

BAB II

LANDASAN TEORI

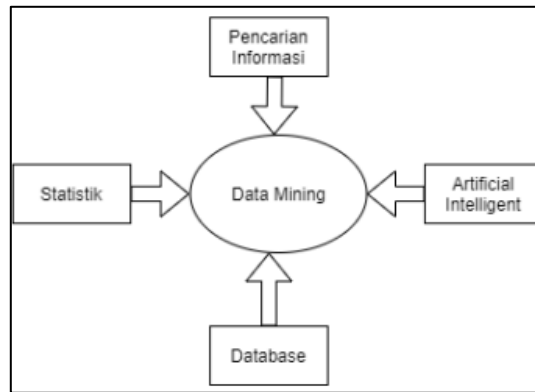
2.1 Data Mining

Menurut Efori Buulolo (2020) dalam *e-book “Data mining untuk Perguruan Tinggi”*. Data mining, juga dikenal sebagai Knowledge Discovery in Database (KDD), adalah proses pengumpulan dan penggunaan data historis untuk menemukan pengetahuan, informasi, keteraturan, pola, atau hubungan dalam sejumlah besar data. Data mining menghasilkan hasil yang dapat digunakan untuk membantu pengambilan keputusan atau memperbaiki keputusan di masa yang akan datang. (Adam, K. 2022).

Beberapa faktor menyebabkan bidang ilmu data mining berkembang pesat, seperti:

1. Kesadaran akan pentingnya data meningkat.
2. Penggunaan yang lebih besar dari hasil pengolahan data dalam berbagai industri, seperti industri bisnis.
3. Kumpulan data yang berkembang dengan begitu cepat.
4. Meningkatkan akses internet melalui navigasi web dan smartphone.
5. Pengembangan software dan hardware khusus untuk data mining.
6. Bidang komputasi komputer mengalami kemajuan yang begitu cepat.
7. Media penyimpanan yang lebih besar dan lebih terjangkau.

Pada gambar 2.1 menjelaskan tentang *data mining*. Data mining bukanlah suatu bidang keilmuan yang berdiri sendiri, namun berkaitan erat dengan bidang keilmuan lain seperti database, statistika, pencarian informasi, dan kecerdasan buatan.

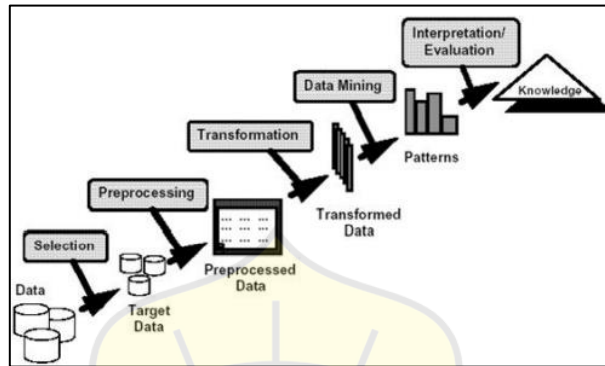


Gambar 2. 1 Bidang Ilmu Data Mining (Sapitri A., 2021)

Berikut adalah Penjelasan dari gambar diatas:

1. *Database – Data mining* : Data mining menggunakan kumpulan data dari database. Data pencarian informasi terpisah dari data operasional dalam database.
2. *Statistic – Data mining* : Statistik adalah bagian dari proses pengambilan keputusan, yang melibatkan pengumpulan data, pengambilan sampel data, dan penggunaan teknik statistik untuk menentukan sampel data, menganalisisnya, dan menampilkan hasilnya.
3. *Pencarian Informasi – Data mining* : Bagian dari proses penambangan data (termasuk interpretasi, analisis, dan penyimpanan data) adalah pengambilan informasi.
4. *Artificial Intelligence : Data mining* : Machine learning ialah cabang penting dari ilmu data mining di mana sistem komputer belajar dari data pelatihan yang mereka gunakan.

Pada gambar 2.2, ada banyak jenis model proses yang digunakan dalam data mining untuk memandu implementasi data mining. Model proses yang umum dipakai antara lain Knowledge Discovery Database (KDD), CRISP-DM dan SEMMA.



Gambar 2. 2 Bidang Ilmu Data Mining (Sapitri A., 2021)

Berikut penjelasan dari setiap tahapan pada KDD:

a. *Domain Understanding and KDD Goals*

Tujuan ditentukan dari sudut pandang pengguna dan digunakan untuk mengembangkan dan memahami domain aplikasi dan pengetahuan sebelumnya.

b. *Selection and Additions*

Fase kedua berfokus pada mengidentifikasi data target dan data sampel atau subkumpulan variabel.

c. *Preprocessing: Data Cleaning etc*

Pembersihan dan pemrosesan awal data adalah operasi dasar untuk menghasilkan data yang konsisten dan bebas gangguan.

d. *Transformation*

Mengonversi data dari satu bentuk ke bentuk lainnya sehingga data tersebut mudah diimplementasikan.

