BAB II
LANDASAN TEORI

2.1 Konsep Data

Pada subbab ini akan dibahas mengenai pengertian, jenis, dan sumber data.

2.1.1 Pengertian Data

Secara sederhana data didefinisikan sebagai kumpulan dari fakta berupa angka, simbol, gambar, ataupun tulisan yang diperoleh melalui pengamatan suatu objek. Data yang dikumpulkan kemudian diolah untuk menghasilkan informasi yang bermanfaat (Andretti Abdillah et al., 2021).

2.1.2 Jenis Data

1. **Data kualitatif**

Data kualitatif adalah data yang berbentuk kalimat, kata, atau gambar seperti teks, narasi, transkrip, dan lain-lain. Data kualitatif diperoleh melalui wawancara mendalam, observasi, dan analisis dokumen. (Andretti Abdillah et al., 2021).

2. **Data kuantitatif**

Data kuantitatif adalah data berupa angka hasil dari suatu pengukuran atau observasi. Data kuantitatif dapat dianalisis menggunakan metode statistik untuk memperoleh prediksi hubungan antar variabel, sehingga dapat ditampilkan dalam bentuk-bentuk statistik (Andretti Abdillah et al., 2021).

2.1.3 Sumber Data

1. **Data primer**

Data primer adalah data yang diperoleh dari tangan pertama, langsung dari sumber utamanya. Contoh sumber data primer adalah data hasil kuisioner terhadap responden, data hasil wawancara langsung, dan data hasil survei (Andretti Abdillah et al., 2021).

2. **Data sekunder**

Data sekunder adalah data yang diperoleh dalam bentuk informasi yang telah ada, dikumpulkan, dan diolah oleh pihak lain. Contohnya, laporan keuangan perusahaan, situs web, internet, dan lainnya (Andretti Abdillah et al., 2021).

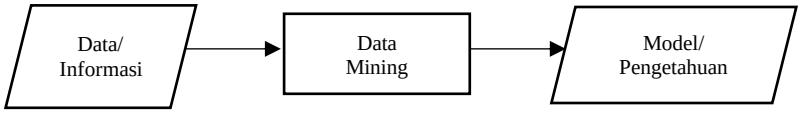
2.2 Konsep Data Mining

Pada subbab ini akan dibahas mengenai pengertian, karakteristik, tugas, pengelompokkan, dan tahapan-tahapan dalam data mining.

2.2.1 Pengertian Data Mining

Data mining merupakan proses penggalian informasi dan pola yang bermanfaat dari data yang sangat besar. Data mining mencakup pengumpulan data, ekstrasi data, analisis data, dan statistik data. Data mining bertujuan untuk menemukan pola yang sebelumnya tidak diketahui. Pola-pola yang telah diperoleh tersebut dapat digunakan untuk menyelesaikan berbagai macam permasalahan. (Arhami & Nasir, 2020).

Secara sederhana, data mining dapat digambarkan sebagai suatu pola, model, kaidah, atau pengetahuan yang dihasilkan dari data mining yang dapat dilihat pada gambar 2.1 berikut (Arhami & Nasir, 2020).



Gambar 2. 1 Model atau pengetahuan merupakan output data mining

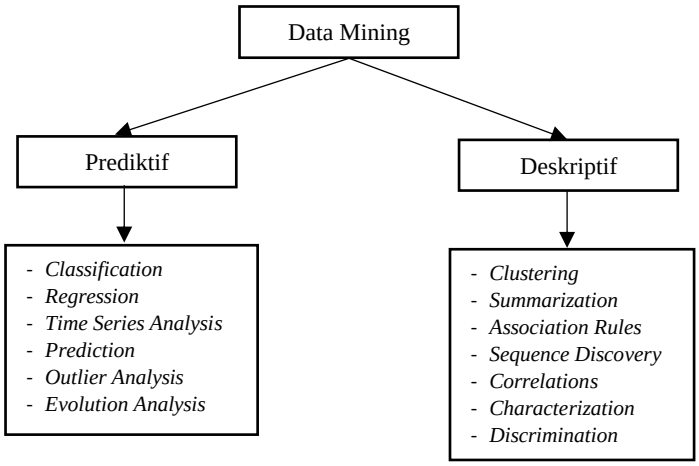
2.2.2 Karakteristik Data Mining

Data mining memiliki karakteristik tertentu. Karakteristik-karakteristik data mining, yaitu (Siregar & Pusphabuana, 2017).

