

BAB II

LANDASAN TEORI

2.1 Data mining

Menurut (Mustika et al., 2021), Data mining ialah metode yang memungkinkan penggunaannya untuk mengakses data dalam skala besar dalam durasi yang relatif singkat. Atau dengan kata lain, data mining merupakan alat dan aplikasi yang melakukan analisis statistik terhadap data melalui proses penggalian atau penggalian data dan informasi yang sebelumnya tidak diketahui. Sederhananya, data mining adalah proses penggalian data untuk menemukan data baru dengan memilih desain serta aturan tertentu dalam informasi yang berjumlah besar, jadi cara kerja data mining sebenarnya adalah memeriksa basis data yang luas untuk menemukan desain atau bentuk baru yang dapat digunakan. untuk membuat keputusan yang berguna.

Data mining adalah analisis kumpulan data untuk menemukan hubungan yang tidak terduga dan merangkum data dengan cara yang berbeda dari sebelumnya, yang dapat dimengerti dan berguna bagi pemilik data. Data mining merupakan salah satu bidang diantara beberapa bidang keilmuan yang menggabungkan pembelajaran mesin, pengenalan pola, statistika, database dan teknik visualisasi untuk menangani masalah pengambilan informasi dari database yang besar. Data mining adalah istilah yang digunakan untuk menggambarkan penemuan pengetahuan dalam database. Penambangan data adalah proses yang menggunakan statistik, matematika, kecerdasan buatan, dan teknik pembelajaran mesin untuk mengekstrak dan mengidentifikasi informasi berguna dan pengetahuan

terikat dari database besar. Berdasarkan pengertian data mining yang telah dijelaskan di atas Data mining adalah pengetahuan yang tersembunyi dalam database yang diproses untuk menemukan pola dan teknik statistik matematika, kecerdasan buatan dan pembelajaran mesin untuk mengekstrak dan mengidentifikasi informasi pengetahuan dari database. (Utomo & Mesran, 2020).

2.2 Fungsi Data mining

Menurut (Mustika et al., 2021), Penambangan data dimasukkan ke dalam berbagai disiplin ilmu seperti kecerdasan buatan, sistem informasi, statistik, pembelajaran mesin, ilmu manajemen, dan basis data. Pada dasarnya data mining memiliki 7 (tujuh) fungsi dasar yaitu *Prediction, Sequencing, Classification, Association, Clustering, Forecasting, Description*.

2.2.1 Prediction

Prediction merupakan proses menemukan pola yang hilang dari data dengan menggunakan analisis regresi buat menemukan data yang hilang. Jika label kelas tidak ada, klasifikasi dipakai untuk prediksi. Ada dua cara buat memprediksi data. Pertama, prediksi data yang tidak tersedia atau hilang memakai analitik prediktif. Kedua, label kelas diprediksi menggunakan model kelas yang dibangun sebelumnya.

2.2.2 Sequencing

Sequencing merupakan bentuk jamak dari asosiasi, dan fungsinya merupakan proses penentuan hubungan yang berbeda dari sekumpulan data yang diperoleh.

2.2.3 Classification

Fungsi *Classification* merupakan meringkas beberapa definisi karakteristik dalam 1 atau lebih kelompok. Ataupun dengan kata lain, ini ialah tindakan yang menetapkan grup ke setiap negara bagian, di mana setiap negara bagian berisi sekumpulan atribut, salah satunya ialah atribut kelas. Ini memang membutuhkan penemuan model yang dapat menginterpretasikan atribut kelas sebagai fungsi dari atribut input.

Klasifikasi merupakan pekerjaan mengevaluasi objek data untuk menempatkannya ke dalam kategori tertentu dari sejumlah kategori yang tersedia. Klasifikasi membangun model dari data pelatihan yang ada dan kemudian menggunakan model tersebut untuk mengklasifikasikan data baru. Klasifikasi dapat didefinisikan sebagai tugas melakukan pelatihan/pembelajaran pada fungsi tujuan yang memetakan setiap kumpulan atribut (fitur) ke satu nomor label kelas yang tersedia. Sistem yang melakukan klasifikasi diharapkan dapat mengklasifikasikan seluruh kumpulan data dengan benar, namun tidak dapat dipungkiri bahwa kinerja sistem tidak bisa 100% benar, sehingga sistem klasifikasi juga harus mengukur kinerjanya. (Utomo & Mesran, 2020).

2.2.4 Association

Association merupakan untuk mengidentifikasi hubungan antara peristiwa yang terjadi pada saat tertentu dengan menginduksi definisi karakteristik dari suatu data.

2.2.5 Clustering

Clustering, juga dikenal sebagai segmentasi, bisa digunakan buat mengidentifikasi kelompok alami kasus berdasarkan sekumpulan atribut dengan mengelompokkan data dengan atribut serupa.

2.2.6 Forecasting

Forecasting adalah salah satu fungsi dari data mining, yang tujuannya ialah untuk memperkirakan hasil data masa depan berdasarkan pola dalam kumpulan data yang besar.