e. *Data Mining (Choosing the Suitable Data Mining Task)*

Metode data mining yang sesuai dipilih berdasarkan tujuan tertentu yang telah ditentukan pada tahap pertama. Contoh metode data mining adalah klasifikasi, regresi, clustering, dan agregasi.

f. *Data Mining (Choosing the Suitable Data Mining Algorithm)*

Memilih algoritma yang tepat untuk mencari pola data pemilihan algoritma didasarkan pada kriteria dan kesesuaian metode data mining.

g. *Data Mining (Implying Data Mining Algorithm)*

Pada tahap ini algoritma yang dipilih diimplementasikan.

h. *Evaluation and Interpretation*

Fase ini berfokus pada interpretasi dan evaluasi, termasuk memeriksa apakah pola atau informasi yang ditemukan bertentangan dengan asumsi yang sudah ada.

i. *Discovered Knowledge*

Gunakan pengetahuan yang ditemukan dari proses KDD untuk memutuskan apa yang harus dilakukan dengan pengetahuan yang dihasilkan.

2.2 Teknik Data Mining

Berdasarkan Vulandari (2017), dalam (Sapitri, Anggraini 2021), "Penemuan" dan "pembelajaran" sangat terkait dengan teknik data mining dan dibagi menjadi 3 teknik pembelajaran pertama, yaitu:

1. Supervised Learning

Supervised Learning ialah teknik yang paling umum digunakan. Tekniknya sama dengan Pemrograman dengan Contoh. Metode ini melibatkan fase pelatihan di mana data pelatihan sebelumnya dengan karakter yang dipetakan ke hasil yang diketahui diproses dalam algoritma data mining. Selama proses ini, algoritma dilatih untuk mengidentifikasi variabel dan nilai kunci. Nilai-nilai ini nantinya akan digunakan sebagai dasar perkiraan ketika data baru diberikan.

2. *Unsupervised Learning*

Teknik pembelajaran ini tidak melibatkan tahapan pelatihan seperti 6 tahapan dalam supervisi learning. Teknik ini menggunakan algoritma untuk menemukan semua pola, seperti hubungan, yang berasal dari kriteria penting dalam data masukan. Metode ini menghasilkan pembentukan berbagai aturan yang menggambarkan penemuan, seperti asosiasi, kelompok, dan segmentasi. Kemudian, hal-hal penting ditemukan melalui analisis aturan ini.

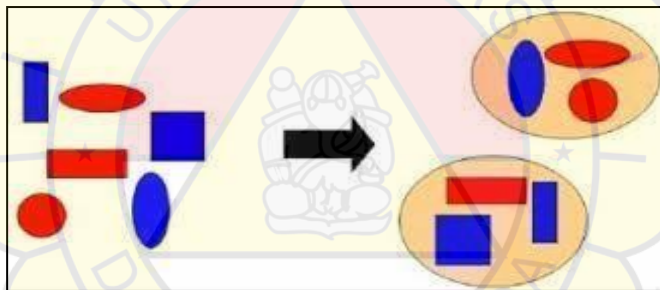
3. *Reinforcement Learning*

Dibandingkan dengan dua metode lainnya, metode pembelajaran ini jarang digunakan. Namun, ia memiliki kontrol adaptif dan penerapan yang terus dioptimalkan dari masa ke masa. Untuk menyelesaikan masalah yang sulit dan bergantung pada waktu, reinforcement learning sangat berguna.

2.3 Teknik *Clustering*

Menurut Kusuma and Andriani dalam (Sapitri, Anggraini, 2021). Teknik *Clustering* atau pengklusteran merupakan teknik yang membagi data menjadi kelompok-kelompok. Pembagian kelompok data tersebut berdasarkan kriteria tertentu.

Pada gambar 2.3, Teknik clustering digunakan untuk mengelompokkan data berdasarkan kriteria yang sama dalam satu area atau kelompok yang sama dan memisahkan data berdasarkan kriteria yang berbeda. Data akan dikumpulkan dalam satu kelompok atau cluster dengan tingkat kemiripan yang paling tinggi dan setiap kelompok akan memiliki tingkat kemiripan yang minimum.



Gambar 2. 3 Penggambaran Teknik Clustering (Sapitri A., 2021)

Berikut merupakan beberapa contoh implementasi teknik *clustering* dalam dunia nyata:

1. *Data summarization* atau peringkasan data. *Data mining* berkaitan dengan meringkas informasi dari data sehingga proses *clustering* sering kali merupakan langkah awal dari banyak algoritma *data mining*.
2. *Customer segmentation* atau segmentasi konsumen yang sering kali digunakan untuk menganalisis perilaku umum dari kelompok konsumen yang serupa. Sebagai contoh pola perilaku pemilihan produk suatu konsumen yang ada pada

suatu kelompok dapat digunakan untuk merekomendasikan produk tersebut kepada konsumen lain yang masih dalam satu kelompok.

3. *Social network analysis* atau analisis jaringan sosial yang digunakan untuk menganalisis perilaku manusia yang diperoleh dari menganalisis dinamika yang terjadi pada komunitas. Jaringan sosial merupakan simpul yang telah user kluster secara erat melalui hubungan yang saling terkait yang biasanyakelompok teman yang sama atau komunitas.
4. Berhubungan dengan masalah *data mining* yang lain.