1. Data mining berhubungan dengan penemuan sesuatu yang tersembunyi dan pola data tertentu yang tidak diketahui sebelumnya.
2. Data mining biasa menggunakan data yang sangat besar. Data yang besar biasanya digunakan untuk membuat hasil lebih dapat dipercaya.
3. Data mining berguna untuk membuat keputusan kritis.

2.2.3 Tugas Data Mining

Secara garis besar, tugas data mining dibagi menjadi dua kategori utama.



Gambar 2. 2 Tugas data mining

2.2.3.1 Tugas Prediksi

Tujuan tugas prediksi adalah untuk memprediksi nilai dari atribut tertentu berdasarkan nilai dari atribut lainnya. Metode-metode prediksi data mining, yaitu (Tan et al., 2006).

1. *Classification*: pembagian data ke dalam beberapa kelompok atau kelas yang telah ditentukan sebelumnya.
2. *Regression*: memahami dan mengukur bagaimana perubahan dalam satu atau lebih variabel prediktor dapat mempengaruhi variabel target.
3. *Time Series Analysis*: pengamatan perubahan nilai atribut dari waktu ke waktu.
4. *Prediction*: perkiraan tentang nilai atau perilaku di masa depan berdasarkan data historis yang telah ada.
5. *Outlier Analysis*: identifikasi dan penanganan data outliers dalam dataset.
6. *Evolution Analysis*: pemahaman perubahan atau evolusi dari data seiring waktu.

2.2.3.2 Tugas Deskripsi

Tujuan tugas ini adalah memperoleh pola (*correlation, trend, cluster, trajectory, anomaly*) untuk menyimpulkan hubungan di dalam data. Tugas deskripsi merupakan tugas yang sering dibutuhkan pada teknik *postprocessing* untuk melakukan validasi dan menjelaskan hasil dari proses data mining. Metode-metode deskripsi data mining, yaitu (Tan et al., 2006):

1. *Clustering*: pengelompokkan beberapa objek yang serupa ke dalam sebuah cluster dan yang tidak serupa ke *cluster* yang lain.
2. *Summarization*: pemetaan data ke dalam subset dengan deskripsi sederhana.
3. *Association rules*: identifikasi hubungan antara data yang satu dengan yang lainnya.
4. *Sequence discovery*: identifikasi pola sekuensial dalam data.
5. *Correlation*: mengidentifikasi hubungan linier antara dua variabel.
6. *Characterization*: menggambarkan dan merangkum karakteristik umum dari dataset
7. *Discrimination*: membandingkan dan mengidentifikasi perbedaan atau pola yang membedakan dua atau lebih kelompok atau subpopulasi dalam dataset.

2.2.4 Pengelompokkan Data Mining

Data mining dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu (Larose, 2005).

2.2.4.1 Deskripsi (*Description*)

Deskripsi bertujuan untuk mengidentifikasi pola yang muncul secara berulang dan mengubah pola tersebut menjadi aturan dan kriteria yang dapat dengan mudah di mengerti oleh para ahli pada domain aplikasinya. Aturan yang dihasilkan harus mudah dimengerti agar efektif meningkatkan tingkat pengetahuan pada sistem.

2.2.4.2 Prediksi (*Prediction*)

Prediksi hampir sama dengan klasifikasi dan estimasi, akan tetapi dalam prediksi dapat diklasifikasikan berdasarkan nilai yang diperkirakan pada masa mendatang. Contoh prediksi adalah memprediksi harga beras dalam tiga bulan yang akan datang dan memprediksi persentase kenaikan kecelakaan lalu lintas tahun depan jika batas bawah kecepatan dinaikkan.

2.2.4.3 Estimasi (*Estimation*)

Estimasi hampir sama dengan klasifikasi, namun untuk variabel target estimasi lebih ke arah numerik daripada ke arah kategori. Model yang dibnagun menggunakan record lengkap yang menyediakan nilai dari variabel target sebagai nilai prediksi. Selanjutnya, estimasi nilai dari variabel target dibuat berdasarkan nilai variabel prediksi. Contoh estimasi yaitu estimasi nilai indeks prestasi kumulatif mahasiswa program pascasarjana dengan melihat nilai indeks prestasi mahasiswa tersebut pada saat mengikuti program sarjana.

