

BAB I

PENDAHULUAN

1.1 Latar Belakang

Beras merupakan salah satu komoditas pangan utama yang menjadi bahan makanan pokok masyarakat di Kota Medan. Seperti halnya komoditas pangan lainnya, beras mengalami fluktuasi harga dari waktu ke waktu. Kenaikan harga beras akan membawa dampak yang signifikan dalam pemenuhan bahan makanan pokok bagi masyarakat. Jika harga melambung akan menyulitkan masyarakat membeli atau mendapatkan persediaan beras dan disisi lain juga akan berpengaruh pada kenaikan komoditas lainnya. Sehingga diperlukan sebuah sistem atau model yang dapat memprediksikan kenaikan harga beras di masa akan datang.

Menurut Nur Hadiansyah (Nur Hadiansyah , 2017) Informasi fluktuasi harga cabai pada masa lalu dapat dijadikan variabel baru yang dapat dimasukkan dalam pemodelan time series, yang dikenal sebagai volatilitas atau simpangan baku bersyarat terhadap waktu. Oleh karena itu pemodelan time series dapat digunakan dengan baik untuk memprediksi fluktuasi harga.

Salah satu metode pemodelan time series yang banyak digunakan untuk melakukan prediksi harga adalah Metode Regresi Linear Sederhana. Hasibuan & Musthofa (Hasibuan & Musthofa, 2022) menggunakan Metode Regresi Linear Sederhana untuk memprediksi harga beras di Kota Padang, dari penelitian tersebut diperoleh akurasi sebesar 96,556% dengan *error RMS* sebesar 0,126.

Berdasarkan latar belakang diatas, Metode Regresi Linear Sederhana memiliki tingkat akurasi yang tinggi dan tingkat error yang rendah, sehingga

dapat digunakan untuk membangun sistem prediksi harga. Oleh karena itu, peneliti tertarik untuk menggunakan Metode Regresi Linear Sederhana dalam penelitian dengan judul **Implementasi Metode Regresi Linear Sederhana untuk Prediksi Harga Beras di Kota Medan.**

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dipaparkan sebelumnya, maka dapat dirumuskan permasalahan sebagai berikut :

1. Bagaimana merancang sistem yang mampu memprediksi harga beras Di Kota Medan ?
2. Bagaimana menerapkan metode Regresi Linear Sederhana untuk memprediksi harga beras Di Kota Medan ?
3. Bagaimana akurasi metode Regresi Linear dalam memprediksi harga beras Di Kota Medan ?

1.3 Batasan Masalah

Batasan-batasan masalah yang digunakan dalam melakukan penelitian ini antara lain, yaitu :

1. Penelitian ini menggunakan data harga beras harian di Kota Medan pada periode Bulan Januari 2021 hingga Bulan Desember 2022.
2. Sistem yang dibangun menggunakan bahasa pemrograman *PHP* dan *MySQL* sebagai basis data (*Database*).
3. Algoritma yang digunakan dalam penelitian ini adalah Regresi Linear Sederhana.

4. Pengujian akurasi menggunakan *Root Square Mean Error (RSME)*.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah dijabarkan, penelitian ini mempunyai tujuan sebagai berikut :

1. Untuk merancang sistem yang mampu memprediksi untuk memprediksi harga beras di Kota Medan
2. Untuk menerapkan metode Regresi Linear Sederhana untuk memprediksi harga beras di Kota Medan
3. Untuk menguji akurasi metode regresi linear dalam memprediksi harga beras di Kota Medan ?

1.5 Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini antara lain, yaitu:

1. Dengan dibangun sistem dapat membantu masyarakat umum untuk mengetahui dan memprediksi pergerakan harga beras di Kota Medan di masa mendatang.
2. Dapat digunakan sebagai bahan referensi untuk penelitian di bidang prediksi harga bahan pangan khususnya beras.
3. Hasil dari perancangan ini akan menambah wawasan dan pengetahuan penulis mengenai sistem prediksi.

1.6 Sistematika Penulisan

Sistematika penulisan pada skripsi ini adalah sebagai berikut :

BAB I : Pendahuluan, pada bab ini berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

BAB II : Tinjauan Pustaka, pada bab ini berisi tentang penjelasan secara singkat mengenai teori yang digunakan dalam penelitian serta teori pendukung dalam penelitian.