Teknik clustering antara lain sering menggunakan metode berikut:

1. *Partitioning method* merupakan teknik yang digunakan untuk mengelompokkan data sesuai dengan jumlah cluster atau kelompok yang ditentukan. Dalam hal ini jumlah kelompok ditentukan di awal. Algoritma yang populer digunakan pada metode partitioning adalah K-means, K-medoids dan CLARA (*Clustering Large Application*).
2. *Hierarchical method* atau metode hierarki merupakan alternatif dari *partitioning method* dan tidak memerlukan menentukan jumlah *cluster* atau kelompok di awal. Terdapat dua pendekatan yaitu *Divisive* dan *Agglomerative*. Pendekatan *divisive* yang merupakan pendekatan *top-down*. Semua objek data dikumpulkan dalam satu *cluster* dan selanjutnya dipecah menjadi *cluster* yang lebih kecil menggunakan iterasi berulang hingga kondisi selesai tercapai. Sedangkan pendekatan *agglomerative* merupakan pendekatan *bottom-up* di

mana setiap kelompok data terpisah sejak awal dan digabungkan hingga kondisi selesai tercapai.

2.4 Teknik Klasifikasi

Menurut (Dito Putro Utomo & Mesran, 2020) dalam jurnal Media Informatika Budidarma dengan judul “Analisis Komparasi Metode Klasifikasi Data Mining dan Reduksi Atribut Pada Data Set Penyakit Jantung”. Klasifikasi merupakan proses penilaian objek data agar dapat ditempatkan di kelas tertentu dari jumlah kelas yang ada. Klasifikasi melibatkan pembuatan model berdasarkan data pelatihan sebelumnya dan menggunakannya untuk mengklasifikasikan data baru. Klasifikasi adalah proses mengajarkan dan mendidik fungsi tujuan dengan memetakan setiap kumpulan fitur (atribut) ke dalam satu nomor label kelas yang tersedia. Meskipun klasifikasi diharapkan dapat mengklasifikasikan seluruh kumpulan data dengan benar, namun tidak dapat dipungkiri bahwa kinerja pada sistem tidak bisa 100% benar, sehingga sistem klasifikasi harus memperkirakan kinerjanya.

2.5 Pemrograman Aplikasi

2.5.1 Web

Menurut Agus Hariyanto dalam (Paramata, Hermawan, dkk, 2022), Web merupakan kumpulan halaman yang berisi data teks, gambar, animasi, suara, video, dan kombinasi dari semua jenis data tersebut. Halaman-halaman ini terhubung satu sama lain melalui jaringan-jaringan halaman, yang merupakan rangkaian bangunan yang saling terkait.

Dari Rohi Abdullah dalam (Ahmad Josi, 2017), Web adalah kumpulan

halaman yang masing-masing berisi informasi dalam bentuk data digital, seperti teks, gambar, video, audio, dan animasi lainnya, yang dikirimkan melalui Internet.

Menurut Destiningrum dan Adrian dalam (Nugraha, dkk, 2021), Web adalah sebuah program yang menampilkan dokumen-dokumen di internet dan memberi pengguna akses ke internet melalui web.

2.5.2 Hypertext Markup Language (HTML)

Menurut (Sepriano dan Melky,2022), Hypertext Markup Language (HTML) ialah bahasa markup yang digunakan untuk membuat halaman web yang menampilkan berbagai informasi di browser Web Internet dan memformat hypertext sederhana menjadi file ASCII untuk menghasilkan tampilan yang terintegrasi. Dengan kata lain, halaman web dibuat dari file yang dibuat oleh software pengolah kata menggunakan perintah HTML dan disimpan dalam bentuk standar ASCII.

2.5.3 Cascading Style Sheet (CSS)

Menurut (Sepriano dan Melky,2022), CSS ialah dokumen situs web yang dirancang guna menstandarkan tampilan dan nuansa situs web. Karena CSS bukanlah bahasa pemrograman, maka CSS merupakan aturan yang dibuat untuk mengontrol beberapa elemen pada suatu halaman web agar lebih mudah diatur, ditertibkan, dan digunakan pada saat mendesain suatu halaman web.

CSS bertanggung jawab untuk mengatur gaya font, warna, jarak, bentuk, dan elemen lainnya yang tidak mungkin dilakukan oleh komponen HTML. Bisa juga disebut bahwa CSS adalah pertumbuhan kode HTML menjadi dokumen web.

Karena CSS memisahkan presentasi dari konten web yang dibuat, membuatnya lebih mudah untuk mengatur dan memilih tampilan website.

2.5.4 JavaScript

Menurut (Siahaan & Sianipar, 2018) dalam e-book nya yang berjudul” JavaScript: Dari A Sampai Z” Javascript ialah bahasa skript terkenal yang di gunakan buat menghasilkan halaman *website* yang bisa berhubungan dengan pengguna serta bisa menanggapi peristiwa yang terjadi di halaman. JavaScript adalah perekat yang menyatukan halaman-halaman situs web. Javascript ialah bahasa Skript yang mengoptimalkan halamanhalaman HTML. Seluruh platform hampir bisa menjalankan javascript.