(*Forecasting*) adalah memprediksi apa yang akan terjadi di masa depan, sedangkan perencanaan adalah menentukan apa yang harus dilakukan di masa depan. Tentu saja ada perbedaan antara peramalan dan perencanaan. Perkiraan data serial yang dibuat biasanya didasarkan pada data masa lalu yang dianalisis menggunakan beberapa metode. Data masa lalu dikumpulkan, dipelajari, dan dianalisis dari waktu ke waktu untuk mencoba menggambarkan apa yang akan terjadi di masa depan. Keakuratan perkiraan berbeda-beda untuk setiap masalah dan berbagai faktor, dan keakuratan perkiraan tidak selalu 100% akurat, namun bukan berarti perkiraan tidak penting. Peramalan telah banyak digunakan dan berperan baik dalam berbagai situasi manajemen, memberikan dukungan untuk perencanaan, pemantauan dan pengambilan keputusan. (Pulungan & Pristiwanto, 2021).

2.2.7 *Description*

Description dalam data mining dipakai buat lebih memahami karakteristik utama dari data. Penggunaan fungsi pendeskripsian membantu menemukan beberapa pola tersembunyi dalam data. Dengan istilah lain, jika terdapat pola berulang pada data, dan memiliki nilai, maka dapat diketahui karakteristik data tersebut.

2.3 Model data mining

Menurut (Urva et al., 2023) pengguna umumnya menggunakan dua jenis model operasional untuk mendapatkan informasi yang dibutuhkan dalam data mining. proses adalah:

1. Validasi Model

Model memakai pendekatan top-down, mengambil hipotesis dari pengguna kemudian menguji validitasnya dengan menggunakan data yang dapat diterapkan sehingga hipotesis tersebut selanjutnya dapat dibuktikan kebenarannya.

2. Model knowledge discovery

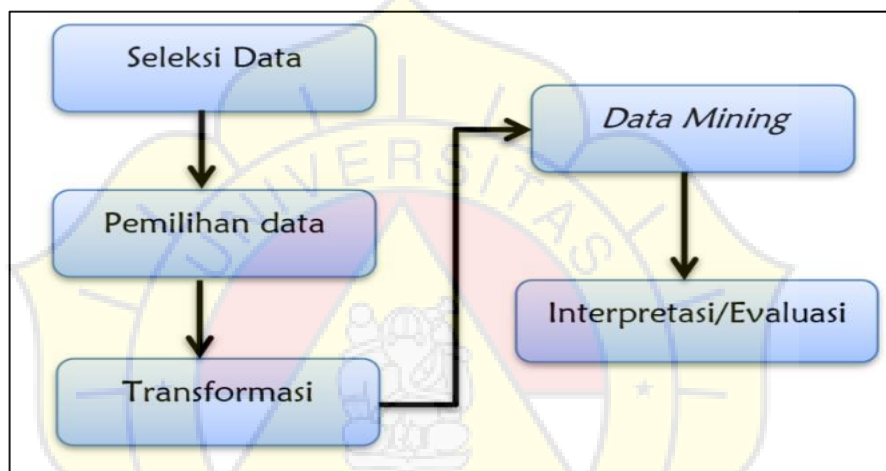
Model memakai pendekatan bottom-up untuk menghasilkan informasi yang tidak diketahui sebelumnya. Knowledge Discovery Model terbagi 2 yaitu:

1. Directed knowledge discovery, pada model ini proses data mining mencoba memilih nilai target dari beberapa field (seperti income, response, age, dll) ke field lainnya.

2. Undirected Knowledge discovery, model ini tidak memiliki domain target karena komputer mencari pola dalam data.

2.4 Proses Data mining

Menurut (Urva et al., 2023), Ada sebagian tahapan dalam mekanisme data mining. Alur mekanisme dari setiap tahapan proses data mining menetapkan pada gambar berikut:



Gambar 2. 1 Tahapan mekanisme data mining (Urva et al., 2023).

1. Seleksi Data

Penting untuk memilih data dari sekelompok data operasional sebelum menambang informasi KDD. Informasi yang dipilih disimpan dalam file, tetapi terpisah dari basis data operasional. Data yang terpilih kemudian digunakan pada tahap selanjutnya.

2. Pemilihan Data

Pemilihan data adalah proses deduplikasi data. Periksa konsistensi data. Membetulkan kesalahan informasi, semacam kesalahan tipografi.

Termasuk di dalamnya proses pengayaan, yaitu proses augmentasi informasi serta data lain yang diperlukan oleh KDD, semacam informasi serta data eksternal.

3. Transformasi

Selama fase transformasi, informasi yang diseleksi dikodekan sehingga informasi tersebut cocok dengan kebutuhan mekanisme pembuatan serta disesuaikan dengan tipe data ataupun skema yang hendak dicari di database.

4. Data Mining

Mekanisme penggunaan metode atau teknik tertentu untuk menemukan sistem ataupun data yang menarik pada data yang terpilih. Teknik, metode serta algoritma dalam data mining sangat beragam. Memilih algoritma yang pas mempengaruhi keseluruhan tujuan serta proses KDD.

5. Interpretasi/Evaluasi

Selanjutnya, pada tahap akhir, setelah menemukan pola data yang dihasilkan oleh mekanisme data mining, diperlukan representasi pihak berkepentingan yang dapat dimengerti. Evaluasi pada tahap ini berkaitan dengan memeriksa pola atau data yang bertentangan atau tidak berlawanan dengan kenyataan serta asumsi sebelumnya.