BAB III : Metodologi Penelitian, pada bab ini berisi tentang tahap-tahap penelitian serta pengumpulan data, perancangan aplikasi menerapkan algoritma regresi linear sederhana.

BAB IV : Hasil Dan Pembahasan, pada bab ini berisi mengenai tampilan dari hasil aplikasi sistem pakar dengan penerapan algoritma regresi linear sederhana.

BAB V : Kesimpulan Dan Saran, pada bagian akhir akan dipaparkan kesimpulan secara keseluruhan pada bab-bab sebelumnya dan juga akan berisi saran atau masukan dalam rangka pengembangan penelitian ini lebih lanjut dikemudian hari.

BAB II

TINJAUAN PUSTAKA

2.1 Prediksi

Prediksi (*prediction*) adalah memperkirakan nilai-nilai data bertipe apa saja dan kapan saja (masa lalu, sekarang, dan masa depan). Terdapat satu istilah yang mirip dengan prediksi, yaitu peramalan (*forecasting*) adalah memperkirakan nilai-nilai data time series dimasa depan (Suyanto, 2018).

Peramalan adalah seni dan ilmu untuk memprediksi kejadian di masa depan dengan melibatkan pengambilan data historis dan memproyeksikannya ke masa mendatang dengan model pendekatan sistematis (Indarwati et al.,2019).

Menurut indarwati dkk. (Indarwati et al.,2019) terdapat tiga tahapan yang harus dilalui dalam perancangan suatu metode peramalan, yaitu :

1. Melakukan analisa pada data masa lampau.

Langkah ini bertujuan untuk mendapatkan gambaran pola dari data bersangkutan.

2. Memilih metode yang akan digunakan.

Terdapat bermacam-macam metode yang tersedia dengan keperluannya. Pemilihan metode dapat mempengaruhi hasil ramalan. Hasil ramalan diukur dengan menghitung error atau kesalahan terkecil. Oleh karena itu, tidak ada metode peramalan yang pasti baik untuk semua jenis data.

3. Proses transformasi dari data masa lampau dengan menggunakan metode yang dipilih Apabila diperlukan maka diadakan perubahan sesuai kebutuhannya

2.2 Regresi Linear Sederhana

Metode regresi merupakan sebuah metode statistik yang melakukan prediksi menggunakan pengembangan hubungan matematis antara variabel, yaitu variabel dependen (Y) dengan variabel independen (X). Variabel dependen merupakan variabel akibat atau variabel yang dipengaruhi, sedangkan variabel independen merupakan variabel sebab atau variabel yang mempengaruhi. Prediksi terhadap nilai variabel dependen dapat dilakukan jika variabel independennya diketahui. Umumnya penjualan atau permintaan suatu produk dinyatakan sebagai variabel dependen yang besar atau nilainya dipengaruhi oleh variabel independen.

Regresi linear menjadi salah satu metode yang dipergunakan dalam produksi untuk melakukan peramalan atau prediksi tentang karakteristik kualitas maupun kuantitas. Hal ini dikarenakan dengan memperkirakan berbagai kombinasi produk, perusahaan dapat memaksimalkan keuntungan serta memperkirakan jumlah produksi yang tepat. Rumus untuk Regresi Linear dengan metode kuadrat terkecil atau sederhana adalah:

$$a = \frac{\sum y_i - b \sum x_i}{n} \dots\dots\dots (2.1)$$

$$b = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2} \dots\dots\dots (2.2)$$

$$y = a + bx \dots\dots\dots (2.3)$$

dimana y adalah kuantiti Harga, x adalah periode penjualan atau bulan penjualan, a adalah konstanta yang menunjukkan besarnya nilai y apabila $x = 0$, dan b adalah besaran perubahan nilai y

2.3 Pengujian Akurasi

Menurut Hasibuan dan Musthofa (Hasibuan & Musthofa, 2022), banyak metode yang dapat digunakan untuk menghitung besarnya *error* dalam melakukan prediksi, beberapa diantaranya, yaitu *Mean Square Error (MSE)*, *Root Mean Square Error (RMSE)*, dan *Mean Absolute Percentage Error (MAPE)*. *MSE* merupakan rata-rata selisih kuadrat antara nilai yang diprediksikan dengan diamati, *RMSE* merupakan akar dari *MSE*, dan *MAPE* merupakan rata-rata diferensial absolut antara nilai yang diprediksi dan actual. Rumus untuk menghitung nilai kesalahan tersebut adalah sebagai berikut:

$$MSE = \frac{1}{n} \sum (Y_t - Y'_t)^2 \dots\dots\dots (2.4)$$

$$RMSE = \sqrt{\frac{1}{n} \sum (Y_t - Y'_t)^2} \dots\dots\dots (2.5)$$

$$MAPE = \frac{1}{n} \sum_{t=1}^n \frac{|Y_t - Y'_t|}{Y_t} \dots\dots\dots (2.6)$$

