

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Dalam penelitian yang dilakukan memiliki bukti keaslian, dimana keaslian tersebut juga di bandingkan dengan penelitian – penelitian yang lainnya.

Penelitian yang dilakukan oleh Reyhan Dzickrillah Laksmana (2019) dengan judul Prediksi Penjualan Roti Menggunakan Metode Exponential Smoothing (Studi Kasus : Harum Bakery). Prediksi pada penelitian ini dilakukan menggunakan tiga metode Exponential Smoothing, yaitu Single Exponential Smoothing, Double Exponential Smoothing, dan Triple Exponential Smoothing. Evaluasi hasil prediksi dilakukan dengan menghitung nilai rata-rata error menggunakan metode Mean Absolute Percentage Error (MAPE). MAPE terkecil metode Single Exponential Smoothing didapatkan pada saat nilai $\alpha = 0,1$ dengan nilai MAPE sebesar 27,4039%, MAPE terkecil metode Double Exponential Smoothing didapatkan pada saat nilai $\alpha = 0,1$ dengan nilai MAPE sebesar 25,124%, dan MAPE terkecil metode Triple Exponential Smoothing didapatkan pada saat nilai $\alpha = 0,1$, $\beta = 0,1$, dan $\gamma = 0,4$ dengan nilai MAPE sebesar 25,303%. Dapat disimpulkan bahwa metode Double Exponential smoothing memiliki akurasi yang lebih baik dibandingkan dengan metode Single Exponential Smoothing dan Triple Exponential Smoothing pada Prediksi Penjualan Roti Studi Kasus Harum Bakery.

Penelitian selanjutnya yang dilakukan oleh Debby Santika (2020) dengan judul Penerapan Metode Triple Exponential Smoothing Dalam Prediksi Penanaman Bibit Kelapa Sawit Pada PT. Palmanco Inti Sawit. Kendala yang dihadapi oleh PT. Palmanco Inti Sawit adalah bagian lapangan harus mendata satu persatu hasil panen sawit yang terjadi sehingga kurang akurat dalam penentuan perhitungan prediksi penanaman bibit kelapa sawit, dan proses perhitungan prediksi pembibitan kelapa sawit masih menggunakan perhitungan sederhana yaitu dengan menggunakan aplikasi Microsoft excel, sering terjadi kesalahan dalam perhitungan prediksi dan jumlah penanaman bibit kelapa sawit tidak sinkron dengan data pembibitan kelapa sawit sesungguhnya, yang menjadi bahan penelitian penulis adalah data penanaman bibit sawit periode tahun 2019 sampai dengan April 2020. Dibutuhkan metode dalam perhitungan prediksi pembibitan kelapa sawit ke periode berikutnya sehingga dapat mempermudah bagian pembibitan dalam menentukan jumlah bibit yang akan diperlukan pada periode yang akan datang serta dalam pembuatan laporan prediksi pembibitan penanaman bibit kelapa sawit dapat diperoleh dengan efektif

Penelitian selanjutnya dilakukan oleh Riyan Asep Susanto (2022) dengan judul Model Perbandingan Metode Prediksi Jumlah Penjualan Produk Aplikasi HRIS Dengan Algoritma Forecasting Time Series Perusahaan SaaS. Berdasarkan hasil analisa yang telah kami lakukan terhadap penelitian pada prediksi Aplikasi HRIS Perusahaan SaaS, maka dapat ditarik suatu kesimpulan, sebagai berikut : Setelah melakukan pengujian dari beberapa metode prediksi pada penjualan seperti exponential smoothing, moving average dan arima maka hasil yang di

dapat metode ARIMA sangat cocok untuk prediksi penjualan Aplikasi HRIS dengan nilai MAPE untuk Produk HRIS A sebesar 4,507644 % dan nilai MAPE untuk Produk HRIS B sebesar 3,811256 %.

II.2. Landasan Teori

II.2.1. Sistem

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 (Eka Iswandy ; 2018 : 72).

II.2.2. Prediksi

Prediksi adalah suatu proses memperkirakan secara sistematis tentang sesuatu yang paling mungkin terjadi di masa depan berdasarkan informasi masa lalu dan sekarang yang dimiliki, agar kesalahannya (selisih antara sesuatu yang terjadi dengan hasil perkiraan) dapat diperkecil. Prediksi tidak harus memberikan jawaban secara pasti kejadian yang akan terjadi, melainkan berusaha untuk mencari jawaban sedekat mungkin yang akan terjadi (Herdianto, 2018 : 8).