2.5 Pemulusan Eksponensial (Exponential Smoothing)

Peramalan Exponential Smoothing merupakan salah satu tata cara dalam time series yang memakai pembobotan informasi masa kemudian secara exponential. Dalam perihal ini, tata cara Exponential Smoothing dibagi atas 3 ialah

Single Exponential Smoothing yang dipakai buat pemulusan informasi yang stasioner, tidak mempunyai trend serta variasi musiman sehingga cuman memakai satu parameter yakni α , Double Exponential Smoothing yang digunakan dalam pemulusan informasi yang muat trend tetapi tidak mempunyai variasi musiman, mempunyai 2 parameter pemulusan ialah α serta β , serta Triple Exponential Smoothing merupakan pemulusan exponential smoothing yang bertujuan buat melaksanakan pemulusan pada informasi yang muat trend, serta variasi musiman dengan memakai 3 parameter pemulusan ialah α , β , serta γ (Lestari et al., 2020).

Pemulusan eksponensial (exponential smoothing) merupakan tata cara peramalan rata-rata bergerak (moving average) dengan metode memastikan pembobotan secara exponential terhadap nilai pengamatan ataupun informasi historis. Tata cara exponential smoothing dibagi jadi 3 ialah, single exponential, double exponential, triple exponential. Tata cara single exponential smoothing dipakai pada informasi time series yang tidak ada baik pola informasi tren ataupun musiman (seasonal). Tata cara double exponential dipakai pada informasi time series yang ada pola informasi tren tetapi tidak memiliki pola informasi musiman. Pada informasi time series yang ada pola informasi musiman (seasonal) tata cara yang bisa dipakai ialah triple exponential smoothing (Febriyanti & Rifai, 2022).

2.6 Triple Exponential Smoothing

Menurut (Andrian et al., 2020), Metode Triple exponential smoothing “*Winter’s Method*” ataupun sering diketahui dengan istilah *Holt - Winter*, adalah kenaikan dari metode pemulusan eksponensial ganda sebagai halnya diperlukannya parameter-parameter ialah level (α), trend (β), serta seasonal (γ) dalam

melaksanakan prediksi . Ada 2 model seasonal buat triple exponential smoothing ialah model Additive Seasonal serta model Multiplicative Seasonal. untuk model Additive Seasonal , tingkat serta trend ditambahkan melalui perhitungan Seasonal untuk model Multiplicative Seasonal , tingkat serta trend dikalikan melalui perhitungan Seasonal. Sebab data mempunyai pola Additive Seasonal, hingga yang bakal diulas di sini cuma triple exponential smoothing dengan Additive Seasonal Model. Proses inisialisasi ataupun penentuan nilai dini untuk peramalan menggunakan tata cara pemulusan eksponensial Holt - Winters ini dibutuhkan paling tidak ada pola trend serta musiman pada 1 periode dengan indeks musiman $lt - L$, serta butuh juga buat menaksir aspek kecenderungan dari 1 fase ke fase berikutnya , semacam untuk sesuatu informasi ada pola tren serta pula musiman sebesar 3 , 4 , 6 ataupun juga 12 untuk 1 periode.

Metode Triple Exponential Smoothing ialah metode dalam time series yang didasarkan atas 3 persamaan pemulusan ialah satu buat stasioner, trend, serta musiman dengan memakai 3 parameter pemulusan ialah α, β, γ yang tiap- tiap bernilai antara 0– 1. (Lestari et al., 2020).

Triple Exponential Smoothing tidak cuma hanya Single Exponential Smoothing yang di looping sebanyak 3 kali. Perihal itu disebabkan Triple Exponential Smoothing berawal dari Single Exponential Smoothing. Tetapi, pada Triple Exponential Smoothing masih ada perhitungan lanjutan. Sehingga, bisa dikatakan Triple Exponential Smoothing bukan sematamata Single Exponential Smoothing yang diulang sebanyak 3 kali, sebab masih ada perhitungan lanjutan pada prosesnya. Sehingga, hasil yang diperoleh lebih smooth dibanding dengan Single Exponential Smoothing (Bayu Saputro et al., 2022).

Berikut langkah-langkah metode Triple Exponential Smoothing:

1. Menghitung nilai Single Exponential Smoothing (SES) dengan rumus 1.

$$St' = \alpha X_t + (1 - \alpha)St-1'$$

2. Menghitung nilai Double Exponential Smoothing (DES) dengan rumus 2

$$St'' = \beta St' + (1 - \beta)St-1''$$

3. Menghitung nilai Triple Exponential Smoothing (TES) dengan rumus 3.

$$St''' = \gamma St'' + (1 - \gamma)St-1'''$$

4. Mencari nilai at dengan rumus 4.

$$at = 3St' - 3St'' + St'''$$

5. Mencari nilai bt dengan rumus 5.

$$bt = \alpha / 2(1-\alpha)^2 [(6 - 5\alpha)St' - (10 - 8\beta)St'' + (4 - 3\gamma)St''']$$

6. Mencari nilai ct dengan rumus 6.

$$ct = \alpha^2 / (1-\alpha)^2 (St' - 2St'' + St''')$$

7. Terakhir yaitu mencari nilai Ft dengan rumus 7.

$$F_{t+m} = at + btm + 0,5ctm^2$$

Keterangan:

St' = nilai SES

St'' = nilai DES

St''' = nilai TES

X_t = data aktual

α = nilai alpha yang nilainya antara 0-1 ,

a_t, b_t, c_t = konstanta smoothing

F_{t+m} = periode yang di ramalkan.