Dimana,

Y_t = nilai aktual

Y'_t = nilai peramalan periode t

n = banyaknya periode

Menurut Sumari et al. (2021) kriteria interpretasi prediksi sangat akurat jika nilai *MAPE* kurang dari 10%. Untuk nilai kriteria lainnya berdasarkan *MAPE*

dapat dilihat pada Tabel 2.1. Sedangkan untuk MSE dan RMSE semakin rendah nilainya maka semakin baik prediksi yang dilakukan (Hayuningtyas & Sari; Hasibuan & Musthofa, 2022).

Tabel 2.1 Kriteria Interpretasi *MAPE*

<i>MAPE</i>	Kriteria Interpretasi
<10%	Prediksi Sangat Akurat
10%-20%	Prediksi Baik
20%-50%	Prediksi Masuk Akal
>50%	Prediksi Tidak Akurat

Sumber: Sumari et al. (2021)

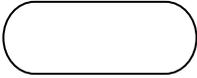
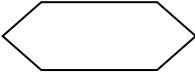
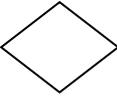
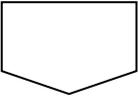
2.4 Flowchart

Menurut Riyowati dan Fadlilah (2019) *flowchart* adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah dan salah satu cara untuk menyajikan suatu algoritma. *Flowchart* menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian.

Secara umum *flowchart* merupakan sebuah bagan atau diagram yang menunjukkan langkah-langkah proses (intruksi) suatu program melalui simbol-simbol grafis dengan urutan tahapannya dihubungkan dengan tanda panah. *Flowchart* atau diagram alir ini merupakan sebuah olustrasi atau penggambaran penyelesaian masalah suatu rancangan sistem. *Flowchart* memiliki simbol-simbol

yang mempunyai arti masing-masing pada proses membangun sistem. Sebagai analisis sistem, *Flowchart* ini dijadikan sebagai dasar untuk mengajukan sebuah program kepada *programmer* atau *developer*. Berdasarkan adanya *flowchart* maka *programmer* mampu membaca dan memahami program sesuai dengan permintaan analisis sistem.

Tabel 2.2 Simbol *Flowchart*

No.	Simbol	Nama	Fungsi
1		<i>Terminator</i>	Permulaan/akhir program
2		<i>flow line</i>	Arah aliran program
3		<i>Preparation</i>	Proses inialisasi/pemberi harga awal
4		Proses	Proses perhitungan/proses pengolahan data
5		<i>Input/output Data</i>	Proses input/output data, parameter, informasi
6		<i>Decision</i>	Perbandingan pernyataan penyeleksian data yang memberikan pihak untuk langkah selanjutnya
7		<i>On page Connector</i>	Penghubung bagian-bagian <i>flowchart</i> yang berada pada satu halaman
8		<i>Off page Connector</i>	Penghubung bagian-bagian <i>flowchart</i> yang berada pada halaman berbeda

2.5 Pemrograman Web

Menurut Putra et al. (2019) *Web* terdiri dari dua macam, yaitu dinamis dan statis, *web* statis adalah *web* yang perubahan pada dokumen *web* tidak bisa dilakukan secara langsung sesuai kebutuhan (dinamis) disebabkan oleh teknologi yang digunakan tidak mendukung untuk melakukan perubahan pada dokumen *web* secara cepat. Teknologi yang digunakan adalah HTML. *Web* Dinamis adalah kebalikan dari *web* statis, kontennya dapat berubah-ubah setiap saat secara dinamis. Banyak teknologi yang digunakan untuk membangun *web* dinamis, seperti *HTML*, *Javascript*, *CSS* dan *PHP*, dan basis data seperti *MySQL* diperlukan untuk menyimpan data.