2.2.4.4 Klasifikasi (*Classification*)

Klasifikasi merupakan proses menemukan sebuah model atau fungsi yang mendeskripsikan dan membedakan data ke dalam kelas-kelas. Klasifikasi melibatkan proses pemeriksaan karakteristik dari objek dan memasukkan objek ke dalam salah satu kelas yang sudah didefinisikan sebelumnya.

2.2.4.5 Pengelompokkan (*Clustering*)

Pengklusteran merupakan pengelompokkan *record*, pengamatan, atau memperhatikan, dan membentuk kelas objek-objek yang memiliki kemiripan. Kluster adalah kumpulan record yang memiliki kemiripan satu dengan yang lainnya dan memiliki ketidakmiripan dengan *record-record* dalam kluster lain.

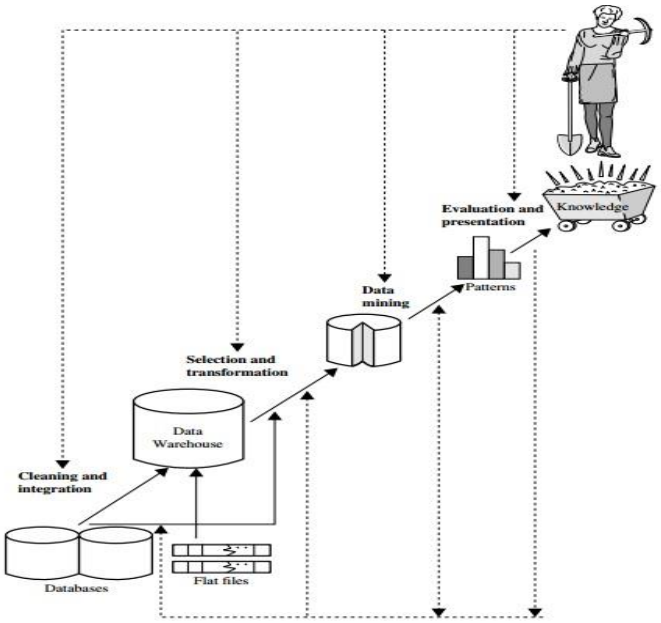
Pengklusteran berbeda dengan klasifikasi yaitu tidak adanya variabel target dalam pengklusteran. Algoritma pengklusteran melakukan pembagian terhadap keseluruhan data menjadi kelompok-kelompok yang memiliki kemiripan. Kemiripan *record* dalam satu kelompok akan bernilai maksimal, sedangkan kemiripan dengan *record* dalam kelompok lain akan bernilai minimal.

2.2.4.6 Asosiasi (*Association*)

Tugas asosiasi dalam data mining adalah menemukan atribut yang muncuL dalam satu waktu. Dalam dunia bisnis lebih umum disebut analisis keranjang belaja (*market basket analysis*). Contoh asosiasi adalah menemukan barang dalam supermarket yang dibeli secara bersamaan dan barang yang tidak pernah dibeli secara bersamaan.

2.2.5 Tahapan Data Mining

Sebagai suatu rangkaian proses, data mining dapat dibagi menjadi beberapa tahapan yang diilustrasikan pada Gambar 2.3.



Gambar 2. 3 Tahapan data mining

Tahapan-tahapan data mining adalah sebagai berikut (Suntoro, 2019).

2.2.5.1 *Data Cleaning* (Pembersihan Data)

Pembersihan data merupakan proses menghilangkan noise dan data yang tidak konsisten atau data tidak relevan. Proses pembersihan data juga mencakup antara lain membuang duplikasi data dan memperbaiki kesalahan pada data, seperti kesalahan cetak.

2.2.5.2 *Data Integration* (Integrasi Data)

Data yang digunakan untuk data mining tidak hanya berasal dari satu database tetapi juga berasal dari beberapa database atau file teks. Integrasi data dilakukan pada atribut-atribut yang mengidentifikasikan entitas-entitas yang unik seperti atribut nama, nim, IPK dan lain-lain.

2.2.5.3 *Data Selection* (Seleksi Data)

Kebutuhan data dari data yang diperoleh tidak semuanya akan dipakai dalam kebutuhan, untuk itu pada data yang diperoleh akan dilakukan penyeleksian data. Ini juga dilakukan untuk membuang data-data yang tidak relevan dan tidak valid yang akan mempengaruhi informasi berikutnya.

2.2.5.4 *Data Transformation* (Transformasi Data)

Data yang telah diproses pada tahap-tahap sebelumnya, kemudian dilakukan penggabungan dalam format data tertentu yang sesuai dengan data mining. Karena metode yang digunakan dalam proses mining membutuhkan format-format data tertentu agar supaya bisa diolah sedemikian rupa.