Pengertian Prediksi sama dengan ramalan atau perkiraan. Menurut kamus besar bahasa Indonesia, prediksi adalah hasil dari kegiatan memprediksi atau meramal atau memperkirakan nilai pada masa yang akan datang dengan menggunakan data masa lalu. Prediksi menunjukkan apa yang akan terjadi pada

suatu keadaan tertentu dan merupakan input bagi proses perencanaan dan pengambilan keputusan.

Prediksi bisa berdasarkan metode ilmiah ataupun subjektif belaka. Ambil contoh, prediksi cuaca selalu berdasarkan data dan informasi terbaru yang didasarkan pengamatan termasuk oleh satelit. Begitupun prediksi gempa, gunung meletus ataupun bencana secara umum. Namun, prediksi seperti pertandingan sepakbola, olahraga, dll umumnya berdasarkan pandangan subjektif dengan sudut pandang sendiri yang memprediksinya

Prediksi yaitu suatu proses untuk memperkirakan suatu hal yang akan terjadi dimasa depan secara sistematis berdasarkan informasi yang didapat dari masa lalu. Prediksi dilakukan untuk mencari informasi sedekat dan semirip mungkin terhadap hal yang akan terjadi. Dalam memprediksi ada dua hal yang harus dilakukan yaitu: (Prasetyo, 2019)

1. Pembangunan model sebagai prototipe untuk disimpan dalam memori
2. Pembangunan model tersebut untuk pengenalan/prediksi pada suatu objek data lain agar diketahui kelas data tersebut berdasarkan model yang sudah disimpan

II.2.3. Metode *Winter Exponential Smoothing*

Metode ini digunakan ketika data menunjukkan adanya trend dan perilaku musiman. Untuk menangani musiman, telah dikembangkan parameter persamaan ketiga yang disebut metode “Holt- Winters” sesuai dengan nama penemunya.

Metode Holt-Winters sering disebut metode pemulusan eksponensial yang melakukan pendekatan. Metode Holt-Winter's *Exponential Smoothing* adalah metode prediksi dengan pendekatan pemulusan eksponensial berdasarkan hasil prediksi pada periode sebelumnya. Metode ini terbagi menjadi dua bagian yakni:

1. Metode Pemulusan Eksponensial Holt-Winters dengan Metode Perkalian Musiman (Multiplicative Seasonal Method) yang digunakan untuk variasi data musiman yang mengalami peningkatan/penurunan (fluktuasi),
2. Metode Pemulusan Eksponensial Holt-Winters dengan Metode Penambahan Musiman (Additive Seasonal Method) yang digunakan untuk variasi musiman yang bersifat konstan

Metode Holt-Winters didasarkan pada tiga persamaan pemulusan, yakni persamaan pemulusan keseluruhan, pemulusan tren, dan persamaan pemulusan musiman. Untuk Pemulusan Eksponensial Holt-Winters dengan Metode Perkalian Musiman mempunyai persamaan sebagai berikut:

Pemulusan Keseluruhan

$$S_t = \alpha \frac{x_t}{I_{t-L}} + (1 - \alpha)(S_{t-1} + b_{t-1})$$

Pemulusan Trend

$$b_t = \beta(S_t - S_{t-1}) + (1 - \beta)b_{t-1}$$

Pemulusan Musiman

$$I_t = \gamma \frac{x_t}{S_t} + (1 - \gamma)(I_{t-L})$$

Ramalan

$$F_{t+m} = (S_t + b_t m) I_{1-L+m}$$

(I Gusti Ngurah Lanang Septiadi Putra ; 2018 : 36)

II.2.3.1. Studi Kasus

Metode tersebut dihitung nilai *Mean Percentage Error* (MPE) dan *Root Mean Square Error* (RMSE), dari hasil perhitungan MPE dan RMSE yang terkecil merupakan nilai akurasi yang terbaik. Dari hasil prediksi dan perhitungan akurasi jumlah produksi rokok pada CV. Sukses Jaya Utama Tenganan dihasilkan perbandingan hasil MPE dan RMSE dari metode SES, *Holt* dan *Holt Winter* dan ditampilkan plot metode prediksi yang tepat sesuai dengan hasil MPE dan RMSE terkecil.

