

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Prediksi**

Memprediksi nilai data jenis apa pun dan kapan pun (masa lalu, sekarang, dan masa depan) adalah inti dari prediksi. Peramalan adalah ungkapan yang mirip dengan prediksi yang memperkirakan nilai data deret waktu di masa depan. (Puteri & Silvanie, 2020).

Prediksi adalah upaya mengantisipasi atau memperkirakan sesuatu yang akan terjadi di masa depan dengan menggunakan berbagai pengetahuan sejarah yang bersangkutan dalam pendekatan ilmiah. Tujuan dari prediksi adalah untuk mengetahui apa yang akan terjadi di masa depan dengan peluang terjadinya yang paling besar. Metode prediksi dapat dilakukan secara kualitatif melalui pendapat para pakar atau secara kuantitatif dengan perhitungan secara matematis. Salah satu metode prediksi kuantitatif adalah menggunakan analisis deret waktu (time series) (Wanto & Windarto, 2017).

Prediksi adalah alat penting untuk memprediksi kejadian di masa depan dengan mendeteksi pola kejadian di masa lalu. Prediksi dapat digambarkan sebagai suatu metode atau strategi untuk meramalkan atau memperkirakan suatu nilai di masa depan dengan menggunakan fakta atau informasi yang relevan baik dari masa lalu maupun masa kini. Manajer pemasaran akan memanfaatkan perkiraan penjualan untuk beberapa item untuk menetapkan tujuan penjualan yang harus dipenuhi di masa depan. (Monalisa et al., 2018).

#### **2.2 Bahan Pokok**

Bahan pokok adalah jenis kebutuhan bahan utama masyarakat yang terdiri atas berbagai bahan-bahan makanan dan juga minuman. Bahan pokok merupakan yang paling banyak yang dikonsumsi oleh masyarakat, menurut keputusan Menteri Industri dan perdagangan no. 115/MPP/KEP/2/1998 tanggal 27 februari 1998 dalam menjelaskan bahwa terdapat Sembilan bahan pokok yaitu beras, bawang, gula pasir, garam, susu, minyak goreng, telur, tepung terigu, dan kacang kedelai.

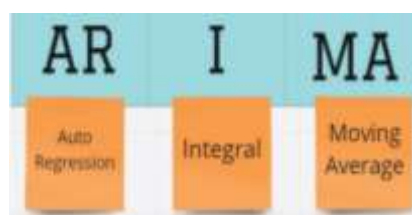
Bahan pokok akan mengalami kenaikan atau penurunan harga dalam rentang waktu tertentu. Salah satu kenaikan harga yang mempengaruhi perekonomian masyarakat adalah kenaikan harga kebutuhan pokok. yang biasanya naik sehingga harga bahan pokok biasanya seperti beras, bawang, gula pasir, garam, susu, minyak goreng, telur, tepung terigu, dan kacang kedelai, Dampaknya, komoditas pokok lainnya akan naik. Banyak orang mengeluh karena tidak mampu membeli makanan ketika harga sedang rendah. Masyarakat menengah ke bawah merasakan kenaikan harga bahan pokok, sedangkan masyarakat menengah ke atas tidak merasakan kenaikan harga karena masih mampu membelinya. Kemiskinan akan meningkat akibat kenaikan harga. Namun tampaknya pemerintah belum siap menghadapi situasi ini. Hal ini terjadi secara berkala, sehingga harga-harga meningkat karena kurangnya stok pangan, kekeringan, serangan serangga, distribusi yang tidak merata, dan banyaknya komoditas yang menumpuk, dan lain-lain. (Rochmaniah & Oktafia, 2019).

### 2.3 Autoregressive Integrated Moving Average (ARIMA)

*Autoregressive Integrated Moving Average (ARIMA)* adalah metode untuk yang memprediksikan pergerakan masa depan dengan memahami pengamatan yang masa lampau. namun harga bahan pokok harian dalam jangka pendek antara 1 hingga 30 hari kedepan. Bahwa ARIMA cocok digunakan dalam peramalan *time series* jangka pendek.

Pendekatan ini juga biasa digunakan untuk melakukan peramalan di berbagai sektor. Pendekatan ini digunakan untuk mengidentifikasi hubungan statistik antara variabel yang diramalkan dan nilai variabel, dengan mengabaikan variabel independen sepenuhnya saat menghasilkan prediksi. (Rasyidi, 2017).

Metode ini dapat memberikan informasi *trend* terdapat arah kenaikan atau penurunan pada suatu *dataset*.



Gambar 2.1 Komponen Model ARIMA (Ayu et al., 2019).

Menurut metode *Autoregressive interated moving average* (ARIMA) adalah metode analisi yang dikenal sebagai *BoxJenkins*. AR digunakan untuk dalam menentukan perubahan sejak terakhir kalinya, dalam metode ini berasal dari penggabungan antara model *Autoregressive* atau disinggakat (AR), dan *moving average* disinggakat (MA) untuk memperlancarkan tren dalam data, dan (I) untuk menghilangkan bentuk deret yang tidak bias di *stasioner*. Metode ini yang telah dikembangkan oleh George Box dan Gwilyn Jenkins.

Metode ARIMA Proses ini dibagi menjadi empat tahap: mencari teknik deret waktu, menghitung parameter metode alternatif, metode pengujian, dan memperkirakan nilai deret waktu Model sementara dievaluasi kembali dengan data historis untuk melihat apakah model sementara yang dihasilkan sudah memadai yang harus terpenuhi didalam memodelkan runtun waktu. Deret nonstasioner dapat ditransfoemasikan kedalam dengan cara pembedaan. Didalam waktu Rata-rata tidak konstan, varians tidak konstan, atau keduanya (rata-rata dan varians tidak konstan) dapat ditemukan dalam deret waktu. (Ayu et al., 2019).

Penelitian ini menggunakan model ARIMA untuk meramalkan harga komoditas fundamental nasional. Model ARIMA(p, d, q) terdiri dari tiga bagian: AR (autoregresi, orde p), MA (rata-rata bergerak, orde q), dan besarnya diferensiasi (integrasi) yang dilakukan (orde d). Diferensiasi adalah transformasi yang digunakan untuk membuat data stabil.