2.2.5.5 *Process Mining* (Proses Mining)

Proses mining merupakan proses yang paling utama pada saat penerapan metode tertentu yang digunakan dalam data mining, yang bertujuan untuk menentukan dan menggali informasi berharga yang terdapat dalam suatu data.

2.2.5.6 *Pattern Evaluation* (Evaluasi Pola)

Evaluasi pola dalam tahap data mining merupakan mengidentifikasi atau menganalisa pola-pola yang terdapat dalam data, kemudian pola-pola tersebut dilakukan evaluasi untuk menilai apakah hipotesa yang didapatkan bisa tercapai atau tidak.

2.2.5.7 *Knowledge Presentation (*Presentasi Pengetahuan*)*

Proses ini adalah bagaimana memformulasikan keputusan atau aksi dari hasil analisis yang didapatkan. Hasil dari presentasi pengetahuan ini juga melibatkan orang-orang yang tidak memahami data mining, karena pengetahuan yang bersifat umum.

2.3 Klasifikasi

Salah satu metode yang dapat dilakukan dengan data mining adalah pengklasifikasian. Klasifikasi pertama kali diterapkan pada bidang tanaman yang mengklasifikasi suatu spesies tertentu, seperti yang dilakukan oleh Carolus von Linne yang pertama kali mengklasifikasi spesies berdasarkan karakteristik fisik (Yuli Mardi, 2019).

Metode-metode atau model-model yang telah dikembangkan untuk menyelesaikan kasus klasifikasi antara lain (Yuli Mardi, 2019).

1. Pohon keputusan C4.5
2. *Naïve bayes*
3. Jaringan saraf tiruan
4. Analisis statistik
5. Algoritma genetik
6. *Rough sets*
7. *K-Nearest Neighbour*
8. Metode berbasis aturan
9. *Memory based reasoning*
10. *Support vector machine*

2.4 Pohon Keputusan (*Decision Tree*)

Pohon keputusan merupakan metode klasifikasi dan prediksi yang sangat kuat dan terkenal. Pohon keputusan berguna untuk mengeksplorasi data, menemukan hubungan tersembunyi antara sejumlah calon variabel input dengan sebuah variabel target. Proses pohon keputusan adalah mengubah bentuk data (tabel) menjadi model pohon, mengubah model pohon menjadi *rule,* dan menyederhanakan *rule* (Kusrini & Lutfi, 2009).

2.5 Algoritma C4.5

Algoritma C4.5 merupakan salah satu algoritma yang digunakan dalam pembentukan pohon keputusan (*decision tree).* Algoritma C4.5 menggunakan perhitungan *entropy* dan *gain* untuk pemilihan atribut menjadi *node*. Berikut adalah tahapan-tahapan pembentukan pohon keputusan (*decision tree*) menggunakan algoritma C4.5 (Lumban Gaol et al., 2021).

1. Mempersiapkan data *training*

Data *training* berasal dari data yang telah dikumpulkan sebelumnya dan telah dikelompokkan ke dalam kelas-kelas tertentu.

2. Menghitung nilai *entropy*

Entropy adalah suatu parameter untuk mengukur tingkat keberagaman (heterogenitas) dari kumpulan data. Jika nilai *entropy* semakin besar, maka tingkat keberagaman suatu kumpulan data semakin besar. Rumus *Entropy* dapat dilihat pada persamaan (1) berikut ini:

$$Entropy(S) = \sum_{i=1}^{n} - pi * log2pi \quad (1)$$

Keterangan:

S = himpunan kasus

n = jumlah partisi S

pi = proporsi Si terhadap S

3. Menghitung nilai *gain*

Gain adalah ukuran efektivitas suatu variabel dalam mengklasifikasikan data. *Gain* dari suatu variabel merupakan selisih antara nilai entropy total dengan entropy dari variabel tersebut. Rumus *gain* dapat dilihat pada persamaan (2) berikut ini:

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{|Si|}{|S|} * Entropy(Si) \quad (2)$$

Keterangan:

S = himpunan kasus

A = atribut

n = jumlah partisi atribut A

|Si| = jumlah kasus pada partisi ke − i

|S| = jumlah kasus dalam S

4. Nilai *gain* tertinggi sebagai *root*

Pemilihan akar (*root*) atribut dilakukan dengan menghitung nilai gain dari semua atribut. nilai gain tertinggi akan menjadi akar pertama.