2.7 Model Autoregressive (AR)

Menurut (Putri & Aghsilni, 2019). Model autoregressive (AR) yakni model stasioner dari informasi deret waktu dimana nilai pengamatan waktu ke- t dipengaruhi oleh nilai pengamatan lebih dahulu. Model autoregressive dengan orde p dinotasikan dengan AR(p). Wujud universal model AR(p) merupakan:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + e_t$$

2.8 Model Moving Average (MA)

Menurut (Putri & Aghsilni, 2019). Model moving average (MA) menampilkan pengamatan pada waktu t , X_t dipengaruhi oleh keliru pada q waktu-waktu t lebih dahulu. Model MA dengan orde q dinotasikan dengan MA(q). Wujud universal model MA(q) merupakan:

$$X_t = e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \dots - \theta_q e_{t-q}$$

2.9 Model ARMA (Autoregressive Moving Average)

Dalam permasalahan analisis informasi runtun waktu, proses AR maupun MA lumayan mencukupi, namun kadang ditemui permasalahan dimana identifikasi model menciptakan kesimpulan kalau informasi menjajaki proses AR sekalian MA

maupun sebagian menjajaki proses AR sebaliknya sebagian lagi menjajaki proses MA. Dalam perihal ini informasi dikatakan menjajaki proses ARMA.(Putri & Aghsilni, 2019).

Model umum untuk campuran proses AR dan MA adalah seperti berikut:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \dots - \theta_q e_{t-q}$$

Dimana ϕ_i dan θ_i adalah berturut-turut koefisien *autoregressive* dan koefisien *moving average*. Model ARMA dapat dituliskan seperti :

$$(1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p) X_t = (1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q) e_t$$

atau bisa ditulis $\phi_p(B) X_t = \theta_q(B) e_t$

2.10 Autoregressive Integrated Moving Avarage (ARIMA)

Model Autoregressive Integrade Moving (ARIMA) yakni salah satu tata cara yang gunakan buat peramalan informasi yang bertabiat time series. Model Autoregressive Integrated Moving Average(ARIMA) yakni model yang secara penuh mengabaikan independen variabel dalam membuat prediksi. ARIMA mengenakan nilai masa lalu kemudian serta saat ini dari variabel dependen buat menciptakan prediksi jangka pendek yang akurat. ARIMA sangat baik ketepatannya buat prediksi jangka pendek, sebaliknya buat prediksi jangka panjang prediksinya kurang baik. Biasanya hendak cenderung flat(mendatar/ konstan) buat periode yang lumayan panjang.

ARIMA yakni tata cara peramalan yang dalam pembuatan modelnya bersumber pada pengaruh waktu dengan memakai informasi masa kemudian serta saat ini selaku peubah yang silih terpaat. Model ARIMA bisa dinotasikan jadi

ARIMA(p, d, q), dimana p merupakan orde dari proses Autoregressive(AR), q ialah orde dari proses Moving Average(MA), serta d yakni orde Differencing.(Catur Putri & Junaedi, 2022). Terdapat beberapa model umum Arima sebagai berikut:

a. Model *Autoregressive* (AR(p))

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + a_t$$

b. Model *Moving Average* (MA(q))

$$X_t = a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \dots - \theta_q a_{t-q} + a_t$$

c. Model *Autoregressive Moving Integrated Average* (ARIMA(p,d,q))

$$Y_t = (1 - B)^d X_t$$

$$Y_t = \phi_1 Y_{t-1} + \dots + \phi_p Y_{t-p} + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q}$$

$$\text{Atau } \phi_p(B)(1-B)^d Y_t = \theta_q(B)a_t$$

Dimana:

$$\Phi(B) = 1 - \phi_1 B - \dots - \phi_p B^p$$

$$\theta_q(B) = 1 - \theta_1 B - \dots - \theta_q B^q$$

Keterangan :

Φ : Parameter *Autoregressive*

θ : Parameter *Moving Average*

a_t : Nilai *error* ke -t

Y_t : *Series* yang *stasioner*

2.11 Pemrograman aplikasi

2.11.1 PHP

Menurut (Mundzir MF, 2018) PHP berawalan dari kata “*Php Hypertext Preprocessor*”, ialah bahasa codingan yang umum buat penanganan pengembangan serta pembuatan suatu website serta dapat dipakai bertepatan dengan HTML. Dikala ini, PHP sering di pakai buat mengerjakan program website web dinamis. Contoh penerapan program PHP merupakan MediaWiki(software di balik Wikipedia) serta forum(phpBB). Sebaliknya, *Xaraya, Postnuke, Mambo, Joomla!*, serta lain- lain ialah contoh aplikasi yang lebih lingkungan berbentuk CMS serta dibentuk memakai PHP. Tidak hanya itu, PHP pula jadi opsi lain dari *ColdFusion Macromedia , JSP/ Java Sun Microsystems, CGI/ Perl , serta ASP. NET/ C#/ VB. NET Microsoft*, guna membuat maupun melaksanakan programnya.

Php sebagai sekelompok skrip ataupun bahasa pemrograman mempunyai tugas utama, ialah sanggup mengelompokan serta mengevaluasi hasil survei ataupun wujud apa juga ke server database serta pad sesi berikutnya hendak menghasilkan dampak beruntun. Dampak beruntun PHP ini berbentuk aksi dari skrip lain yang hendak melaksanakan interaksi dengan database, mempertemukan serta menggabungkan data, setelah itu memperlihatkannya saat terdapat tamu web memerlukannya(menunjukkan data cocok permintaann user).