Javascript ialah bahasa sisi- klien yang di desain untuk browser pc kamu, 15 tidak untuk server. Javascript dibuat langsung di browser. Mozilla Firefox, Microsoft Internet Explorer, Goggle Chrome, Opera, hampir semua browser.

Karena JavaScript terhubung ke browser dan terintegrasi erat dengan HTML, saat browser memuat halaman, server mengirimkan seluruh konten dokumen, termasuk kode HTML dan pernyataan JavaScript. Konten HTML kemudian dibaca baris demi baris hingga tag awal javascript terbaca, setelah itu penerjemah javascript mengambil alih. Pemrosesan HTML terhubung ketika tag penutup javascript diterima.

2.5.5 PHP

PHP dibuat oleh Rasmus Lerdorf pertama kali pada tahun 1995 dan digunakan sebagai bahasa pemrograman untuk Personal Home Pages di server-side HTML-embedded. Sebelumnya, PHP dikenal sebagai FI (Form Interpreted), itu merupakan kumpulan script yang dipakai sebagai pengolah data form dari web.

Rasmus kemudian mempublikasikan kode sumbernya dan menamakannya PHP/FI, kependekan dari Hypertext Preprocessor/Form Interpreter. Dengan dirilisnya kode sumber ini secara open source, banyak programmer yang tertarik untuk mengembangkan PHP.

Menurut Risawandi (2019: 2) PHP punya beberapa keunggulan dari beberapa bahasa pemrograman lainnya, antara lain:

1. PHP merupakan bahasa pemrograman scripting yang tidak membutuhkan kompilasi ketika digunakan.
2. Web server yang mendukung PHP bisa ditemukan pada banyak tempat, mulai dari Apache, IIS, LightTPD hingga Xitami, dan konfigurasinya cukup sederhana.
3. Pengembangan lebih sederhana karena banyak pengembang dan milis yang sedia membantu.
4. Dari segi pengertian, PHP merupakan bahasa scripting yang paling sederhana karena memiliki banyak bahan referensi.

5. PHP merupakan bahasa open source yang dapat digunakan di berbagai mesin seperti Linux, Unix, Macintosh, dan Windows. Itu juga dapat menjalankan runtime melalui konsol dan juga mempunyai kemampuan akan melaksanakan perintah sistem.

2.5.6 Database

Menurut Rahimi (2020: 1) Kumpulan data terorganisir yang biasanya disimpan dan dapat diakses secara elektronik melalui sistem komputer disebut database. Basis data dikembangkan dengan menggunakan teknik perancangan dan pemodelan formal untuk memungkinkan peningkatan kompleksitas data.

DBMS atau Sistem manajemen basis data, juga dikenal sebagai " database management system " (DBMS), adalah perangkat lunak yang digunakan untuk mengelola basis data. DBMS bertindak sebagai antarmuka antara pengguna basis data (pengguna dan aplikasi DBMS langsung) dan data yang disimpan. Selain itu, DBMS mampu mengatur dan mengelola pengumpulan data yang lebih besar.

DBMS yang mendukung hubungan antar tabel adalah RDBMS. Contohnya adalah Oracle, MS SQL Server, MySQL, DB2, dan Microsoft Access.

2.5.7 MySQL

Menurut Rahimi (2020: 2) MySQL merupakan DBMS open source dengan 2 bentuk lisensi yaitu Free Software dan Shareware (piranti lunak berlisensi dengan pemakaian terbatas). Oleh sebab itu, MySQL merupakan server basis data gratis yang bersertifikat oleh GNU General Public License (GPL), yang memungkinkan

Anda menggunakannya guna kebutuhan pribadi atau komersial tanpa bayar sertifikat yang tersedia.

Seperti disebutkan sebelumnya, MySQL adalah RDBMS, atau Sistem Manajemen Basis Data Relasional. Oleh karena itu, istilah baris, kolom, dan tabel digunakan di MySQL. Misalnya, database MySQL mungkin memiliki satu atau lebih tabel.

MySQL ialah software sistem manajemen database SQL atau DBMS multi-threaded, multi-user yang membantu bahasa basis data SQL sebagai bahasa interaktif guna kelola data. MySQL juga merupakan mesin database ataupun basis data server yang membantu database SQL menjadi bahasa interaktif untuk kelola data.

2.5.8 Visual Studio Code

Menurut (Yudi dan Puji, 2019), *Visual Studio Code (VS Code)* ini merupakan editor teks ringan dan kuat yang dibuat oleh Microsoft untuk sistem operasi multi-platform, yang berarti juga tersedia dalam versi Linux, Mac, dan Windows. Text editor tersebut langsung mendukung bahasa pemrograman JavaScript, TypeScript, dan Node.js, serta bahasa pemrograman lainnya dengan bantuan plug-in yang dapat diinstal melalui Visual Studio Code Marketplace. Contoh bahasa pemrograman ini termasuk C++, C#, Python, Go, Java, dan lainnya.