5. Partisi data dengan gain tertinggi

Ulangi langkah kedua sampai semua *record* terpartisi dengan sempurna

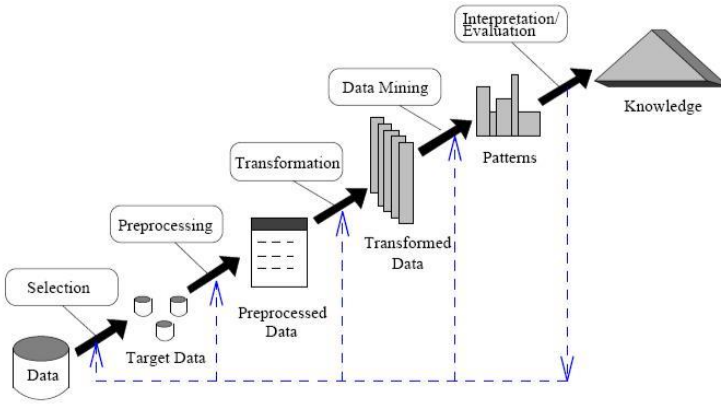
6. Proses partisi pohon keputusan

Proses partisi pohon keputusan akan berhenti jika:

a. Semua *record* pada simpul N mendapatkan kelas yang sama.
b. Tidak ada atribut di dalam *record* yang dipartisi lagi.
c. Tidak ada *record* di dalam cabang yang kosong.

2.6 Knowledge Discovery in Database (KDD)

Knowledge Discovery in Database (KDD) merupakan metode untuk memperoleh pengetahuan dari database yang ada. Dalam database terdapat tabel-tabel yang saling berhubungan atau berelasi. Hasil pengetahuan yang diperoleh dalam proses tersebut dapat digunakan sebagai basis pengetahuan (*knowledge base*) untuk keperluan pengambilan keputusan (Yuli Mardi, 2019).



Gambar 2. 4 Tahapan proses KDD

Istilah *Knowledge Discovery in Database* (KDD) dan data mining seringkali digunakan secara bergantian untuk menjelaskan proses penggalian informasi tersembunyi dalam suatu basis data yang besar. Sebenarnya kedua istilah tersebut memiliki konsep yang berbeda, tetapi berkaitan satu sama lain dan salah satu tahapan dalam keseluruhan proses KDD adalah data mining (Yuli Mardi, 2019).

Proses KDD secara garis besar dapat dijelaskan sebagai berikut.

2.6.1 *Data Selection* (Seleksi Data)

Pemilihan (seleksi) data perlu dilakukan sebelum tahap penggalian informasi dalam *Knowledge Discovery in Database* (KDD). Pada tahap ini, dilakukan pemilihan data yang akan digunakan dalam proses data mining. Data yang digunakan dalam penelitian ini berasal dari database kemahasiswaan jurusan sistem informasi.

2.6.2 *Pre-processing / Cleaning* (Pembersihan)

Sebelum proses data mining dilaksanakan, perlu dilakukan proses *cleaning* pada data. Proses *cleaning* mencakup antara lain membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data.

2.6.3 *Transformation* (Transformasi)

Proses transformasi merupakan proses untuk mengubah data dari bentuk asalnya menjadi data yang siap ditambang. Tahap ini mempermudah dalam proses penggalian data untuk menemukan suatu pengetahuan baru. Berikut contoh konsep dataset setelah melalui tahap transformasi dan akan melalui proses mining.

Tabel 2. 1 Contoh konsep dataset

Nama	NIM	JK	IPS 1	IPS 2	IPS 3	IPS 4	Status Mahasiswa (Aktif/Non-Aktif)
Dini Abshari	2020051 074017	P	3,45	3,56	3,61	3,70	Aktif
…	…	…	…	…	…	…	…
Reynaldi	2021051 074018	L	3,02	0,00	2,11	2,08	Non-Aktif

2.6.4 Data Mining

Pada tahap ini dilakukan pemilihan algoritma dan implementasi dari algoritma yang dipilih. Proses data mining ini melakukan pencarian pola atau informasi menarik dalam data terpilih dengan menerapkan *decision tree* algoritma C4.5.

2.6.5 *Interpretation/Evaluation* (Interpretasi/Evaluasi)

Pada tahap ini dilakukan interpretasi dari pola yang dihasilkan dengan menggunakan algoritma C4.5. Interpretasi mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesis yang ada sebelumnya.

