menyediakan asumsi, kendala, dan *input* terhadap suatu sistem.
3. Masukan (*input*) adalah sumber daya (data, bahan baku, peralatan, energi) dari lingkungan yang dikonsumsi dan dimanipulasi oleh suatu sistem.
4. Keluaran (*output*) adalah sumber daya atau produk (informasi, laporan, dokumen, tampilan layar komputer, barang jadi) yang disediakan untuk lingkungan sistem oleh kegiatan dalam suatu sistem.
5. Komponen (*component*) adalah kegiatan-kegiatan atau proses dalam sistem yang mentransformasikan input menjadi bentuk setengah jadi (*output*). Komponen ini bisa merupakan subsistem dari sebuah sistem.
6. Penghubung (*interface*) adalah tempat di mana komponen atau sistem dan lingkungannya bertemu atau berinteraksi.
7. Penyimpanan (*storage*) adalah area yang dikuasai dan digunakan untuk penyimpanan sementara dan tetap dari informasi, energi, bahan baku, dan sebagainya. Penyimpanan merupakan suatu media penyangga di antara komponen tersebut bekerja dengan berbagai tingkatan yang ada dan

memungkinkan komponen yang berbeda dari berbagai data yang sama (Aris, 2019).

II.2.2. Informasi

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya yaitu:

1. Akurat (*Accurate*) berarti informasi harus bebas dari kesalahan-kesalahan dan tidak biasa 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 (Deppi Linda, 2019).

Informasi dapat diibaratkan sebagai darah yang mengalir didalam tubuh manusia, seperti halnya informasi di dalam sebuah perusahaan yang sangat penting untuk mendukung kelangsungan perkembangannya, sehingga terdapat alasan bahwa informasi sangat dibutuhkan bagi sebuah perusahaan. Akibat bila kurang mendapatkan informasi, dalam waktu tertentu perusahaan akan mengalami ketidakmampuan mengontrol sumber daya, sehingga dala mengambil keputusan-keputusan strategis sangat terganggu, yang pada akhirnya akan mengalami kekalahan dalam bersaing dengan lingkungan pesaingnya. Disamping itu, sistem

informasi yang dimiliki seringkali tidak dapat bekerja dengan baik. Masalah utamanya adalah bahwa sistem informasi tersebut terlalu banyak informasi yang tidak bermanfaat atau berarti sistem terlalu banyak data (Andri Kristanto, 2018).

II.2.3. Sistem Informasi

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.2.4. *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, 2019).

Adapun rumus dari perhitungan metode *Double Exponential Smoothing* adalah sebagai berikut (Cahyarizki Adi Utama, 2019)

1. Pemulusan Eksponensial Tunggal:

$$S'_t = \alpha X_t + (1 - \alpha) S'_{t-1} \quad \dots\dots\dots(1)$$

2. Pemulusan Eksponensial Ganda:

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

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

3. Pemulusan Trend:

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

4. Pemulusan Trend:

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

5. Ramalan

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

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* ($1/n$)

II.2.5. 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, 2019).

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 table (Neni Purwati dan Hendra Kurniawan, 2019).

II.2.6. Bentuk-bentuk Normalisasi

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

1. 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.

2. 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.

3. 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.

4. 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 transisi, dengan kata lain setiap atribut bukan kunci haruslah bergantung pada *primary key* secara menyeluruh (Ganda Yoga Swara, 2019).

II.2.7. PHP

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah *web* dan bisa digunakan pada dokumen HTML. PHP dirancang untuk dapat bekerja sama dengan *database server* dan dibuat sedemikian rupa sehingga pembuatan dokumen HTML yang dapat mengakses *database* menjadi begitu mudah. Tujuan dari bahasa *scripting* ini adalah untuk membuat aplikasi dimana aplikasi tersebut yang dibangun oleh PHP pada umumnya akan memberikan hasil pada *web browser*, tetapi proses secara keseluruhan dijalankan di *server* (Saipul Anwar, 2019).

II.2.8. MySQL

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 tersebut terhubung oleh suatu relasi terdefinisi yang memungkinkan *user* 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 (Saipul Anwar, 2019).

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

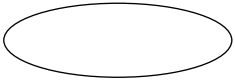
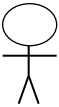

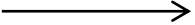

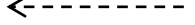
Hasil pemodelan pada OOAD (*Object Oriented Analysis And Design*) terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan 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:

II.2.9.1. *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, 2019).

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

Tabel II.1. Simbol *Use Case*




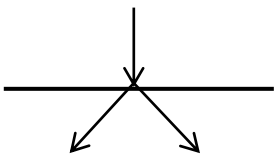
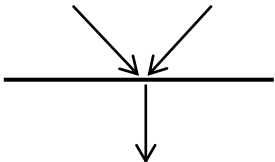
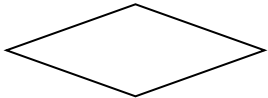

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang 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 : Ade Hendini : 2019)

II.2.9.2. 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

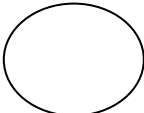
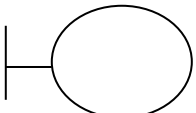

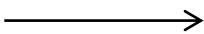
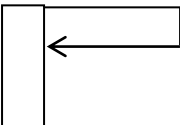


Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau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

(Sumber : Ade Hendini : 2019)

II.2.9.3. 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 : 2019)

II.2.9.4. *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, Relasi *Associations*, *Generalitation* dan *Aggregation*, atribut (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality* (Ade Hendini, 2019).

Tabel II.4. Tabel *Multiplicity Class Diagram*

<i>Multiplicity</i>	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 : 2019)