Model *Autoregressive Integrated Moving Average* (ARIMA) dapat dibangun dengan tiga tahapan yaitu identifikasi, estimasi parameter, dan pemeriksaan diagnostik Berikut ini langkah-langkah dalam menentukan ARIMA sebagai berikut : (Rasyidi, 2017)

1. Uji Stasioner Data
  - a. Menghitung rata-rata dilakukan dengan menjumlahkan seluruh nilai data penelitian kemudian dibagi dengan jumlah data. Dengan rumus sebagai berikut:

$$\bar{X} = \frac{1}{N} \sum_{i=1}^n X_i$$

keterangan

$\bar{x}$  merupakan rata-rata hitung

$x_i$  merupakan nilai sampel ke- $i$

$n$  merupakan jumlah sampel

- b. Setelah mean atau nilai rata rata didapatkan, maka selanjutnya dapat dilakukan dengan mencari nilai standart deviasi, standart deviasi digunakan untuk mencari nilai statatisik uji. Dengan rumus sebagai berikut:

$$SD = \sqrt{\frac{\sum (Y_t - \bar{Y})^2}{N}}$$



Keterangan :

SD = Standar Deviasi

$Y_t$  = Data ke –  $n$

$\bar{Y}$  = Nilai rata-rata (Mean)

$N$  = Banyak data

- c. Setelah mendapatkan nilai standart deviasi pada data awal, langkah selanjutnya adalah mengitung nilai Statistik Uji. Untuk mencari nilai Statistik Uji dapat menggunakan rumus:

$$Z = \frac{Xn - \mu_0}{\sigma / \sqrt{n}}$$

Keterangan:

$\mu_0$ : adalah hipotesis nol (nilai mean) yang akan diuji

$\sigma$ : adalah deviasi standar yang diberikan

$n$ : adalah jumlah sampel

$x_n$ : adalah rata-rata sampel

$z$ : adalah statistic yang diuji

- d. Selanjutnya menghitung diferensiasi, untuk menghitung difrensiasi data perlu dilakukan pengurangan pada data time series awal untuk mendapatkan nilai time series baru, data time series baru tersebut yang akan dilakukan kembali uji unit root testnya. Rumus difernsiasi dapat di sederhakan dengan rumus sebagai berikut:

$$Y_t = Y_n - Y_{n-1}$$

Keterangan :

$Y_t$  = differencing

$Y_n$  = nilai pada waktu-n

$Y_{n-1}$  = nilai pada waktu n-1

## 2. Analisis Plot Deret Waktu

- a) Menghitungnya ACF secara manual dapat digunakan persamaan rumus berikut ini Rumus Perkiraan Mean Sample:

$$\bar{y} = \sum_{t=1}^T y_t / T$$

Keterangan :

$\bar{y}$  = nilai rata rata

$Y_t$  = nilai pada waktu

- b) Setelah mendapatkan meannya, maka dapat dilanjutkan dengan menghitung ACF, persamaan Rumus ACF sebagai berikut, Rumus Autokorelasi (ACF):

$$ACF = \frac{\sum_{n=1}^{n-1} (Y_n - \bar{Y})(Y_{n-1} - \bar{Y})}{\sum_{n=1}^{n-1} (Y_n - \bar{Y})^2}$$

Keterangan:

$Y_n$  = nilai pada ke-n

$\bar{Y}$  = nilai rata-rata

- c) Setelah didapatkan nilai ACF, nilai ACF tersebut akan digunakan untuk melakukan perhitungan nilai PACF untuk mencari nilai PACF dapat menggunakan rumus berikut:

$$PACF(n) = (ACF(n) - PACF(n-1) * ACF(n-1)) / (1 - PACF(n-1)^2)$$

## 3. Identifikasi Model

Setelah model model yang memungkinkan di dapatkan, tahap selanjutnya adalah melakukan indentifikasi parameter model terbaik. Identifikasi dilakukan dengan cara mencari nilai MSE dengan fungsi MSE pada Gambar 4.12. Fungsi MSE digunakan untuk mengevaluasi performa model ARIMA dalam memprediksi

data time series. Nilai MSE dapat menilai akurasi model, Semakin kecil nilai MSE, semakin akurat prediksi model.

#### 4. Forecasting

- a. Setelah melakukan uji stasioneritas data, indentifikasi nilai plot ACF dan PACF dan indentifikasi model. maka didapat model Arima (3,1,1) Persamaan Model arima dapat ditulis dengan berikut:

$$Y_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + \epsilon_t - \theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-2} - \dots - \theta_q \epsilon_{t-q}$$

Keterangan:

$X_t$  adalah nilai dalam deret waktu pada waktu t

c adalah konstanta

$\phi_1, \phi_2, \dots, \phi_p$  adalah parameter autoregressive

$\theta_1, \theta_2, \dots, \theta_q$  adalah parameter moving average

$\epsilon_t$  adalah noise atau error pada waktu t

- b. Kemudian menghitung nilai Estimasi AR menggunakan persamaan rumus sebagai berikut:

$$S_{yy} = \sum_{t=1}^N (Y_t - \bar{Y})^2$$

Keterangan :

$S_{yy}$  adalah jumlah kuadrat deviasi total.

$Y_t$  adalah nilai pada waktu tt dalam deret waktu (time series).

$\bar{Y}$  adalah rata-rata dari seluruh nilai dalam deret waktu.

- c. Selanjutnya mencari nilai  $S_{y_t y_{t-1}}$  dengan menggunakan rumus berikut:

$$S_{y_t y_{t-1}} = \sum_{t=1}^N (Y_t - \bar{Y})(Y_{t-1} - \bar{Y})$$

Keterangan:

$S_{y_t y_{t-1}}$  adalah jumlah produk deviasi dari  $Y_t$  dan  $Y_{t-1}$ .

$Y_t$  adalah nilai pada waktu tt dalam deret waktu (time series).

$Y_{t-1}$  adalah nilai pada waktu t-1 dalam deret waktu.

$\bar{Y}$  adalah rata-rata dari seluruh nilai dalam deret waktu.

- d. Maka Estimasi parameter AR pertama dapat dihitung menjadi

$$\hat{\phi}_1 = \frac{S_{Y_t Y_{t-1}}}{S_{YY}}$$

- e. Selanjutnya dengan mencari estimasi parameter MA sebagai berikut Langkah-langkah untuk mencari  $\theta_1$  yang memaksimalkan log-likelihood function adalah sebagai berikut:

$$\theta_1 = \frac{\sum_{t=2}^n a_{t-1} a_t}{\sum_{t=2}^n a_{t-1}^2}$$

- f. Setelah seluruh nilai AR dan MA telah di hitung maka persamaan arima sebelumnya dapat di tuliskan menjadi:

$$Y_t = c + (-0,3762) X_{t-1} + (-0,3762) X_{t-2} + \dots + (-0,3762) X_{t-p} + \varepsilon_t - 0,3762 \varepsilon_{t-1}$$

### 2.3.1 Mean Squared Error

MSE (*Mean Squared Error*) adalah salah satu metrik evaluasi yang umum digunakan dalam statistika dan analisis deret waktu, termasuk dalam konteks metode ARIMA. Dalam konteks ARIMA (*AutoRegressive Integrated Moving Average*), MSE digunakan untuk mengukur sejauh mana model ARIMA memprediksi dengan akurat. MSE dihitung dengan cara mengambil selisih antara nilai aktual dan nilai yang diprediksi oleh model, kemudian mengkuadratkannya, dan akhirnya mengambil rata-rata dari kuadrat selisih tersebut. Rumus MSE dapat dituliskan sebagai berikut, dengan (n) sebagai jumlah observasi, (yt) sebagai nilai aktual, dan ( $\hat{y}_t$ ) sebagai nilai yang diprediksi.

$$MSE = \frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2$$

Dalam konteks ARIMA, langkah-langkah umumnya adalah:

1. Pemodelan ARIMA: Membangun model ARIMA yang sesuai dengan deret waktu yang akan diprediksi.
2. Pemisahan Data: Memisahkan data menjadi set pelatihan (training set) dan set pengujian (testing set).