2.5.1 Hyper Text Markup Language

Menurut Endra dan Aprilita (2018), *Hypertext Markup Language (HTML)* merupakan salah satu bahasa yang biasa digunakan oleh pengguna dalam membuat tampilan yang digunakan oleh *Web Application*. *HTML* merupakan bahasa dasar pembuatan *web*. *HTML* menggunakan tanda (*mark*), untuk menandai bagian-bagian dari *text*. *HTML* disebut sebagai bahasa dasar, karena dalam membuat *web*, jika hanya menggunakan *HTML* maka tampilan *web* terasa hambar (Rerung, 2018:18).

Berdasarkan pengertian di atas, maka dapat disimpulkan bahwa HTML merupakan bahasa pemrograman yang digunakan pada dokumen *web* atau bahasa

standar untuk menyebarkan informasi pada *web* dan menampilkan halaman *web* dimana saja serta bersifat statis. Struktur dasar lengkap pada HTML sangat mudah dimengerti seperti berikut :

```
<!DOCTYPE HTML>

<html lang="en-US">

    <head> <meta charset="UTF-8">

        <title>

        </title>

    </head>

    <body>

    </body>

</html>
```

2.5.2 Cascading Style Sheet

Cascading Style Sheet (CSS) adalah suatu bahasa *style sheet* yang digunakan untuk mendesain sebuah halaman website dengan menggunakan penanda yang dikenal dengan sebutan *id* dan *class*. Penggunaan yang paling umum dari *CSS* adalah untuk memformat halaman *web* yang ditulis dengan *HTML* dan *XHTML*. Kode *CSS* dapat disisipkan ke dalam sebuah dokumen *HTML* dan juga bisa diletakkan pada dokumen *CSS* tersendiri yang berekstensi *.css* kemudian dipanggil menggunakan kode *HTML* (Limbong dan Sriadhi, 2021).

2.5.3 PHP Hypertext Preprocessor

Menurut Sahi (2020) *PHP* merupakan singkatan dari *PHP Hypertext Preprocessor* yang digunakan sebagai bahasa *script server-side* dalam pengembangan *web* yang disisipkan pada dokumen *HTML*. *PHP* dapat di-

download secara bebas dari situs resminya <http://www.php.net>. Kelebihan dari *PHP*, yaitu :

- a. Tidak melakukan sebuah proses kompilasi dalam penggunaannya.
- b. Didukung oleh banyak *Web Server* seperti *apache*, *IIS*, *Lightpd*, *nginx*, hingga *Xitami* dengan konfigurasi lebih mudah.
- c. Pengembangan lebih mudah, karena banyaknya milis-milis dan *developer* yang siap membantu pengembangan.
- d. Mudah dipelajari dan dipahami karena memiliki referensi yang banyak.
- e. Bersifat *open source* sehingga dapat digunakan di beberapa mesin (*Linux*, *Unix*, *Macintosh*, *Windows*) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

2.5.4 Javascript

Javascript adalah bahasa yang berbentuk kumpulan skrip yang pada fungsinya berjalan pada suatu dokumen *HTML*. *Javascript* memberikan kemampuan tambahan terhadap bahasa *HTML* dengan mengizinkan pengekseskuan perintah-perintah di sisi *user*, yang artinya di sisi *browser* bukan di sisi *server web*. *Javascript* bergantung kepada *browser* yang memanggil halaman *web* yang berisi skrip-skrip dari *Javascript* dan tentu saja terselip di dalam dokumen *HTML* (Sahi, 2020).

2.5.5 XAMPP

XAMPP adalah perangkat lunak bebas, yang sangat mendukung banyak system operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai

server yang berdiri sendiri (*localhost*), yang terdiri dari program *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. Nama *XAMPP* merupakan singkatan dari *X* (empat sistem operasi apapun), *Apache*, *MySQL*, *PHP* dan *Perl*. Program ini tersedia dalam *GNU General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis (Sidik, 2018).

2.5.6 MySQL

MySQL adalah sistem manajemen database *SQL Open Source* yang paling populer, dikembangkan, didistribusikan, dan didukung oleh *Oracle Corporation*. Menurut *DB-Engine*, *MySQL* menempati urutan kedua sebagai *database* terpopuler setelah *Oracle Database*. *MySQL* mendukung banyak aplikasi yang paling banyak diakses, termasuk *Facebook*, *Twitter*, *Netflix*, *Uber*, *Airbnb*, *Shopify*, dan *Booking.com*. Kata “*MySQL*” dieja secara resmi dalam bahasa Inggris sebagai berikut “*My ess-cue-el*”. Kata “*MySQL*” kadang hanya dieja “*my sequel*” saja dan itu sudah menjadi sebuah kebiasaan. Kelebihan utama *MySQL* antara lain: Kemudahan penggunaa, Keandalan, Skalabilitas, Kinerja, Ketersediaan tinggi, Keamanan, Fleksibilitas (Anonim, 2022).