Banyak fitur VSCode, masuk juga IntelliSense, integrasi Git, debugging, serta perluasan yang meningkatkan fungsionalitas editor text, akan selalu diperbarui

sejalan kemajuan rilis. VSCode diperbarui terus berkala tiap bulan, dan inilah yang melainkan VSCode dari editor text yang lain.

Selain itu, editor teks VSCode bersifat publik, sehingga bisa lihat kode sumber dan ikut andil dalam pembangunannya. Kode sumber bisa diakses lewat tautan Github. VSCode sangat disukai oleh pengembang aplikasi karena mereka dapat terlibat dalam proses pengembangan di masa mendatang.

2.5.9 XAMPP

Menurut (Aprilian & K Saputra, 2020) XAMPP merupakan piranti lunak sumber terbuka yang dibesarkan oleh Apache. XAMPP adalah paket piranti lunak yang menawarkan penyaluran Apache untuk MariaDB, server Apache, Hypertext Preprocessor (PHP), dan bahasa pengolahan dan laporan praktis (Perl). Selain itu, itu berfungsi sebagai tuan rumah dan server lokal. Server lokal ini berperan untuk pc desktop ataupun laptop kamu sendiri. Pemakaian XAMPP merupakan buat memverifikasi klien ataupun website kamu saat sebelum memuat ke server website jarak jauh. piranti lunak server XAMPP ini berikan kamu area yang sama untuk mencoba proyek MySQL, *Hypertext Preprocessor* (PHP), Apache serta *Practical Extraction and Reporting Language* (Perl) di pc lokal. XAMPP adalah singkatan dari cross-platform, (A) Server Apache, (M)eter, MariaDB, (P)HP dan (P)Perl. Lintas platform umumnya berfungsi di komputer mana pun dengan sistem operasi apa pun.

2.5.10 Hosting

Menurut Nugi Abdiansyah (2018: 32) Hosting adalah penyewaan tempat untuk menerima data-data yang diperlukan suatu website agar data tersebut dapat diakses melalui internet. Jenis data yang bisa diakses antara lain file, gambar, email, aplikasi, program, dan database.

Selain itu, layanan hosting situs web dan aplikasi web menawarkan dukungan tambahan seperti keamanan, pencadangan, dan performa situs web, sehingga pelanggan dapat menghabiskan lebih banyak waktu untuk melakukan tugas utama situs web.

2.6 Algoritma Sistem

2.6.1 Algoritma ARIMA

Menurut (Ester, M. T., 2021), *Moving Average* sering juga sering disebut model rata-rata bergerak terintegrasi autoregresif. Model moving average (MA) pertama kali digunakan oleh Slutsky pada tahun 1973. Namun, baru pada tahun 1938 Wold mengusulkan landasan teori untuk proses kombinasi ARMA. Wold menetapkan model MA yang dikembangkan berdasarkan prosedur identifikasi dan evaluasi tiga arah yang efisien (untuk proses AR dan MA). Sedangkan menurut pandangan Victor E. McGee, pengertian Moving Average adalah istilah ini mempunyai dua arti yang berbeda, yang pertama adalah untuk suatu barisan periodik, moving average orde-K adalah rata-rata pergerakan terakhir. K observasi. Yang kedua dalam pemodelan MA disebut "rata-rata bergerak", yang berarti bahwa nilai deret periodik dalam waktu dipengaruhi oleh elemen kesalahan saat ini dan kemungkinan

elemen kesalahan tertimbang. Di model ini, variabel terpengaruh oleh nilai residual periode sebelumnya. Oleh karena itu model MA dengan prinsipnya sama dengan model regresi, namun variabel sisa periode sebelumnya berfungsi sebagai variabel bebas.

ARIMA adalah singkatan dari Autoregressive Integrated Moving Average, suatu teknik yang digunakan untuk memprediksi, mengontrol, dan menganalisis rangkaian siklus. ARIMA Box-Jenkins merupakan hasil karya George Box dan Gwilym Jenkins. Metode ARIMA merupakan metode proyeksi yang menggabungkan metode smoothing, regresi dan dekomposisi. Dalam model ARIMA, variabel bebasnya adalah nilai sebelumnya dari variabel terikat (lagged) dan nilai sisa dari periode sebelumnya.

Berikut langkah-langkah algoritma Arima:

$$Y_t = c + e_t - m_1e_{t-1} - m_2e_{t-2} + \dots + m_qe_{t-q}$$

Di mana :

Y_t = Nilai variabel dependen pada waktu t

C = Konstanta

e_t = Nilai residual pada period ke t

m_1, m_2, m_3 = Koefisien atau parameter dari model Moving Average

$e_{t-1}, e_{t-2}, e_{t-q}$ = Nilai residual sebelumnya (lag)

Besar kecilnya tatanan model moving average bergantung pada besarnya sisa lag yang dimasukkan ke dalam persamaan model. Untuk lebih jelasnya, silakan lihat model berikut:

$$Y_t = c + e_t - m_1 e_{t-1}$$

Model tersebut merupakan model MA orde 1 atau ARIMA (0,0,1).

$$Y_t = c + e_t - m_1 e_{t-1} - m_2 e_{t-2}$$

Model tersebut merupakan model MA orde 2, atau ARIMA (0,0,2).