3. Pelatihan Model: Melatih model ARIMA menggunakan data pelatihan.
4. Pengujian Model: Menggunakan model untuk membuat prediksi pada set pengujian.
5. Perhitungan MSE: Menghitung MSE untuk mengevaluasi seberapa baik model ARIMA memprediksi nilai pada set pengujian.

Semakin rendah nilai MSE, semakin baik model ARIMA dalam memprediksi data. Namun, perlu dicatat bahwa MSE dapat dipengaruhi oleh nilai ekstrem (*outliers*) dan mungkin tidak selalu memberikan gambaran yang lengkap tentang kualitas prediksi model. Oleh karena itu, selain MSE, juga berguna untuk mempertimbangkan metrik evaluasi lainnya dan melakukan analisis visual terhadap hasil prediksi. (Fejriani et al., 2020)

#### 2.4 Monte Carlo

Algoritma *Monte Carlo* adalah pendekatan numerik untuk menyelesaikan masalah matematika dengan banyak variabel yang sulit diselesaikan, seperti dengan kalkulus integral atau metode numerik lainnya. Algoritma *Monte Carlo* ini merupakan pada umumnya yang dilakukan oleh komputer dan memakai berbagai teknik simulasi komputer. Metode ini banyak digunakan salah satu metode komputasi ini untuk memecahkan berbagai persoalan metode *Monte Carlo*.

Simulasi *Monte Carlo* berasal dari statistik sampling, menggunakan bilangan acak sebagai input data dan probabilitas sebagai sarana untuk menyediakan masalah dunia nyata dengan simulasi. Metode *Monte carlo* memiliki struktur program dan proses simulasi yang fleksibel. Oleh karena itu, metode *Monte Carlo* dipilih sebagai metode simulasi untuk menentukan proses degradasi komponen/sistem.

Pengambilan sampel acak adalah dasar dari simulasi Monte Carlo. Pendekatan Monte Carlo digunakan ketika suatu sistem terdiri dari elemen-elemen yang melibatkan pertimbangan probabilitas. Simulasi Monte Carlo didasarkan pada eksperimen elemen probabilitas menggunakan sampel random (acak) (Geni, 2021). Model simulasi Monte Carlo merupakan salah satu jenis simulasi probabilistik dimana penyelesaian suatu permasalahan didasarkan pada proses pengacakan yang



melibatkan distribusi probabilitas variabel data berdasarkan data di atas serta distribusi probabilitas teoritis. Bilangan acak digunakan untuk menggambarkan kejadian acak dan berurutan yang terjadi akibat modifikasi simulasi. Angka acak memiliki ciri yang sama di setiap rangkaian angka acak yang dibuat, dan kemungkinan terjadinya angka acak tidak terpengaruh oleh angka di atas. Proyeksi permintaan ditentukan melalui simulasi dengan pendekatan Monte Carlo. (Hasugian et al., 2022).

Simulasi Monte Carlo digunakan untuk mengatasi masalah probabilitas dan statistik yang sulit. Simulasi Monte Carlo dijalankan selama prosedur. iterasi Pengulangan dan prosedur penghitungan tertentu digunakan untuk mencapai temuan yang semakin tepat. Simulasi Monte Carlo adalah jenis simulasi probabilistik lain yang menggunakan probabilitas untuk mengatasi masalah sistem data simple dan data random. Metode *Monte Carlo* Menggunakan variabel acak, banyak parameter yang digunakan dalam proses komputasi. Akibatnya, semakin banyak iterasi yang dilakukan, maka hasil yang diperoleh akan semakin spesifik atau tepat. Langkah-langkah penentuan Monte Carlo adalah sebagai berikut: (Rahayu, 2019).

1. Frekuensi data

Sebelum mendapatkan nilai distribusi, komulatif dan interval data, langkah pertama yang harus dilakukan adalah menghitung nilai frekuensi. untuk dapat mendapatkan nilai frekuensi hal yang dilakukan adalah mengamati data dan menghitung jumlah kemunculan harga dalam himpunan data tersebut.

2. Nilai Distributif, Komulatif dan Interval

Untuk mencari nilai distributif dapat menggunakan kumpulan nilai data (nilai individual atau nilai yang dikelompokkan dalam interval tertentu) diikuti dengan nilai frekuensi yang diperlukan. langkah yang dilakukan adalah dengan Rumus:

$$\text{Nilai Distributif Ke-}n = \text{Nilai frekuensi} / \text{Jumlah Seluruh data}$$

Setelah hasil nilai distributif telah didapatkan, maka langkah selanjutnya dapat dilakkukan dengan mencari nilai komulatif, nilai komulatif digunakan untuk penentuan rentang nilai intrveal dari suatu harga yang sudah dilakukan observasi sebelumnya. Untuk mencari nilai komulatif dapat digunakan rumus:

*Nilai Ke-n = Penjumlahan Komulatif sebelumnya + Nilai Distribusi ke-n+1*

### 3. Pembangkitan Angka Acak dan Peramalan

Langkah selanjutnya adalah melakukan pembangkitan nilai acak. Untuk dapat mencari nilai acak dapat digunakan Rumus:

$$\text{Bilangan Acak} = \text{rand}(\text{max}-\text{min}) * 100-1$$

Keterangan:

Max = Nilai tertinggi

Min = Nilai Terendah

Setelah angka random telah di dapatkan, kemudian sudah dapat dilakukan perbandingan bilangan random dengan data interval yang sudah didapatkan sebelumnya. Untuk menyeleksi nilai random dengan nilai interval yang telah di buat perlu dilakukan pengkelompokkan data secara manual dengan membandingkan hasil interval dari pembangkitan nilai acak dengan interval sebelumnya yang dihitung berdasarkan nilai komulatif sebelumnya.

#### 2.4.1 Mean Squared Error

*Mean Squared Error* (MSE) adalah metrik yang digunakan untuk mengukur sejauh mana hasil dari algoritma Monte Carlo mendekati nilai yang sebenarnya atau ekspektasi dari suatu variabel atau fenomena. MSE dihitung dengan mengambil perbedaan antara hasil Monte Carlo dan nilai sebenarnya, mengkuadratkan nya, dan kemudian mengambil rata-rata dari seluruh perbedaan kuadrat ini. MSE dapat dipecahkan menjadi dua komponen, yaitu varians (variance) dan bias. Varians mengukur sejauh mana hasil Monte Carlo tersebar di sekitar nilai sebenarnya, sedangkan bias mengukur sejauh mana hasil tersebut condong ke atas ke bawah dari nilai sebenarnya. MSE dapat menurun seiring dengan meningkatnya jumlah sampel (n) dalam algoritma Monte Carlo. Dengan kata lain, semakin banyak sampel yang digunakan, semakin kecil MSE yang dapat dicapai. Formula MSE dapat dinyatakan sebagai berikut:

$$MSE = \frac{1}{n} \times \sum (y_i - \hat{y}_i)^2$$

Dimana:

MSE : adalah *Mean Squared Error*

N : adalah jumlah sampel atau iterasi dalam algoritma Monte Carlo  
Yi : adalah jumlah nilai sebenarnya dari variabel atau fenomena  
Yi : adalah hasil perkiraan dari algoritma Monte Carlo pada iterasi ke-i

Tujuan utama dalam algoritma Monte Carlo adalah untuk meminimalkan MSE sehingga hasil estimasi lebih mendekati nilai sebenarnya. Semakin kecil nilai MSE, semakin baik kinerja algoritma Monte Carlo. (Dedrizaldi et al., 2019)

## 2.5 Sistem Pendukung Keputusan (SPK)

Sistem pendukung keputusan (DSS) adalah sistem yang membantu manajer mengambil keputusan tergantung pada kondisi tertentu semiterstruk. Sistem pendukung keputusan singkatan dari menjadi alat yang diperluas untuk pembuat keputusan keterampilan mereka, tetapi bukan sebagai ganti kebijaksanaan mereka. SPK dimaksudkan untuk keputusan yang menuntut penilaian algoritmik yang sepenuhnya tidak dapat didukung.