2.11.2 HTML

Menurut (Rohi Abdulloh, 2018) *Hypertext Markup Language* ialah kepanjangan dari HTML, adalah bahasa standar website yang dijalankan penggunaannya dengan W3C berbentuk tag- tag yang mengatur tiap elemen dari web. HTML berfungsi selaku pengelola bentuk halaman web yang meletakkan tiap elemen web setara layout yang di idamkan.

HTML umumnya ditaruh dalam suatu file berekstensi .html. Buat mengetik skrip HTML bisa memakai text editor semacam *Notepad* selaku wujud sangat simpel ataupun text editor khusus yang bisa mengidentifikasi tiap faktor skrip HTML serta memperlihatkan dengan warna yang berbeda maka dari itu gampang dibaca, semacam *Notepad++*, *Sublime Text*, serta aplikasi lainnya yang sejenisnya.

2.11.3 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 event yang terjalin pada halaman. Javascript ialah perekat yang menggabungkan halaman-halaman website.

Javascript ialah bahasa Skript yang mengoptimalkan halaman-halaman HTML. Seluruh platform hampir bisa menjalankan javascript. Javascript ialah bahasa sisi- klien yang di desain untuk browser pc kamu, tidak untuk server. Dia dibentuk dengan langsung ke dalam *Browser*.

Mozilla Firefox, Microsoft Internet Explorer, Goggle Chrome, Opera, nyaris seluruh browser.

Sebab javascript terpaut dengan browser, dia sangat terintegrasi dengan HTML. Kala browser muat suatu halaman, server hendak mengirim konten utuh dari dokumen, tercantum HTML serta statemen- statemen javascript. Konten HTML setelah itu dibaca serta di interpretasi baris demi baris hingga tag pembuka javascript dibaca, pada dikala itu interpreter javascript mengambil alih. Kala tag penutup javascript diraih, pemroses HTML bersinambung.

2.11.4 CSS

Menurut (Salamah, 2021) dalam e-book nya yang berjudul "Tutorial Cascading Style Sheets (CSS)" CSS ialah kependekan dari "Cascading Style Sheets". CSS mempunyai sifat "Style Sheet Language" yaitu bahasa pemrograman buat desain web. CSS ialah bahasa pemrograman yang dipakai buat merancang halaman website, CSS menggunakan tag yang disebut Id dan Class. Seiring perkembangan dunia pemrograman dan teknologi, CSS tidak hanya untuk HTML dan XHTML. Namun sudah bisa digunakan untuk mendesain tampilan aplikasi Android.

(CSS) *Cascading Style Sheet* merupakan script pemrograman yang dirancang buat menghias dan mendesain kerangka atau tata letak halaman web supaya lebih elegan dan mengumkan. CSS ialah dokumen mandiri yang bisa disertakan dalam script HTML serta hanya direferensikan oleh HTML saat menentukan gaya. CSS dapat melakukan lebih dari bahasa pemrograman inti seperti HTML dan PHP. Saat memakai CSS, Anda bisa

mengatur jenis font, ukuran kolom, garis antar paragraf , warna teks, dan jenis latar belakang yang digunakan. Tidak itu saja, CSS juga mampu merancang layout, menampilkan perubahan pada perangkat yang lain, serta berbagai efek hendak digunakan pada web. CSS sangat sederhana dipelajari dan powerfull karena bisa mengontrol bentuk dokumen HTML. Dari yang simple hingga yang kompleks. Tidak kaget jika CSS saat ini digunakan dalam kombinasi dengan HTML dan PHP di hampir semua jenis situs web (Noviana, 2022).

2.11.5 Database

Menurut (Fitri, 2020) Database ialah sekumpulan data yang terorganisir, yang umumnya disimpan dan diakses secara elektronik dari suatu sistem komputer. Pada saat pangkalan data menjadi semakin kompleks, maka pangkalan data dikembangkan menggunakan teknik perancangan dan pemodelan secara formal. Piranti lunak yang dapat dipakai buat mengelola basis data disebut sistem manajemen basis data (database management system) atau disingkat DBMS. DBMS ialah piranti lunak yang dirancang untuk bisa melakukan pengaturan dan mengelola koleksi data dalam jumlah yang besar dan dapat memanipulasi data secara lebih mudah. DBMS merupakan interface atau antar muka antara pengguna basis data (baik pengguna DBMS langsung maupun aplikasi) dengan data yang disimpan. RDBMS ataupun relationship database manajemen system ialah salah satu jenis DBMS yang mendukung hubungan antar tabel. Contoh RDBMS di antaranya adalah Oracle, MySQL , MS SQL Server, DB2, Ms Access.

2.11.6 MYSQL

Menurut (Fitri, 2020) MySQL ialah DBMS yang open source dengan 2 lisensi, merupakan free software serta shareware. Jadi MySQL ialah basis data server yang leluasa bersama dengan GNU *General Public License*(GPL) lalu kita bisa gunakan buat keperluan individu maupun komersial tanpa wajib membayar lisensi yang terdapat. Sebagaimana yang telah disebutkan, MySQL masuk ke dalam tipe RDBMS(*Relational database Management Sistem*). Hingga dari itu, sebutan semacam tabel, kolom, baris, digunakan pada MySQL. Contohnya di dalam MySQL suatu basis data ada satu maupun sebagian tabel.