Perhitungan nilai uji akurasinya adalah: Perhitungan ME pada data SKM:

$$ME = \frac{19,489}{20,456} = 0,9527676$$

$$\text{Perhitungan MAE pada data SKM: } MAE = \frac{187,701}{20,456} = 9.175876$$

Perhitungan MPE pada data SKM:

$$MPE = \frac{388,428.961}{20,456} = 18.9885$$

Perhitungan MAPE pada data SKM: MAPE = = 86.71921%

$$\frac{1,773,928.1598}{20,456}$$

Tabel II.1. Analisis perbandingan nilai uji akurasi pada SKM antara ES, *Holt*, dan HW

Metode	ME	RMSE	MAE	MPE	MAPE	MASE
Holt winter	5.89899	16.08622	12.86158	18.98851	86.71921	NaN

Holt	5.376522	17.46891	14.16163	-29.9119	130.0378	NaN
ES	0.9527676	11.06828	9.175876	25.47472	89.30094	NaN

Sumber : I Gusti Ngurah Lanang Septiadi Putra ; 2018

Tabel II.2. Analisis perbandingan nilai uji akurasi pada SKT antara ES, *Holt*, dan HW

Metode	ME	RMSE	MAE	MPE	MAPE	MASE
Holt winter	7.960411	20.95657	16.51261	107.2144	158.9237	NaN
Holt	7.469058	20.87563	15.57709	60.72031	194.808	NaN
ES	-0.4868798	14.55605	12.34057	-14.56956	143.8908	NaN

Sumber : I Gusti Ngurah Lanang Septiadi Putra ; 2018

Ketepatan metode yang digunakan diukur dari akurasi yang mampu memprediksi data pada periode yang akan datang. Perbandingan tingkat akurasi prediksi hasil analisis antara metode ES, *Holt* dan *Holt winters* pada data produksirokok CV. Sukses jaya Utama Tenganan pada SKM dan SKT dari bulan Januari sampai Mei dapat dilihat pada tabel 1. Dari tabel 1 dan 2 dapat disimpulkan bahwa metode *Holt winter* merupakan metode yang paling akurat dibandingkan dengan metode *Holt* dan ES, karena memiliki nilai uji akurasi yang paling kecil.

Tabel II.3. Hasil prediksi SKM dengan menggunakan metode ES, *Holt* dan HW

No.	Data Aktual	ES	<i>Holt</i>	HW
1	11,057	19,604	35.56545	24.413286
2	20,456	19,604	37.13090	23.403274
3	14,266	19,604	38.69635	22.792969
4	5,500	19,604	40.26180	31.511672
5	15,731	19,604	41.82725	31.311398
6	15,110	19,604	43.39270	19.442965
7	30,304	19,604	44.95816	29.713108

Sumber : I Gusti Ngurah Lanang Septiadi Putra ; 2018

Tabel II.4. Hasil prediksi SKT dengan menggunakan metode SES, *Holt* dan HW

No.	Data Aktual	SES	<i>Holt</i>	HW
1	58,100	28.69977	27.99948	24.413286
2	58,100	28.69977	27.99896	23.403274
3	58,100	28.69977	27.99844	22.792969
4	58,100	28.69977	27.99792	31.511672

5	8,280	28.69977	27.99740	31.311398
6	8,280	28.69977	27.99688	19.442965
7	5,321	28.69977	27.99636	29.713108

Sumber : I Gusti Ngurah Lanang Septiadi Putra ; 2018

Berdasarkan Tabel 3 dan 4 metode *Holt Winter* cocok untuk memprediksikan jumlah produksi pada CV. Sukses Jaya Utama Tenganan pada masa yang akan datang sebab metode *Holt Winter* dapat meramalkan data dengan pola musiman dengan *trend* atau tanpa *trend*.

II.2.4. PHP

PHP adalah sebuah bahasa pemrograman *scripting* untuk membuat halaman *web* yang dinamis. Walaupun dikenal sebagai bahasa untuk membuat halaman *web*, tapi PHP sebenarnya juga dapat digunakan untuk membuat aplikasi command line dan juga GUI. *Website* yang dibuat menggunakan PHP memerlukan *software* bernama *webserver* tempat pemrosesan kode PHP dilakukan. *Serverweb* yang memiliki *software* PHP parser akan memproses input berupa kode PHP dan menghasilkan *output* berupa halaman *web*. PHP bersifat terbuka dan multiplatform, karenanya dapat dijalankan di banyak merek *webserver* (seperti apache dan IIS) (Ali Zaki, 2017)

II.2.5. MySQL

MySQL adalah suatu sistem manajemen basis data relasional (RDBMS-*Relational Database Management System*) yang mampu bekerja dengan cepat, kokoh, dan mudah digunakan. Contoh RDBMS lain adalah *Oracle*, *Sybase*. Basis data memungkinkan anda untuk menyimpan, menelusuri, menurutkan dan

mengambil data secara efisien. *Server MySQL* yang akan membantu melakukan fungsionalitas tersebut. Bahasa yang digunakan oleh MySQL tentu saja adalah *SQL-standar* bahasa basis data relasional di seluruh dunia saat ini.