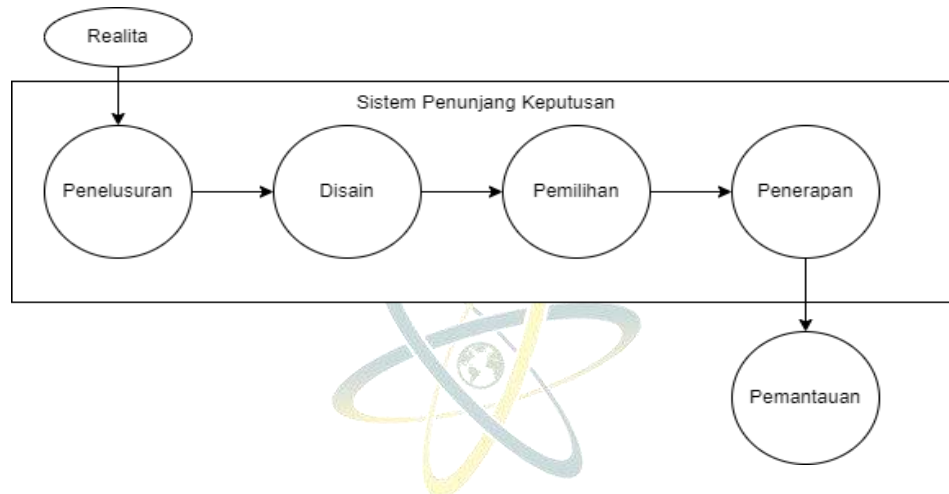
Sistem pendukung keputusan (SPK) ditujukan untuk membantu manajemen menganalisis situasi dengan sedikit struktur dan standar yang tidak jelas tidak akan menggunakan SPK otomatisasi pengambilan keputusan, melainkan memberikan alat interaktif yang memungkinkan pengambil keputusan melakukan analisis menggunakan model yang ada. (Hidayah & Fetrina, 2017).

## 2.6 Tahapan Sistem Pendukung Keputusan (SPK)

Berikut ini terdapat beberapa tahapan-tahapan proses dalam sistem pendukung keputusan, terdiri dari 4 yaitu:

1. Tindakan menemukan dan mengenali unsur-unsur yang menimbulkan kesulitan disebut pencarian (kecerdasan).
2. Disain (*desain*), Secara khusus, menciptakan berbagai strategi untuk mengatasi masalah tersebut.
3. Pemilihan (*choice*), memilih salah satu dari banyak desain yang mampu menyelesaikan masalah.

4. Implementasi, atau penerapan pendekatan yang dipilih pada suatu sistem yang menggunakan alat komputer sebagai alatnya (Na'am, 2017).



Gambar 2.2 Diagram Aliran Proses SPK (Na'azm, 2017)

## 2.7 Python

Python adalah bahasa pemrograman tingkat tinggi yang serbaguna dan mudah dipelajari. Bahasa ini menjadi sangat populer dalam dunia Machine Learning karena memiliki ekosistem yang kuat, beragam library, dan framework yang mendukung pengembangan model Machine Learning dengan mudah.



Gambar 2.3 Logo Python

Kelebihan Python:

1. Banyak Library Machine Learning: Python memiliki library yang kaya dan populer untuk Machine Learning, seperti scikit-learn, TensorFlow, Keras,

PyTorch, dan banyak lagi. Library-library ini menyediakan algoritma Machine Learning dan alat untuk pengolahan data yang efisien.

2. Sintaks yang Mudah Dibaca: Sintaks Python yang bersih, sederhana, dan intuitif membuat kode Machine Learning mudah dipahami, di-debug, dan dipelihara.
3. Komunitas Besar dan Aktif: Python memiliki komunitas Machine Learning yang besar dan aktif. Dukungan dari komunitas ini memungkinkan pengembang untuk memperoleh bantuan, berbagi pengetahuan, dan berkolaborasi dalam pengembangan solusi Machine Learning.
4. Interoperabilitas yang Baik: Python mudah diintegrasikan dengan bahasa pemrograman lain seperti C, C++, dan Java, sehingga memungkinkan penggunaan pustaka atau fungsi dari bahasa lain dalam proyek Machine Learning Python.
5. Pustaka Visualisasi yang Kuat: Python menyediakan library visualisasi yang kuat seperti Matplotlib dan Seaborn, yang memudahkan untuk memvisualisasikan data dan hasil dari model Machine Learning.

Kekurangan Python:

1. Performa yang Lebih Lambat: Python adalah bahasa interpretatif, yang artinya kode dijalankan baris per baris. Ini membuat performanya cenderung lebih lambat daripada bahasa kompilasi seperti C++ atau Java.
2. Global Interpreter Lock (GIL): GIL di Python membatasi eksekusi kode secara paralel pada level thread. Ini berarti bahwa meskipun Python mendukung pemrograman paralel dengan library tertentu, eksekusi multi-threading tidak bisa secara efektif memanfaatkan banyak core CPU.
3. Keterbatasan pada Perangkat Mobile: Jika Anda ingin membangun model Machine Learning yang akan dijalankan di perangkat mobile, Python mungkin bukan pilihan utama karena ukuran yang besar dan konsumsi daya yang tinggi.
4. Kurangnya Dukungan Just-in-Time (JIT) Compilation: Python tidak memiliki JIT Compilation (kompilasi saat runtime), yang dapat mempengaruhi kinerja dalam beberapa aplikasi Machine Learning yang membutuhkan pemrosesan intensif.

## 2.8 XAMPP

XAMPP adalah perangkat lunak berbasis *web server* yang bersifat *open source* (bebas), sebuah *software web server apache* yang didalamnya sudah tersedia database server MySQL dan dapat mendukung pemrograman PHP. XAMPP merupakan software yang mudah digunakan, gratis dan mendukung instalasi di *Linux* dan *Windows*. XAMPP digunakan sebagai *standalone server* (berdiri sendiri) atau biasa disebut dengan *localhost*. Keuntungan lainnya adalah cuma menginstal satu kali sudah tersedia *Apache web server, MySQL Database Server, PHP* dan *Perl* (Hanafri et al., 2018).

yang XAMPP adalah singkatan dari X (empat sistem operasi apa saja). XAMPP adalah perangkat lunak gratis yang mendukung berbagai sistem operasi dan merupakan kumpulan berbagai aplikasi. Apache, MySQL, PHP, dan Perl adalah beberapa contohnya. Xampp adalah sebuah program yang memberikan rangkaian perangkat lunak dalam satu paket. Apache (Web Server), MySQL (Database), PHP (Server Side Scripting), Perl, server FTP, PhpMyAdmin, dan beberapa perpustakaan tambahan lainnya disertakan dalam paket Xampp. Xampp adalah singkatan dari :

X : Alat ini kompatibel dengan berbagai sistem operasi, termasuk Windows, Linux, Mac OS, dan Solaris.