MySQL ialah aplikasi yang terkategori basis data server serta bersifat Open source. Open source memberitahukan kalau aplikasi ini dilengkapi dengan *source code* (kode yang digunakan buat membikin MySQL), tidak hanya itu pasti saja wujud executable- nya ataupun kode yang bisa dilakukan secara langsung dalam system. (Ricoida et al., 2019).

2.11.7 XAMPP

Menurut (Aprilian & K Saputra, 2020) XAMPP merupakan piranti lunak sumber terbuka yang dibesarkan oleh Apache. Paket piranti lunak XAMPP menyediakan penyaluran Apache buat MariaDB, server Apache, Hypertext Preprocessor (PHP), serta Practical Extraction and Reporting Language(Perl). Serta itu pada dasarnya merupakan tuan rumah lokal ataupun 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. Wujud lengkap XAMPP adalah X kependekan dari Cross- Platform, (A) Server Apache,(M) MariaDB,(P) PHP serta (P) Perl. Cross- platform umumnya yaitu bisa bergerak di pc manapun dengan OS apapun.

2.11.8 Visual Studio Code

Menurut (Permana & Romadlon, 2019) Visual Studio Code (VS Code) ini yakni suatu bacaan editor ringan dan profesional yang terbuat oleh Microsoft buat sistem pembedahan multiplatform, maksudnya ada pula buat tipe Linux, Mac, serta Windows. Bacaan editor ini secara langsung menunjang bahasa pemrograman JavaScript, Node.js serta Typescript, dan bahasa pemrograman yang lain dengan dorongan plugin yang bisa dipasang via marketplace Visual Studio Code(semacam C++, C#, Python, Java, Go, dst). Banyak sekali fitur- fitur yang disediakan oleh Visual Studio Code, antara lain Intellisense, Debugging, Git Integration, serta fitur ekstensi yang menaikkan keahlian bacaan editor. Fitur- fitur tersebut hendak terus meningkat bersamaan dengan bertambahnya tipe Visual Studio Code. Pembaruan tipe Visual Studio Code ini pula dicoba berkala tiap bulan, serta inilah yang membedakan VS Code dengan bacaan editor- teks editor yang lain. 21 Bacaan editor VS Code pula bertabiat open source, yang mana kode sumbernya bisa kamu amati serta kamu bisa berkontribusi buat pengembangannya. Kode sumber dari VS Code ini juga bisa dilihat di link Github. Perihal ini pula yang membuat VS Code jadi kesukaan para

pengembang aplikasi, sebab para pengembang aplikasi dapat turut dan dalam proses pengembangan VS Code ke depannya.

2.12 Pemodelan UML

Menurut (Destriana, 2021) UML ialah bahasa buat memvisualisasi, membangun, menspesifikasi serta mendokumentasikan artifacts(bagian dari data yang dipakai buat ditampilkan oleh mekanisme pembuatan piranti lunak, artifact tersebut bisa berbentuk model, uraian ataupun piranti lunak) dari sistem piranti lunak, seperti untuk pemodelan bisnis serta pola non piranti lunak yang lain. Tidak hanya itu UML merupakan bahasa pemodelan yang memakai konsep orientasi object. Dibawah bendera Rasional Software Corps UML di dirikan oleh James Rumbaugh, Ivar Jacobson, serta Grady Booch. UML sediakan notasi- notasi yang menolong memodelkan sistem dari bermacam prespektif. UML bukan cuma dipakai dalam pemodelan piranti lunak, tetapi nyaris dalam seluruh bidang yang memerlukan pemodelan.

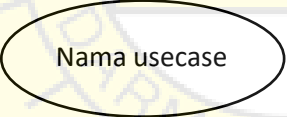

Unified Modeling Language(UML) ialah suatu bahasa pemodelan piranti lunak yang sudah distandardisasi selaku media penyusunan cetak biru(blueprints) piranti lunak(Pressman). UML mampu saja dipakai buat spesifikasi, visualisasi, dokumentasi serta kontruksi sekian banyak bagian- bagian dari sistem yang terdapat dalam piranti lunak. dengan kata lain, semacam halnya seseorang arsitek dalam membuat dokumen cetak biru yang dipakai oleh industri konstruksi guna membangun suatu bangunan, arsitek piranti lunak membuat diagram- diagram UML guna menolong programmer/ pengembang membuat perangkat lunak. Guna

berikutnya, semakin kita mengenali sebagian kosakata yang dipakai UML, kita akan semakin awam dalam menguasai spesifik (Abdillah, 2021).

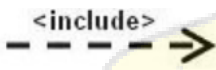
2.12.1 Use Case Diagram

Use case diagram ialah pemodelan guna kegiatan system data yang hendak dibentuk. Use case menjelaskan suatu hubungan antara 1 ataupun lebih aktor menggunakan system data yang hendak dibentuk. Use case dipakai buat mengenali peranan apa saja yang terdapat untuk suatu sistem data serta siapa saja yang berwenang memakai fungsi- fungsi tersebut (Julianto & Setiawan, 2019) . Berikut ini adalah simbol-simbol diagram use case, seperti terlihat pada Tabel 2.1 berikut ini:

Tabel 2. 1 Use Case diagram (Julianto & Setiawan, 2019).