Model ARIMA merupakan gabungan antara model AR dan model MA, sehingga pada model ini yang menjadi variabel independen adalah nilai variabel dependen sebelumnya (lagged) dan nilai residu periode sebelumnya.

$$Y_t = a + b_1 Y_{t-1} + b_2 Y_{t-2} \dots + b_p Y_{t-p} - m_1 e_{t-1} - m_2 e_{t-2} - \dots - m_q e_{t-q} + e_t$$

Peramalan dengan menggunakan ARIMA terjadi dalam lima tahap, yaitu:

1. Pemeriksaan kestasioneran data
2. Pengidentifikasian model
3. Pengestimasikan parameter model
4. Pengujian model, dan
5. Penggunaan model.

2.6.2 Algoritma *Double Exponential Smoothing*

Menurut (A. Krisma, 2019), Metode *Exponential Smoothing* juga disebut pemulusan atau penghalusan. Adanya bobot (disebut juga konstanta pemulusan)

menunjukkan bahwa metode pemulusan eksponensial mempunyai rata-rata pergerakan tertimbang sebagai eksponensial dari seluruh pengamatan sebelumnya. Nilai aktual menentukan seberapa besar pengaruh observasi terkini terhadap prediksi. Ketika nilainya mendekati 1, maka nilai paling baru dari model ramalan akan disesuaikan secara signifikan oleh adanya kesalahan pada nilai prediksi model sebelumnya. Sebaliknya, nilai prediksi model yang paling baru akan sama dengan model yang paling lama jika nilainya mendekati nol. Oleh karena itu kita harus bisa menentukan nilai optimalnya. Nilai optimal biasanya diwakili oleh nilai minimum MSD (Mean Square Deviation) atau RMSE (Root Mean Square Error), MAPE (Mean Absolute Percent Error) atau MAE (Mean Absolute Error).

Metode *Double Exponential Smoothing* ialah model linier yang dikemukakan oleh Brown. Pada metode ini penghalusan dilakukan sebanyak dua kali. Karena nilai pemulusan tunggal dan ganda tertinggal dari data aktual jika ada elemen tren, prinsip dasar metode pemulusan eksponensial linier Brown mirip dengan rata-rata pergerakan linier.

Berikut langkah-langkah algoritma *Double Exponential Smoothing*:

1. Perhitungan *single exponential smoothing*

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

2. Perhitungan *double exponential smoothing*

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

3. Perhitungan Nilai α_t

$$\alpha_t = S'_t + (S'_t - S''_t)$$

4. Perhitungan Nilai bt

$$bt = \alpha(1 - \alpha)(S^t - S^{t-1})$$

5. Terakhir perhitungan Nilai Ft

$$F_{t+m} = \alpha t + bt m$$


2.7 Pemodelan UML

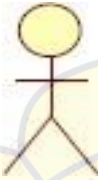


Menurut (Rachmat Destriana, dkk. 2021), UML merupakan bahasa yang digunakan untuk mendefinisikan, memvisualisasikan, mengatur, dan mendokumentasikan artefak (informasi yang digunakan untuk membuat perangkat lunak; artefak dapat berupa perangkat lunak, model, atau deskripsi) pada sistem perangkat lunak seperti pemodelan bisnis dan perangkat lunak tidak standar lainnya. Selanjutnya, UML merupakan bahasa pemodelan yang memakai konsep berorientasi objek. Berikut merupakan alat-alat yang dipakai untuk desain berorientasi objek berbasis UML.

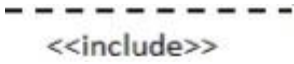

2.7.1 Use Case Diagram

Pada Tabel 2.1 menjelaskan tentang *Use Case Diagram*. *Use Case diagram* yaitu penjelasan fungsi suatu sistem dari prespektif user. Sebuah use case menceritakan bagaimana sebuah sistem digunakan untuk menjelaskan interaksi biasa pengguna pada sistem.

Tabel 2. 1 *Use Case Diagram*

Gambar	Keterangan
	Unit (dilambangkan dengan kata kerja) yang bertukar pesan antara unit



	<p>dan aktor adalah contoh fitur sistem yang disebut UseCase.</p>
	<p>Aktor merupakan abstraksi dari individu atau sistem tambahan yang mengaktifkan fungsi sistem yang dimaksud. Untuk menemukan kegiatan, pembagian kerja dan tanggung jawab perannya dalam sistem sasaran harus ditentukan. Sistem atau individu dapat memainkan berbagai peran. Penting untuk diingat bahwa aktor berinteraksi dengan use case tetapi tidak mengendalikannya.</p>
	<p>kumpulan antara use case dan aktor ditampilkan menjadi garis tanpa panah, yang tidak menampilkan data namun menunjukkan siapa atau apa yang secara langsung meminta interaksi.</p>
	<p>Asosiasi antara aktor dan kasus penggunaan menggunakan panah</p>


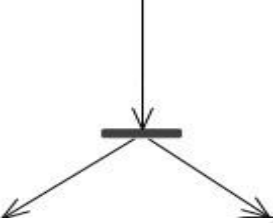
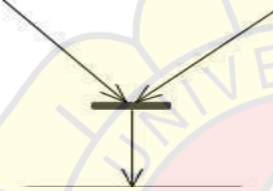


	berongga untuk menunjukkan kapan aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , dalam use case lain (wajib) atau dipanggil oleh use case lain, misalnya memanggil fungsi program
	<i>Extend</i> , yaitu extension dari use case lain jika kondisi atau syarat terpenuhi.