A : Apache adalah program server web. Tanggung jawab utama Apache adalah menghasilkan halaman web yang akurat bagi pengguna bergantung pada kode PHP yang disediakan oleh pembuat web atau pengguna.

M : MySQL, adalah program server data. SQL, yang merupakan singkatan dari Structured Query Language, adalah nama lain untuk pengembangan. SQL adalah bahasa terstruktur untuk pemrosesan database.

P : PHP, adalah bahasa pemrograman web yang memungkinkan pengguna mengembangkan skrip sisi server untuk situs web mereka.

P : Perl, yang merupakan bahasa pemrograman tujuan umum yang dibuat oleh Larry Wall pada komputer Unix (Novendri, 2019).



Gambar 2.4 Logo XAMPP (Novendri, 2019).

## 2.9 Website

*Website* adalah situs web, kumpulan data dokumen berupa halaman *website* atau area. Sebuah *website* terdiri dari beberapa halaman yang saling berhubungan dibawah nama domain, umumnya berisi konten teks, video, gambar, audio, dan lain-lain. Dengan cara ini pada umumnya *website* memiliki satu halaman utama yang disebut homepage. Homepage adalah halaman arahan yang akan terbuka setiap kali mengakses nama domain *website*. Anda harus memiliki perangkat (komputer, smartphone) yang terhubung dengan web atau internet untuk dapat membuka sebuah *website*. Situs web atau halaman sering kali ditulis dalam Hyper Text Markup Language (HTML), yang dapat diedit diperoleh melalui HTTP atau HTTPS, sebuah konvensi yang melewati data berbeda dari server *website* untuk ditampilkan kepada para user melalui *web browser* (Novendri, 2019).

*Website* adalah situs-situs yang saling berhubungan yang memiliki domain sebagai alamat *World Wide Web (www)* dan hosting sebagai media penyimpanan data dalam jumlah besar. *Website* dapat dikunjungi melalui jaringan internet dengan memanfaatkan platform yang disebut browser, seperti *Chrome.*, *Mozilla Firefox*, *Internet Explorer (IE)*, *Opera* dan sebagainya (Rina Noviana, 2022).

## 2.10 MySQL (My Structure Query Language)

MySQL adalah sebuah kerangka kerja administrasi basis informasi SQL yang bersifat *Open Source* dan paling dikenal saat ini. Kerangka dasar informasi MySQL menjunjung tinggi beberapa elemen seperti multithreaded, multiuser, dan kerangka kerja administrasi kumpulan data SQL (DBMS). Informasi ini dibuat untuk motivasi dibalik kerangka kumpulan data yang cepat, solid, dan mudah digunakan.







Gambar 2.6 Logo MySQL (Anhar, 2010).

### 2.11 Visual Studio Code

Visual Studio Code adalah manajer konten yang ringan dan kuat yang dibuat oleh Microsoft untuk kerangka kerja multiplatform, artinya tersedia bahwa itu juga dapat diakses untuk kerangka multiplatform, artinya tersedia bahwa itu juga dapat diakses untuk versi Linux, Mac, dan Windows. Pengelolah kata Ini secara langsung mendukung bahasa pemrograman seperti Javascript, TypeScript, dan Note Js, serta bahasa pemrograman lainnya melalui modul yang tersedia melalui pusat komersial Visual Studio Code, termasuk C++, C#, Python, Go, Java, dan lainnya. PHP, dan sebagainya (Gligorijevic et al., 2019).



Gambar 2.7 Logo Visual Studio Code

Gambar di atas ini adalah Logo Visual Studio Code, pengelola lontan VS Code juga bersifat *open source*, dimana anda mendapatkan kode sumber dan anda dapat menambahkan pergantian untuk pengembangannya. Hal ini juga membentuk VS Code sebagai favorit para perkembangan aplikasi karena para perkembangan pada perangkat lunak maupun ikut serta dalam proses pengembangan VS Code kedepannya.

### 2.12 Flowchart

*Flowchart* merupakan gambar, diagram, dan grafik yang dipakai buat memperlihatkan urutan atau interaksi suatu program. *Flowchart* sering dirancang khusus untuk membantu memecahkan masalah yang memerlukan penyelidikan atau evaluasi. Struktur organisasi dibuat untuk menunjukkan bagaimana data diproses untuk menjelaskan aspek fisik. Didefinisikan arau digunakan untuk mendokumentasikan item fisik sistem informasi, termasuk yang ada dan yang akan datang produksi yang dibuat.


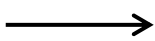




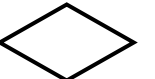


*Flowchart* adalah prosedur pemecahan masalah yang ditulis dalam simbol tertentu karena menggambarkan alur logis dari prosedur pemecahan masalah. *Flowchart* ini tidak hanya diperlukan sebagai alat komunikasi juga diperlukan sebagai sumber daya, dan sebelum melanjutkan untuk memahami apa yang dimaksud dengan diagram alur, kita harus mengomunikasikannya. Aturan dalam perancangan *Flowchart* yaitu:

1. *Flowchart* ditampilkan dari aras kebawah, kanan ke kiri.
2. Setiap aktivitas/proses dalam *flowchart* harus ditentukan secara eksplisit.
3. Setiap diagram alur harus dimulai dengan keadaan awal dan harus diakhiri dengan satu atau lebih keadaan terminal akhir. /terminator/status berhenti.
4. Gunakan konektor dan status konektor halaman PFF dengan label yang sama untuk menunjukkan hubungan antara bagian-bagian dari algoritma yang terputus-putus, misalnya sebagai akibat dari perpindahan/perubahan halaman.

Tujuan dari *flowchart* ini adalah untuk menjelaskan fase pemecahan masalah dengan cara yang sederhana dan rapi, gunakan simbol standard untuk ketajaman dan kejelasan (Sitorus, 2017). Banyak sekali notasi yang digunakan dalam penggunaan *flowchart*. Simbol-simbol dalam diagram alur mempunyai arti yang berbeda-beda. Simbol-simbol yang tercantum di bawah ini sering digunakan dalam proses pembuatannya. *Flowchart*, sebagai berikut:

Tabel 2.1 Simbol Flowchart (Sitorus, 2017).