2.7 Confusion Matrix

Confusion matrix adalah metode yang digunakan untuk mendefinisikan ukuran performansi dari algoritma klasifikasi. Sebuah matrix dari prediksi akan dibandingkan dengan kelas asli untuk mengukur kinerja model dalam memprediksi label yang benar dan salah (Dwi Untari et al., 2018). Berikut adalah tabel *confusion matrix*.

Tabel 2. 2 Confusion Matrix

Prediksi	Aktual	
	Negative	*Postive*
Negative	*True Negative (TN)*	*False Negative (FN)*
Positive	*False Positive (FP)*	*True Positive (TP)*

True Positive (TP) adalah jumlah record positif yang diklasifikasikan sebagai positif. *False Positive* (FP) adalah jumlah record negatif yang diklasifikasikan sebagai positif. *True Negative* (TN) adalah jumlah record negatif yang diklasifikasikan sebagai negatif. *False Negative* (FN) adalah jumlah record positif yang diklasifikasikan sebagai negatif.

Akurasi adalah metrik yang mengukur sejauh mana algoritma dapat memprediksi dengan benar kelas-kelas data. Rumus menghitung akurasi sebagai berikut.

$$Akurasi = \frac{TP + TN}{TP + TN + FN + FP} \times 100\% \quad (3)$$

2.8 Split Validation

Split validation adalah teknik validasi yang membagi data menjadi dua bagian yaitu data training dan data testing. Data training atau data latih adalah data yang akan dipakai dalam melakukan pembelajaran sedangkan data testing atau data uji adalah data yang belum pernah dipakai dalam pembelajaran dan akan berfungsi sebagai data untuk menguji kebenaran atau akurasi hasil pembelajaran dari data training (Dwi Untari et al., 2018).

2.9 RapidMiner

RapidMiner adalah platform perangkat lunak ilmu data yang pertama kali dikembangkan oleh Tim Wenzel pada tahun 2001. RapidMiner menyediakan lingkungan terintegrasi untuk persiapan data, pembelajaran mesin, pembelajaran dalam, penambangan teks, dan analisis prediktif. Hal ini digunakan untuk bisnis, komersial, penelitian, pendidikan, pelatihan, *rapid prototyping*, dan pengembangan aplikasi serta mendukung semua langkah dalam proses pembelajaran mesin termasuk persiapan data, hasil visualisasi, validasi model, dan optimasi (Siregar & Pusphabuana, 2017).

2.10 Web

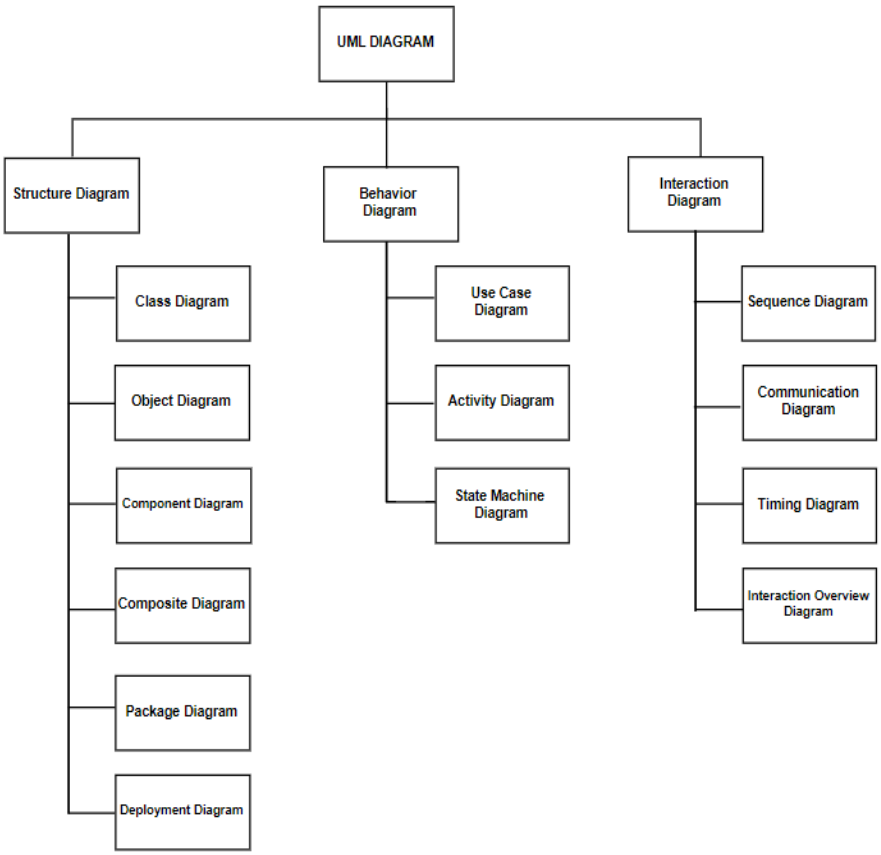
Website adalah kumpulan halaman web yang berfungsi untuk menampilkan berbagai informasi dalam bentuk tulisan, gambar, dan suara dari sebuah domain yang terbentuk dalam suatu rangkaian yang saling terkait. Suatu halaman web yang sudah terhubung dengan suatu halaman web lain biasanya disebut dengan hyperlink, sedangkan teks yang terhubung oleh teks lain disebut sebagai hypertext (Kinaswara et al., 2019).