Simbol	Penjelasan
Use Case 	Fungsionalitas yang ditunjukkan oleh system ketika komponen berpindah pesan antar unit ataupun aktor; sering ditunjukkan menggunakan kata kerja di awal frase nama use case.
Aktor  Nama aktor	Proses , orang , serta system lain yang berhubungan dengan sistem informasi yang sedang dibuat terletak di luar sistem informasi yang sedang dibuat itu sendiri ,



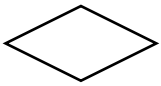

	<p>sehingga aktor belum tentu orang meskipun simbol aktor ialah gambar seseorang; mayoritas kata benda digunakan di depan frase nama aktor untuk mengekspresikan.</p>
<p>Asosiasi</p> <p>_____</p>	<p>Interaksi antar aktor serta use case yang berpartisipasi untuk use case atau use case mempunyai interaksi dengan actor.</p>
<p>Extend</p> <p><<extends>></p> <p>-----></p>	<p>Hubungan use-case terlampir ke use-case yang menambahkan use-case bisa berdiri sendiri bahkan tanpa use-case tambahan; sama dengan dasar pewarisan pada pemrograman berorientasi objek; Biasanya use case lain memiliki nama yang sama, seperti panah yang menunjuk ke use case yang ditambahkan; biasanya, use case yang diperluas memiliki jenis yang sama dengan use case induknya.</p>
<p>Generalization</p> <p>—————></p>	<p>Relasi generalisasi serta spesialisasi (Umum-Khusus) Jarak antara 2 use case di mana satu fungsi merupakan</p>



	<p>fungsi umum dari fungsi lainnya, misalnya: arah panah menunjuk ke use case yang digeneralisasikan (generik)</p>
<p>Include</p> 	<p>Hubungan Use case tambahan adalah hubungan ke use case, dimana use case yang ditambahkan membutuhkan use case untuk menjalankan fungsinya atau berfungsi sebagai syarat untuk eksekusi use case. Ada dua pandangan utama tentang penyertaan dalam kasus penggunaan:</p> <ol style="list-style-type: none"> 1. Include artinya use case tambahan akan selalu dipanggil saat use case tambahan dijalankan. 2. Include artinya use case tambahan akan selalu dicek apakah use case yang ditambahkan sudah dieksekusi sebelum use case tambahan dijalankan.

2.12.2 Activity Diagram

Activity diagram ataupun diagram aktivitas menggambarkan alur kerja serta aktivitas dari suatu system serta mekanisme bisnis ataupun menu dalam piranti lunak. Diagram aktivitas lebih menekankan pada penggambaran kegiatan system ataupun aktivitas yang bisa digunakan sistem, dari pada apa yang diperbuat aktor (Julianto & Setiawan, 2019). Berikut ini adalah simbol-simbol aktivitas diagram, seperti terlihat pada Tabel 2.2 di bawah ini:

Tabel 2. 2 Simbol Activity diagram (Julianto & Setiawan, 2019).


Simbol	Penjelasan
Status awal 	Keadaan awal aktivitas sistem, diagram aktivitas mempunyai keadaan awal.
Aktivitas 	Suatu kegiatan yang diperbuat oleh sistem, sering didahului oleh kata kerja.
Percabangan 	Asosiasi cabang apabila lebih dari satu opsi aktivitas.
Penggabungan 	Untuk menyatukan asosiasi, untuk menyatukan lebih dari 1 aktivitas mewujudkan satu aktivitas.



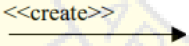
<p>Status akhir</p> 	<p>Keadaan akhir dari eksekusi system, diagram aktivitas mempunyai keadaan akhir.</p>
<p>Swimlane</p> 	<p>Pemisahan organisasi bisnis yang bertanggung jawab atas kegiatan yang berlangsung.</p>


2.12.3 Sequence Diagram

Sequence diagram "menggambarkan perilaku objek dalam kasus penggunaan dengan mengartikan siklus hidup object dan pesan yang dibawa dan masuk di antara objek. Oleh sebab itu, hendak memvisualkan diagram urutan, perlu untuk memahami objek yang berperan dalam kasus penggunaan dan objek yang dipakai ke dalam metode objek kelas". Menghasilkan diagram sekuen juga dibutuhkan guna mengamati skenario yang ada pada use case menurut Rosa dan Shalahuddin dalam (Julianto & Setiawan, 2019). Berikut ini adalah simbol-simbol sequence diagram, seperti terlihat pada Tabel 2.3 di bawah ini:

Tabel 2. 3 Sequence diagram (Julianto & Setiawan, 2019).

Simbol	Penjelasan
<p>Aktor</p> 	<p>Proses, orang, ataupun sistem berbeda yang terhubung dengan sistem data yang sedang dibikin terletak di luar sistem informasi yang sedang dibikin itu sendiri, sehingga aktor belum tentu</p>

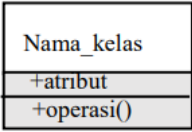




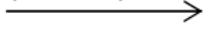

	orang meskipun simbol aktor adalah gambar seseorang; biasanya kata benda digunakan di depan frase nama aktor untuk mengekspresikan.
<p>Lifeline</p> 	Mewakili siklus hidup suatu objek
<p>Objek</p> <div style="border: 1px solid black; padding: 2px; width: fit-content;"> <p><u>Nama objek:</u> <u>nama kelas</u></p> </div>	Mendeklarasikan object yang berinteraksi dengan pesan.
<p>Waktu aktif</p> 	Mendeklarasikan bahwa objek itu aktif dan interaktif, dan segala sesuatu yang berhubungan dengan waktu aktif ini ialah langkah yang dibuat di dalamnya.
<p>Pesan tipe Create</p> 	Mendeklarasikan objek menciptakan objek lain, dan panah menunjuk ke objek yang dibikin. Arah panah menunjuk ke object yang memiliki operasi/metode , sebab ini memanggil operasi/metode , operasi/metode yang dipanggil harus ada dalam diagram kelas sesuai dengan kelas objek yang berinteraksi. Kelas objek yang berinteraksi.