MySQL dikembangkan, dipasarkan dan disokong oleh sebuah restoran garuda Swedia bernama MySQL AB. RDBMS ini berada di bawah bendera GNU GPL sehingga termasuk produk *Open Source* dan sekaligus memiliki lisensi komersial. Apabila menggunakan MySQL sebagai basis data dalam suatu situs *Web* (Antonius Nugraha Widhi Pratama, 2017).

II.2.6. Teknik Normalisasi

Normalisasi adalah suatu proses untuk membuat data yang tidak normal menjadi data yang normal. Bentuk data yang tidak normal / data mentah biasa disebut juga *unnormalized form*. Masing – masing level normalisasi mempunyai aturan tersendiri.

1. First Normal Form

Suatu tabel dikatakan dalam keadaan *first normal form* (1NF) jika :

- a. Tidak ada perulangan record data dalam tabel.
- b. Setiap sel memiliki satu nilai saja. Artinya tidak ada perulangan group dan array.
- c. Data yang diinputkan memiliki tipe data yang sama dengan tipe data kolom dalam tabel.

2. Second Normal Form

Suatu tabel dikatakan dalam keadaan *Second Normal Form* (2NF) jika tabel tersebut sudah dalam keadaan *First Normal Form* (1NF) dan jika semua atribut yang bukan kunci tabel, baik *primary key* maupun *foreign key* tergantung pada semua kunci dalam tabel.

3. *Third Normal Form*

Suatu tabel dikatakan dalam keadaan *third normal form* (3NF) jika tabel tersebut sudah dalam keadaan *second normal form* (2NF) dan jika tidak terdapat ketergantungan yang transitif. Artinya, data-data yang mungkin diisi berulang-ulang dapat dibuat sebuah tabel baru.

4. *Boyce-Codd Normal Form* (BCNF)

Tabel dikatakan dalam keadaan *boyce-codd normal form* (BCNF) jika tabel tersebut dalam keadaan *third normal form* (3NF) dan setiap determinan adalah kunci kandidat.

5. *Fourth Normal Form* (4NF)

Suatu tabel dikatakan dalam keadaan *fourth normal form* (4NF) jika tabel tersebut dalam keadaan *boyce-codd normal form* (BCNF) dan jika tidak terdapat ketergantungan nilai ganda.

6. *Fiveth Normal Form* (5NF)

Tabel dikatakan dalam keadaan *Fiveth Normal Form* (5NF) jika tabel tersebut dalam keadaan *fourth normal form* (4NF) dan jika setiap ketergantungan dalam join ada pada tabel sudah konsekuen dengan kunci kandidat pada tabel tersebut (Ema Utami ; 2015 : 73-76).

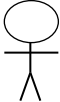
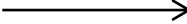
II.2.7. Unified Modelling Language (UML)

Unified Modeling Language (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (Indra Griha Tofik Isa ; 2017 : 141)

a) *Use case Diagram*

Use case diagram digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut (*user*). sehingga pembuatan *use case diagram* lebih dititik beratkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Sebuah *use case diagram* mempresentasikan sebuah interaksi antara aktor dengan sistem.

Tabel II.5. Simbol *Use case*

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukan 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 : Indra Griha Tofik Isa ; 2017 : 141)

b) *Class Diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Tabel II.6. *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

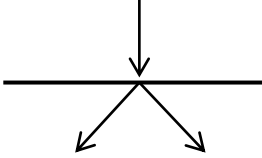
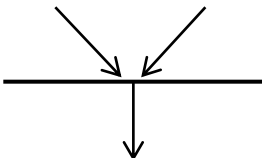
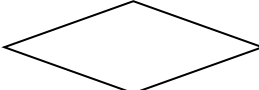
(Sumber : Indra Griha Tofik Isa ; 2017 : 141)

c) *Activity Diagram*

Activity Diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity Diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity Diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-trigger oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *Activity Diagram* tidak

menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Tabel II.7. 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 parallel 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 : Indra Griha Tofik Isa ; 2017 : 141)

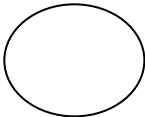
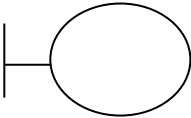
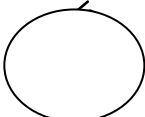
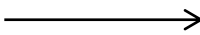
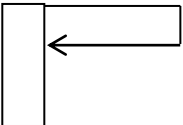
d) Sequence Diagram

Sequence Diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence Diagram* terdiri atas dimensi vertikal

(waktu) dan dimensi horizontal (objek-objek yang terkait).

Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*.

Tabel II.8. 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 : Indra Griha Tofik Isa ; 2017 : 141)