2.6 Unified Modelling Language (UML)

Menurut Rosa dalam Irmayani & Susyati (2017) “*UML* adalah salah satu standar bahasa yang banyak digunakan didunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”.

2.6.1 Use Case Diagram

Sebuah *Use Case diagram* menyatakan visualisasi interaksi yang terjadi antara pengguna (aktor) dengan sistem. Diagram ini bisa menjadi gambaran yang bagus untuk menjelaskan konteks dari sebuah sistem sehingga terlihat jelas batasan dari sistem. Ada 2 elemen penting yang harus digambarkan, yaitu aktor dan *Use Case*.

Aktor adalah segala sesuatu yang berinteraksi langsung dengan sistem, bisa merupakan orang (yang ditunjukkan dengan perannya dan bukan Namanya /personilnya) atau sistem komputer yang lain. Aktor dinotasikan dengan simbol gambar orang-orangan (*stick-man*) dengan nama kata benda di bagian bawah yang menyatakan peran/ sistem. Aktor bisa bersifat primer, yaitu yang menginisiasi berjalannya sebuah *Use Case*, atau sekunder, yaitu yang membantu berjalannya sebuah *Use Case*.

Use Case dinotasikan dengan simbol elips dengan nama kata kerja aktif di bagian dalam yang menyatakan aktivitas dari perspektif aktor. (Kurniawan, 2018)

Tabel 2.3 *Use Case Diagram*

No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Menspesifikasikan himpunan peran pengguna ketika berinteraksi dengan <i>use case</i> .

2		<i>Depedency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
3		<i>Generalizati on</i>	Hubungan dimana objek (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>accentor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara (<i>ancentor</i>).
5		<i>use case</i>	System yang menyediakan unit – unit yang saling bertukar pesan antar unitatau actor

Sumber: (Heriyanto, 2018)

2.6.2 Activity Diagram

Activity Diagram menggambarkan *work flow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas dapat dilakukan oleh sistem. (Heriyanto, 2018)

Tabel 2.4 *Activity Diagram*

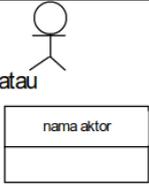
No	Gambar	Nama	Keterangan
1		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.

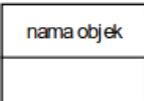
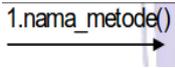
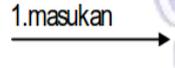
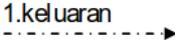
2		<i>Final Node</i>	Bagaimana objek dibentuk atau dihancurkan.
4		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

2.6.3 Sequence Diagram

Sequence diagram merupakan diagram yang menggambarkan perilaku objek pada *use case* yang mendeskripsikan *life time* (waktu hidup) dari objek dan pesan yang dikirimkan dan diterima oleh antar objek (Andani, 2019). Sequence diagram memiliki Komponen-komponen yang dapat dilihat pada Tabel 2.5

Tabel 2.5 Komponen *Sequence diagram* (Putri, 2020)

No.	Komponen	Nama	Keterangan
1.		Aktor (<i>Actor</i>)	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri

No.	Komponen	Nama	Keterangan
2.		Garis Hidup (<i>Lifeline</i>)	Menyatakan kehidupan suatu objek
3.		Objek (<i>Object</i>)	Menyatakan objek yang berinteraksi pesan.
4.		Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif
5.		Pesan tipe	Menyatakan suatu objek membuat objek
		<i>create</i>	yang lain, arah panah mengarah pada objek yang dibuat.
6.		Pesan tipe <i>call</i>	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain dengan dirinya sendiri.
7.		Pesan tipe <i>send</i>	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.		Pesan tipe <i>return</i>	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode

No.	Komponen	Nama	Keterangan
			menghasilkan suatu kembalian ke objek tertentu.
9.		Pesan tipe <i>destroy</i>	Menyatakan suatu objek mengakhiri hidup objek yang lain.