<p>Pesan tipe Send l : masukan →</p>	<p>Merupakan object yang mengirimkan input /data / informasi ke object lain, dengan panah yang menunjuk ke objek pengirim.</p>
<p>Pesan tipe return l : ...keluaran→</p>	<p>Menunjukkan bahwa suatu objek kembali ke objek yang ditentukan setelah menjalankan operasi atau metode tertentu, dan panah menunjuk ke objek yang menerima pengembalian.</p>
<p>Pesan tipe destroy() </p>	<p>Deklarasikan objek untuk mengakhiri hidup objek lain, arah panah menunjuk ke objek yang diakhiri, yang terbaik adalah jika ada create maka ada destroy.</p>

2.12.4 Class Diagram

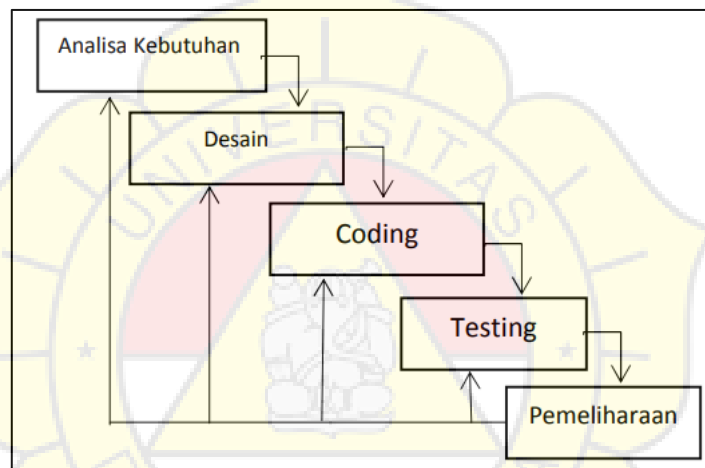
Diagram kelas ialah memvisualkan struktur sistem dengan mendefinisikan kelas-kelas yang membangunnya. Kelas mempunyai properti dan metode serta operasi yang dipanggil (Julianto & Setiawan, 2019). Berikut adalah notasi class diagram seperti pada Tabel 2.4 berikut ini:

Tabel 2. 4 Class diagram (Julianto & Setiawan, 2019).

Simbol	Deskripsi
<p>Kelas</p> 	Kelas bentuk sistem.
<p>Antar muka</p> 	Konsep yang sama dengan antarmuka dalam pemrograman berorientasi objek.
<p>Association</p> 	Hubungan antar kelas memiliki arti umum, dan asosiasi sering beserta dengan multiplisitas.
<p>Asosiasi berarah</p> 	Sebuah hubungan antar class memiliki implikasi bahwa sebuah class dipakai oleh class lain, dan asosiasi seringkali juga datang dengan multiplisitas.
<p>Generalisasi</p> 	Hubungan antar class memiliki implikasi generalisasi-generalisasi-spesialisasi (umumnya khusus).
<p>Kebergantungan / dependency</p> 	Hubungan antar kelas memiliki implikasi saling ketergantungan antar kelas.
<p>Agregasi/aggregation</p> 	Hubungan antar kelas memiliki makna keseluruhan-bagian.

2.13 Metode Pengembangan Sistem

Menurut (Julianto & Setiawan, 2019) , Pada perancangan system informasi ini dijalankan oleh sistem development life cycle (SDLC) dengan memakai proses waterfall. SDLC ialah kerangka kerja yang terstruktur secara berurutan untuk peningkatan sistem informasi (piranti lunak). 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. 2 Metode Waterfall

1. Analisa Kebutuhan

System pada sesi ini ialah sesi analisis terhadap kebutuhan- kebutuhan system yang dibutuhkan buat memudahkan proses pembuatan serta peningkatan sistem tersebut. Sesi ini mengambil analisis proses bisnis yang lagi bergerak, analisis permasalahan, sistem usulan, serta analisis kebutuhan fungsional.

2. Desain

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

3. Coding (Pengkodean)

Ialah sesi perwujudan sistem yang berawal dari integrasi antara desain sistem yang sudah terbuat dengan memakai bahasa pemrograman Visual Basic serta desain database memakai DBMS MySQL.

4. Testing (Pengujian Sistem)

Ialah sesi menguji terhadap sistem yang sudah dibentuk, dengan tujuan buat mengenali kelayakan dan kekurangan sistem tersebut. Hasil dari sesi ini dijadikan selaku bahan penilaian buat melaksanakan revisi serta akumulasi untuk sistem yang sudah dibentuk.

5. Pemeliharaan Sistem

ialah tahap terakhir dalam perancangan sistem (piranti lunak), dimana sistem yang sudah dibuat akan mengalami perubahan- perubahan dan penambahan sesuai dengan kebutuhan pengguna.



TEKNOLOGI INFORMASI
UNIVERSITAS DARMA PERSADA