Simbol	Nama	Keterangan
--------	------	------------

	Terminator	Permulaan atau akhir dari program
	Garis alir ( <i>Flow Line</i> )	Arah aliran program
	Preparation	Proses inisialisasi atau pemberian harga awal.
	Proses	Proses perhitungan atau proses pengolahan
	<i>Input / Output data</i>	Proses <i>input</i> atau <i>output</i> data, parameter maupun informasi
	<i>Predefined process</i> (sub program)	Peramalan sub program atau proses menjalankan sub program
	<i>Decision</i>	Perbandingan pernyataan, pemilihan data yang menghadirkan kemungkinan untuk fase berikutnya
	<i>One page connector</i>	Penghubung bagian-bagian <i>flowchart</i> yang berada pada satu halaman
	<i>Off page connector</i>	Penghubung bagian-bagian <i>flowchart</i> yang berada pada halaman berbeda

Simbol-simbol di atas mempunyai banyak jenis dan peranan, diantaranya berfungsi sebagai penghubung suatu simbol dengan simbol lainnya, seperti yang terlihat pada simbol Flow, kutipan dihalaman dan diluar halaman (*on-page* dan *off-page reference*). Apalagi ada tandanya digunakan untuk menampilkan proses yang sedang berjalan dan apa akhirnya ada notasi untuk memasukkan *input* dan menampilkan *output*.

### 2.13 Unified Modelling Language (UML)

*Unified Modelling Language (UML)* adalah salah satu alat yang paling berguna handal di bidang pengembangan sistem apa yang terjadi dengan orientasi objek standar industri visualisasi dalam desain dan dokumentasi sistem perangkat lunak. Tentang perkembangan teknologi pemrograman berorientasi objek, bahasa pemodelan standar untuk pengembangan perangkat lunak yang dibuat menggunakan prinsip pemrograman berorientasi objek, yaitu *Unified Modelling Language (UML)*. Biasanya hanya lima gambar dalam sebuah UML, yaitu *Use Case Diagram*, *Class Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Deployment Diagram* (Syarif et al., 2021).

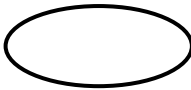
*Unified Modelling Language* merupakan salah satu dari metode pemodelan visual untuk desain dan produksi perangkat lunak berorientasi pada objek UML adalah standar penulisan atau semacam *blue print* yang berisi proses bisnis, ditulis kursus dalam bahasa tertentu (Prihandoyo, 2018).

### 2.13.1 Use Case Diagram

*Use Case Diagram* adalah aktivitas atau interaksi yang berkelanjutan antara aktor dan sistem. Alternatifnya, strategi biasanya digunakan untuk merancang perangkat lunak sistem informasi untuk memperoleh kebutuhan fungsional sistem saat ini. Studi Kasus Fungsionalitas yang disediakan oleh sistem ketika unit bertukar pesan antar unit atau aktor digambarkan dalam diagram. (Syarif et al., 2021).

*Use Case Diagram* Terdapat use case yang merupakan gambaran deskripsi fungsional yang diinginkan dari suatu sistem dan menggambarkan interaksi antara aktor dan sistem. Terdapat aktor yang merupakan gambaran dari komponen sistem yang beroperasi pada sistem. (Prihandoyo, 2018).

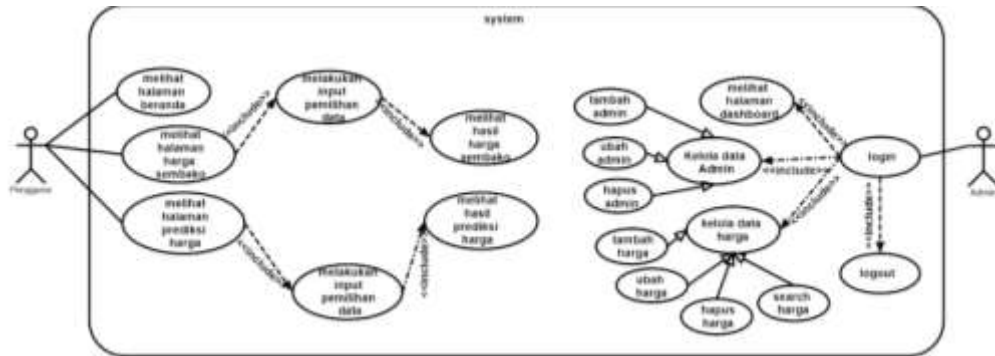
Tabel 2.2 Simbol *Use Case* (Prihandoyo, 2018).

Simbol	Nama	Keterangan
	<i>Use Case</i>	Fungsionalitas yang disediakan oleh sistem ketika unit bertukar pesan dengan unit atau aktor lain; biasanya

		ditentukan dengan menyertakan kata kerja di awal frasa nama kasus penggunaan.
	Aktor ( <i>actor</i> )	Orang atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibangun, meskipun simbol bagi aktor adalah gambaran seseorang, namun aktor tersebut tidak selalu berupa orang; umumnya ditunjukkan dengan menggunakan kata benda di awal frasa nama aktor
	Asosiasi ( <i>Association</i> )	Komunikasi antar aktor dan use case yang mengambil bagian dalam use case atau berinteraksi dengan aktor
	Ekstensi ( <i>extend</i> )	Kaitan antara use case tambahan dan use case dimana use case baru dapat berdiri sendiri meskipun tidak ada use case lebih lanjut.
	Generalisasi ( <i>generalization</i> )	Hubungan antara generalisasi dan spesialisasi (generik-spesifik) antara dua use case dimana fungsi yang satu lebih umum dibandingkan fungsi lainnya.
	<i>Include / uses</i>	Use case tambahan mempunyai link ke sebuah use case dimana use case yang ditambahkan tersebut memerlukan use case lain untuk menjalankan fungsinya atau sebagai

		prasyarat untuk menjalankan use case ini.
--	--	---

Contoh *Use Case*:



Gambar 2.8 Contoh *Use Case Diagram* (Ludiastami et al., 2018).

Diagram diatas merupakan sebuah contoh *Use Case Diagram* sederhana dapat melihat halaman beranda, melihat halaman harga sembako dan melihat halaman memprediksi Harga Bahan Pokok yang terdiri dari dua factor yaitu pengguna dan admin.





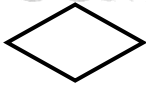
### 2.13.2 Activity Diagram

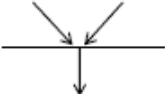

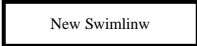
*Sa Activity Diagram* merupakan sistem Proses atau tindakan perangkat lunak. Activity Diagram juga digunakan untuk menetapkan urutan atau pengelompokan tampilan sistem. Setiap aktivitas memiliki antarmuka tampilan dan gaya menu tersendiri yang disajikan pada program.

*Activity Diagram* mendefinisikan alur kerja atau tindakan sistem, proses bisnis, atau menu perangkat lunak. Perlu ditekankan bahwa diagram aktivitas mewakili aktivitas sistem, bukan apa yang dapat dicapai oleh pelaku aktivitas, namun sistem tidak dapat mencapainya. (Oktavia & Sucipto, 2021).