2.7.2 Activity Diagram

Pada Tabel 2.2 menjelaskan tentang *Activity Diagram*. Diagram aktivitas menunjukkan dinamika alami suatu sistem melalui model aliran dan kontrol antar aktivitas. Diagram aktivitas menunjukkan berbagai aliran aktivitas dalam sistem yang sedang dirancang, bagaimana setiap proses dimulai, kemungkinan pilihan, dan bagaimana proses berakhir.

Tabel 2. 2 *Activity Diagram*

Gambar	Keterangan
	<i>Start Point</i> , yang terletak di pojok kiri atas berfungsi sebagai awal kegiatan.
	<i>End Point</i> , penutup aktiviti.





	<p><i>Activities</i>, menggambarkan sebuah bisnis operasi.</p>
	<p><i>Fork/cabang</i>, berguna untuk memperlihatkan pelaksanaan kegiatan secara berurutan atau penggabungan dua kegiatan yang berurutan menjadi satu.</p>
	<p><i>Join</i> (penggabungan) atau <i>rake</i>, berguna untuk menunjukkan adanya dekomposisi.</p>
	<p><i>Decision Points</i>, mendeskripsikan pilihan yang mengatakan benar atau salah.</p>
	<p><i>Swimlane</i>, pemisahan diagram aktivitas, memperlihatkan siapa yang melakukan aktivitas apa.</p>

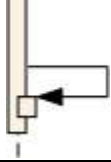


2.7.3 Sequence Diagram

Pada Tabel 2.3 menjelaskan tentang *Diagram Sequence*. Diagram Sequence dapat dipakai untuk mendeskripsi interaksi antar objek pada urutan peristiwa ataupun waktu tertentu pada suatu proses.

Diagram Sequence menggunakan simbol berikut:

Tabel 2. 3 *Sequence Diagram*

Gambar	Keterangan
	<p><i>Entity Class</i>, yaitu komponen sistem yang terjadi dari kumpulan entitas kelas yang menjadikan deskripsi awal sistem dan berperan menjadi dasar pembuatan database.</p>
	<p><i>Boundary Class</i>, memuat sekumpulan kelas yang bertindak sebagai antarmuka ataupun interaksi dengan sistem guna satu atau lebih aktor, seperti menampilkan form entry dan cetak.</p>
	<p><i>Control class</i>, sebuah entitas yang berisi logika aplikasi dan tidak bertanggung jawab atas entitas tersebut; contohnya yaitu aturan bisnis yang mengaitkan banyak objek dan penghitungan.</p>
	<p><i>Message</i>, simbol untuk mengirimkan pesan antar kelas.</p>

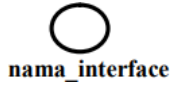


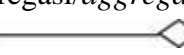
	<i>Recursive</i> , mendeskripsikan pesan yang dikirim secara pribadi.
	Aktivasi berarti melakukan operasi suatu objek, panjang kotak ini sama dengan waktu aktivasi operasi.
	Lifeline, garis putus-putus yang menghubungkan objek dengan lifeline.

2.7.4 Class Diagram

Pada Tabel 2.4 menjelaskan tentang *Class Diagram*. Diagram class menunjukkan ikatan antara kelas dan detail terperinci tentang setiap kelas dalam model desain sistem. Diagram kelas pun memperlihatkan peraturan dan tanggungjawab entitas yang menetapkan perbuatan sistem.

Tabel 2. 4 *Class Diagram*

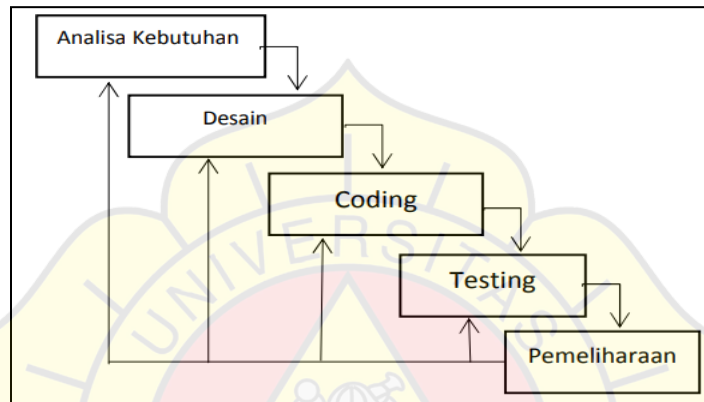
Simbol	Deskripsi
<p style="text-align: center;">Kelas</p> <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <p>Nama_kelas</p> <hr style="border: 0; border-top: 1px solid black;"/> <p>+atribut</p> <hr style="border: 0; border-top: 1px solid black;"/> <p>+operasi()</p> </div>	Kelas bentuk sistem.