2.11 UML

UML (*Unified Modeling Language*) adalah bahasa permodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek (Rosa & Shalahuddin, 2018) .

2.11.1 Jenis Diagram UML

UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar berikut.



Gambar 2. 5 Diagram UML

Berdasarkan gambar 2.14 berikut penjelasan singkat dari pembagian kategori tersebut:

a. *Structure diagram*: kumpulan diagram yang digunakan untuk menggambarkan strutur statis dari sistem yang dimodelkan.

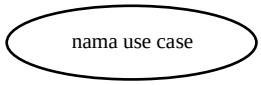
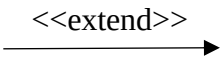
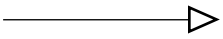
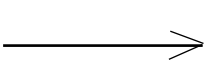
b. *Behavior diagram*: kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada suatu sistem.

c. *Interaction diagram*: kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

2.11.1.1 Use Case Diagram

Use case adalah rangkaian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. Use case digunakan untuk membentuk tingkah laku benda dalam sebuah model serta direalisasikan oleh sebuah kolaborasi (Rosa & Shalahuddin, 2018). Simbol yang digunakan pada use case diagram dapat dilihat pada Tabel 2.1 berikut:

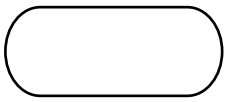
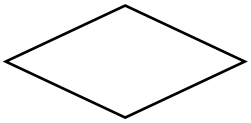
Tabel 2. 3 Diagram use case

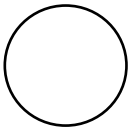
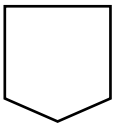
Simbol	Deskripsi
Use case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor
Aktor / *actor*  nama aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri
Asosiasi / *association*	Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor
Ekstensi / *extend* 	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu
Generalisasi / *generalization* 	Hubungan generalisasi dan spesialisasi antara dua buah use case
Include 	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini

2.12 Flowchart

Flowchart (bagan alir) merupakan suatu jenis diagram yang merepresentasikan algoritma atau langkah-langkah instruksi yang berurutan dalam sistem. Fungsi utama dari flowchart adalah memberi gambaran jalannya sebuah program dari satu proses ke proses lainnya (Rosa & Shalahuddin, 2018). Berikut simbol-simbol yang digunakan dalam flowchart:

Tabel 2. 4 Diagram flowchart

Simbol	Deskripsi
Terminator 	Simbol yang menyatakan awal atau akhir suatu program.
Flow	Simbol yang digunakan untuk menggabungkan antara simbol yang satu dengan simbol yang lainnya.
Decision 	Simbol yang menunjukkan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu ya atau tidak.
Input/*Output* 	Simbol yang menyatakan proses input atau output tanpa melihat jenisnya.
Process 	Simbol yang menyatakan suatu proses yang dilakukan komputer.
Predefined Process 	Simbol untuk pelaksanaan suatu bagian (sub-program) atau *prosedure*.

Connector 	Simbol untuk keluar-masuk atau penyambungan proses dalam lembar kerja yang sama.
Connector 	Simbol untuk keluar-masuk atau penyambungan proses dalam lembar kerja yang berbeda.

2.13 HTML

HTML (HyperText Markup Language) merupakan bahasa yang digunakan untuk mendeskripsikan struktur sebuah halaman web. Statement dasar dari HTML disebut tags. Sebuah tag dinyatakan dalam sebuah kurung siku (<>). Tags yang ditujukan untuk sebuah dokumen atau bagian dari suatu dokumen haruslah dibuat berupa pasangan yang terdiri dari tag pembuka dan tag penutup. Dimana tag penutup menggunakan tambahan tanda garis miring (/) di awal nama tag (Sari et al., 2022).