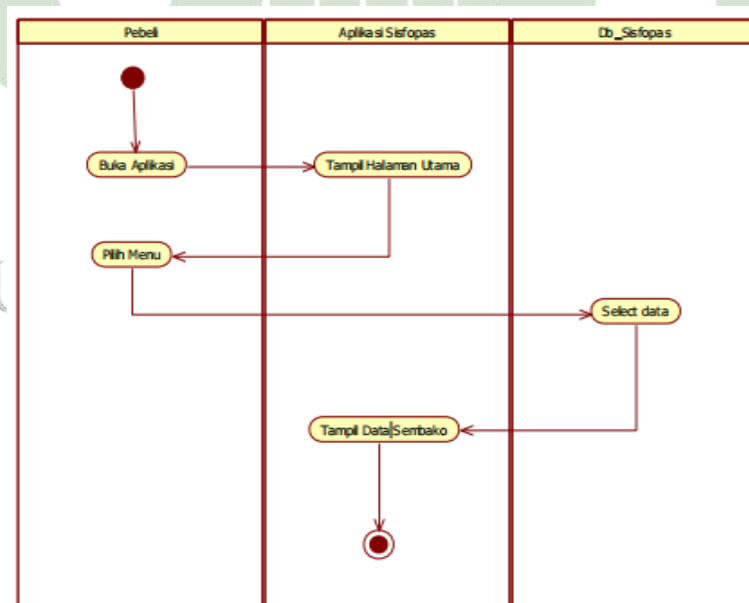
Berikut ini merupakan simbol-simbol yang digunakan dalam *Activity Diagram*, adalah:

Tabel 2.3 Simbol *Activity Diagram* (Oktavia & Sucipto, 2021).

Simbol	Nama	Keterangan
	Status Awal ( <i>Start Point</i> )	Diagram aktivitas memiliki keadaan awal aktivitas sistem.
	Aktivitas ( <i>Activitis</i> )	Tugas yang dilakukan sistem, yang umumnya dimulai dengan kata kerja
	Percabangan ( <i>Decision</i> )	Percabangan terjadi ketika lebih dari satu aktivitas tersedia.

	Penggabungan ( <i>Join</i> )	Digunakan untuk menunjukkan adanya dekomposisi
	Status akhir ( <i>End Point</i> )	Status akhir yang dilakukan sistem
	<i>Swimlane</i>	Memisahkan organisasi yang bertanggung jawab terhadap aktivitas yang terjadi

Contoh Activity Diagram:



Gambar 2.9 Contoh Use Case Diagram (Ludiastami et al., 2018).

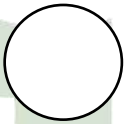
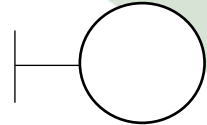
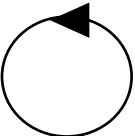
Gambar diatas merupakan sebuah contoh *Activity Diagram* sederhana yang dirancang untuk membuat tampilan daftar harga bahan pokok aktifitas ini dilakukan oleh pembeli untuk melihat daftar harga pada sistem harga kebutuhan pokok.


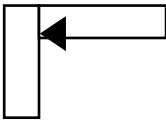


### 2.13.3 Sequence Diagram

*Sequence Diagram* adalah diagram yang dibuat untuk menunjukkan alur interaksi antar benda. *Sequence Diagram* harus mempunyai informasi yang sama dengan dan *Class Diagram*. Satu *Sequence Diagram* akan menampilkan satu *Use Case*.

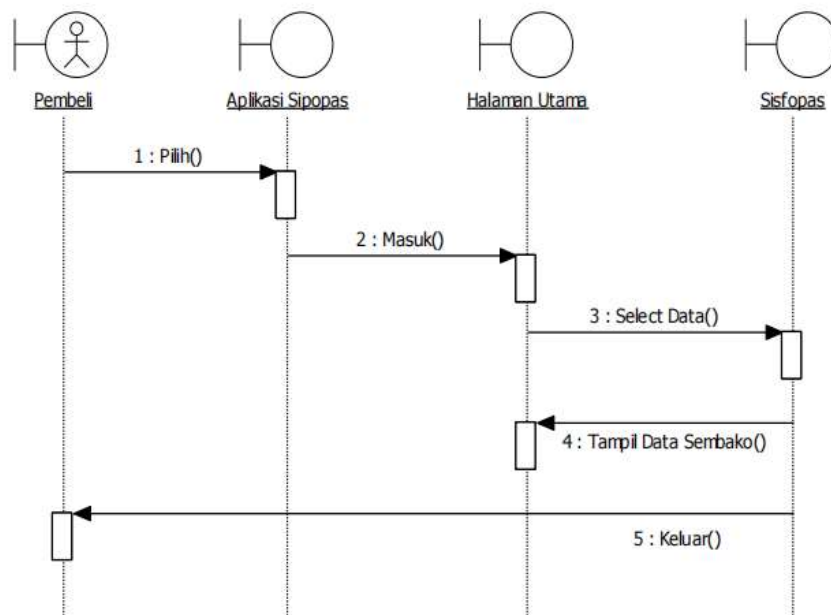
*Sequence Diagram* adalah dibuat untuk menentukan jalannya interaksi antar objek konten urutan diagram harus identik dengan *Use Case* dan *Class Diagram*.

Tabel 2.4 Simbol *Sequence Diagram* (Hananto & Priyatna, 2017).

Simbol	Nama	Keterangan
	<i>Entity Class</i>	Merupakan komponen sistem yang terdiri dari kumpulan kelas-kelas berupa entitas yang membentuk gambaran awal sistem dan menjadi landasan dalam membangun database.
	<i>Boundary Class</i>	Berisi sekumpulan kelas yang berfungsi sebagai antarmuka atau titik kontak antara satu atau lebih aktor dan sistem, seperti tampilan formulir masuk dan formulir cetak.
	<i>Control Class</i>	Perhitungan dan aturan bisnis yang melibatkan beberapa objek adalah contoh objek yang menyertakan logika aplikasi tetapi tidak

		memiliki tanggung jawab terhadap suatu entitas.
	<i>Message</i>	Simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i>	Menjelaskan transmisi pesan ke diri sendiri.
	<i>Activation</i>	<i>Activation</i> menggambarkan pelaksanaan operasi suatu objek; panjang kotak ini berhubungan dengan waktu aktivitas suatu operasi.
	<i>Lifeline</i>	Garis putus-putus yang menghubungkan objek ke garis hidup telah diaktifkan.

### Contoh Sequence Diagram



Gambar 2.10 Contoh *Use Case Diagram* (Ludiastami et al., 2018).

Gambar diatas merupakan sebuah contoh *Sequence Diagram* sederhana yang dirancang untuk mrrmbuat sebuah sistem daftar harga bahan pokok.

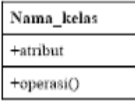

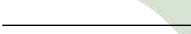

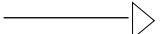
#### 2.13.4 Class Diagram

*Class Diagram* adalah Tabel yang akan dihasilkan dalam suatu sistem. Diagram ini digunakan untuk mendefinisikan kelas-kelas yang akan dihasilkan untuk membangun sistem dan menjelaskan struktur sistem.

*Class Diagram* biasanya Kelas, Relasi, Asosiasi, Generalisasi dan Agregasi, properti, Operasi (Metode), Visibilitas, jumlah akses objek eksternal ke operasi atau properti koneksi kelas yang menyertakan informasi multiplisitas atau kardinalitas semuanya disertakan. (Oktavia & Sucipto, 2021).