<p>Antar muka</p> 	<p>Rancangan antarmuka pemrograman berorientasi objek.</p>
<p><i>Association</i></p> 	<p>Hubungan kelas-ke-kelas memiliki makna yang sama, dan seringkali terkait dengan multiplisitas.</p>
<p>Asosiasi berarah</p> 	<p>Hubungan antar class menunjukkan bahwa sebuah class digunakan oleh class lain, dan asosiasi biasanya disertai dengan multiplisitas.</p>
<p>Generalisasi</p> 	<p>Generalisasi, generalisasi, dan spesialisasi mempengaruhi hubungan antar class.</p>
<p>Kebergantungan / <i>dependency</i></p> 	<p>Ketergantungan antar kelas dipengaruhi oleh hubungan antar kelas.</p>
<p>Agregasi/<i>aggregation</i></p> 	<p>Hubungan yang ada di antara kelas memiliki makna umum</p>

2.8 Metode Pengembangan Sistem

Menurut (Julianto & Setiawan, 2019), Perancangan sistem informasi ini dilakukan dengan menggunakan proses waterfall melalui Systems Development Life Cycle

(SDLC). Kerangka kerja yang terstruktur secara berurutan yang dimaksudkan untuk meningkatkan sistem informasi (piranti lunak) dikenal sebagai SDLC. Pada Gambar 2.4, *Waterfall* ialah metodologi peningkatan sistem klasik serta setidaknya banyak dipakai oleh *developer* sistem. Mengenai bagan ataupun fase *waterfall* ialah seperti gambar berikut ini:



Gambar 2. 4 Metode Waterfall (Julianto & Setiawan, 2019)

1. Analisa Kebutuhan

System pada sesi ini ialah sesi Analisis kebutuhan sistem untuk memudahkan proses pembuatan dan peningkatan sistem. Analisis kebutuhan fungsional, sistem usulan, proses bisnis yang bergerak, dan masalah dilakukan selama sesi ini.

2. Desain

Sesi ini ialah proses pembuatan tampilan sistem yang disamakan dengan hasil analisis kebutuhan yang sudah dicoba untuk sesi analisa.

3. *Coding* (Pengkodean)

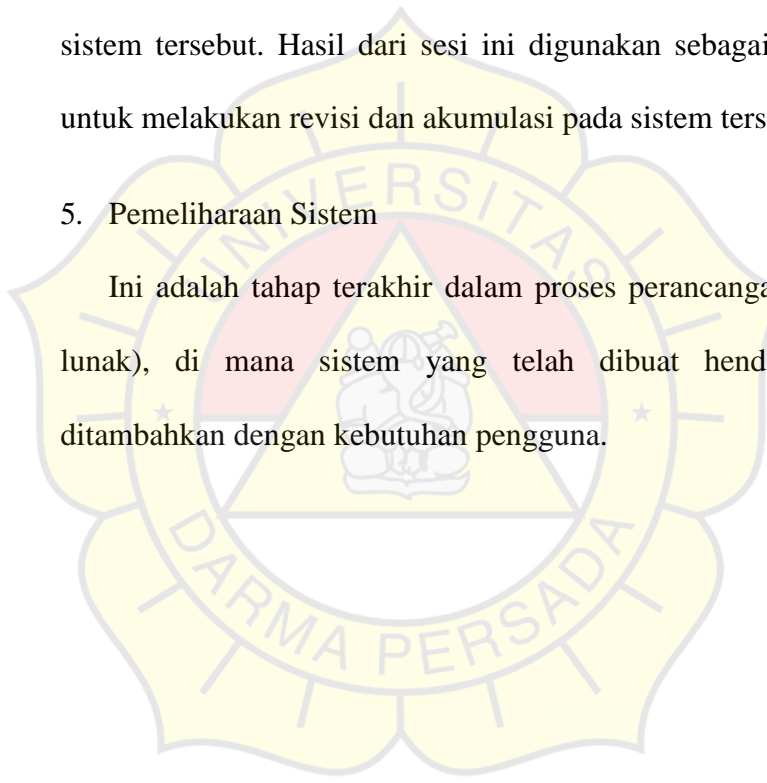
Ini adalah sesi perwujudan sistem yang dimulai dengan mengintegrasikan desain sistem yang sudah dibuat dengan Visual Basic dan desain database dengan DBMS MySQL.

4. *Testing* (Pengujian Sistem)

Ini adalah sesi uji sistem untuk mengetahui seberapa baik dan buruk sistem tersebut. Hasil dari sesi ini digunakan sebagai bahan penilaian untuk melakukan revisi dan akumulasi pada sistem tersebut.

5. Pemeliharaan Sistem

Ini adalah tahap terakhir dalam proses perancangan sistem (piranti lunak), di mana sistem yang telah dibuat hendak dirubah dan ditambahkan dengan kebutuhan pengguna.





TEKNOLOGI INFORMASI

UNIVERSITAS DARMA PERSADA