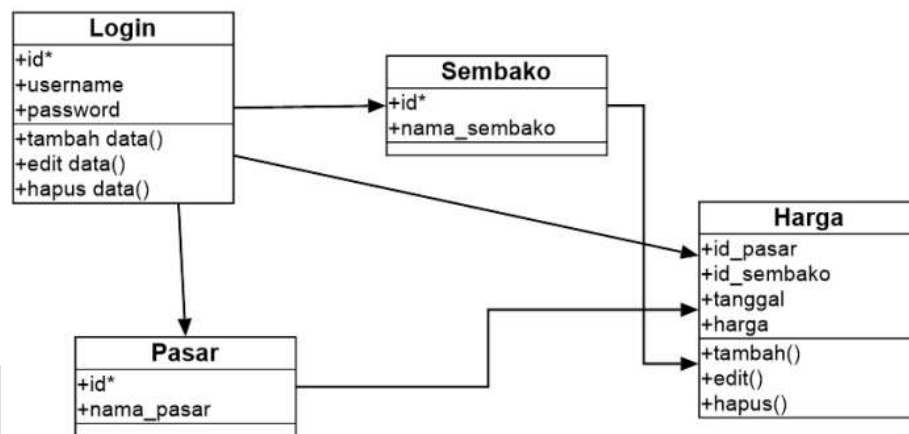
Berikut merupakan Atribut yang digunakan dalam *Class Diagram*, yaitu:

Tabel 2.5 Atribut *Class Diagram* (Oktavia & Sucipto, 2021).

Simbol	Nama	Keterangan
	Class	Kelas pada struktur sistem
	Antarmuka / <i>Interface</i>	Gagasan antarmuka dalam pemrograman berorientasi objek serupa.
	Asosiasi / <i>Association</i>	<i>Asosiasi</i> dan relasi antar kelas pada umumnya disertai dengan keberagaman.
	Asosiasi berarah / <i>Directed Association</i>	Asosiasi adalah hubungan antar kelas dimana arti dari satu kelas digunakan oleh kelas lainnya. Asosiasi umumnya disertai dengan keberagaman.
	<i>Generalisasi</i>	Relasi antar kelas dengan makna <i>generalisasi-spesialisasi</i> (umum khusus).

----->	Kebergantungan/ <i>Dependency</i>	Hubungan kelas dengan pengertian saling ketergantungan antar kelas
—◇	Agregasi/ <i>Aggregation</i>	Relasi antar kelas dengan makna semua bagian.

Contoh Class Diagram:



Gambar 2.11 Contoh *Use Case Diagram* (Ludiastami et al., 2018).

Gambar diatas merupakan contoh dari *Class Diagram* yang dirancang untuk menunjukkan hubungan antar tabel yang digambarkan dalam bentuk *Class Diagram*. Yaitu tabel login, sembako, harga dan pasar.

## 2.14 Penelitian Terdahulu

Penelitian terdahulu merupakan salah satu sumber tambahan perbandingan dan motivasi untuk penelitian selanjutnya. Penelitian pada bagian ini mencakup berbagi hasil penelitian sebelumnya yang terkait dengan subjek ini. Dengan menyelesaikan tahap ini, Anda akan dapat menentukan ruang lingkup studi yang ingin Anda lakukan. Terdapat beberapa penilitian ini yang relavan dengan penelitian yang sudah dilakukan sebelumnya, diantaranya terdapat didalam tabel berikut:

Tabel 2.6 Penelitian Terdahulu

NO	Peneliti	Asal dan tahun	Judul	Kesimpulan
1.	Sara Septian Aruna, Yohanes Anton Nugroho	2021	Perbandingan Metode Arima Dan Sarima Dalam Peramalan Penjualan Kelapa (Aruan & Nugroho, 2021)	Pendekatan ARIMA dengan menggunakan model merupakan cara terbaik untuk memproyeksikan penjualan kelapa di UKM Pak Balen Pasar Kandak Medan. Validasi peramalan ARIMA dan penjualan aktual menghasilkan kesimpulan bahwa pendekatan ARIMA tepat untuk meramalkan penjualan kelapa di UKM Pak Balen Pasar Kandak Medan. Sebab, temuan peramalan ARIMA mendekati data penjualan kelapa sebenarnya.
2.	Wahyu Ngestisari, Bambang Susanto, Tundjung Mahatma	2020	Perbandingan Metode ARIMA dan Jaringan Syaraf Tiruan Untuk Peramalan	Namun kondisi iklim, pola produksi beras, kondisi pasar, kenaikan biaya transportasi, dan panjang jalur distribusi beras dari produsen ke



			Harga Beras (Ngestisari et al., 2020)	konsumen semuanya mempengaruhi kestabilan ketersediaan beras dan harga beras sehingga menyebabkan harga beras naik dan turun setiap bulannya.
3.	Mohammad Arif Rasyidi	2017	Prediksi Harga Bahan Pokok Nasional Jangka Pendek Menggunakan ARIMA (Rasyidi, 2017)	Karena barang-barang kebutuhan pokok sangat penting bagi kehidupan, permintaan terhadap barang-barang tersebut cenderung stabil (tidak elastis) terhadap fluktuasi harga. Ketika harga diperkirakan akan naik drastis, pemerintah dapat memastikan bahwa barang-barang kebutuhan pokok tetap murah bagi konsumen melalui mekanisme operasi pasar dengan memiliki proyeksi harga yang tepat.
4.	Kandari Puteri, Astried Silvanie	2020	<i>Macheni Learning</i> Untuk Model Prediksi Harga Sembako Dengan Metode	Karena faktor eksternal yang terus berubah, serta kebutuhan akan informasi harga pangan pokok sehari-hari, biaya

			<p>Regresi Linier Berganda (Puteri &amp; Silvanie, 2020)</p>	<p>pangan dapat naik dan turun (berfluktuasi) setiap saat. Selain itu, penelitian ini dilakukan dengan menggali kumpulan data yang terdiri dari 90.945 baris data, yang pada akhirnya akan menghasilkan informasi baru yang penting yaitu proyeksi harga sembako harian.</p>
5.	<p>Dea Ludiastami, Ana Kurniawati, Dina Anggriani</p>	2018	<p>Aplikasi Prediksi Harga Bahan Pokok Menggunakan Algoritma Monte Carlo Berbasis Web (Ludiastami et al., 2018)</p>	<p>Untuk di butuhkan aplikasi dapat memberikan proyeksi harga bahan pokok yang dapat digunakan masyarakat untuk menentukan di mana akan membeli bahan pokok tersebut.</p> <p>Penelitian ini tentang prediksi telah banyak dilakukan oleh penelitian tentang prediksi prestasi siswa SMA menggunakan JST Backpropagation. Tujuan penelitian ini adalah memebuat aplikasi yang dapat memprediksi harga</p>

				<p>bahan pokok untuk lima barang di Jakarta pusat dengan algoritma Monte Carlo, yang di harapkan dapat membantu pengguna aplikasi untuk memperoleh harga termurah dari pasar-pasar yang ada.</p>
--	--	--	--	--



UNIVERSITAS ISLAM NEGERI  
SUMATERA UTARA MEDAN