2.14 PHP

PHP (*Hypertext Preprocessor*) merupakan suatu bahasa pemograman yang difungsikan untuk membangun suatu website dinamis. PHP menyatu dengan kode HTML. HTML digunakan sebagai pembangun dan pondasi dari kerangka layout web, sedangkan PHP difungsikan sebagai prosesnya sehingga dengan adanya PHP sebuah web akan sangat mudah di *maintenance* (Welling & Thomson, 2003).

2.15 Codeigniter

Codeigniter merupakan sebuah *web framework* dengan model MVC (*Model, View, Controller*) untuk membangun aplikasi web menggunakan PHP. Codeigniter dirancang untuk menjadi sebuah *web framework* yang ringan dan mudah untuk digunakan. Dengan codeigniter, jumlah kode yang ditulis dapat diminimalisir dan mempermudah pengembangan proyek (Subagia, 2019).

2.16 Bootstrap

Bootstrap adalah framework CCS, HTML, dan JavaScript yang memiliki fungsi untuk mendesain sebuah web yang responsif. Bootstrap diciptakan untuk mempermudah proses desain web bagi berbagai tingkat pengguna, mulai dari level pemula hingga yang sudah berpengalaman (Rozi, 2015).

2.17 XAMPP

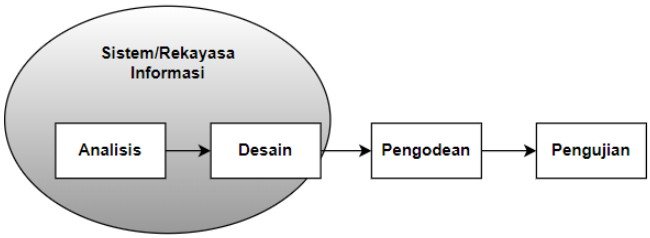
XAMPP merupakan web server yang bersifat *open source* dan merupakan gabungan dari Apache, MySQL, dan Filezila. Kelebihan dari XAMPP adalah dapat berjalan pada beberapa sistem operasi seperti windows, linux, dan lain-lain. XAMPP digunakan untuk membuat *server local* atau biasa disebut *localhost* (Fathansyah, 2012).

2.18 Metode Pengembangan Sistem

Metode pengembangan sistem merujuk pada serangkaian langkah dan pendekatan sistematis yang digunakan untuk merancang, mengembangkan, dan mengimplementasikan perangkat lunak. Tujuan dari metode pengembangan sistem adalah untuk mengelola proses pengembangan secara efisien dan memenuhi kebutuhan pengguna dan pemangku kepentingan. Beberapa metode pengembangan sistem yang umum digunakan adalah sebagai berikut (Rosa & Shalahuddin, 2018).

2.18.1 Model Waterfall

Model *waterfall* (air terjun) adalah model yang menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*). Berikut adalah gambar model air terjun (Rosa & Shalahuddin, 2018).



Gambar 2. 6 Model waterfall

Penjelasan tahapan-tahapan model *waterfall* adalah sebagai berikut. (Rosa & Shalahuddin, 2018).

1. **Analisis**

Analisis kebutuhan perangkat lunak adalah proses pengumpulan kebutuhan yang dilakukan secara intensif untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*.

2. **Desain**

Desain perangkat lunak adalah proses yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.

3. **Pengodean**

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. **Pengujian**

Pengujian fokus pada perangkat lunak secara dari segi lojik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5. **Pemeliharaan (*maintenance*)**

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke user. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian. Tahap pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.18.2 Model *Prototype*

Model prototype dimulai dari mengumpulkan kebutuhan pelanggan terhadap perangkat lunak yang akan dikembangkan. Selanjutnya, dibuatlah sebuah program prototipe yang bertujuan memberikan gambaran lebih jelas kepada pelanggan mengenai rancangan sebenarnya. Program prototipe biasanya menyediakan tampilan dengan simulasi alur perangkat lunak sehingga tampak seperti perangkat lunak yang sudah jadi. Berikut adalah gambar model *prototype* (Rosa & Shalahuddin, 2018).

2.18.3 Model RAD

Rapid Application Development (RAD) adalah model proses pengembangan perangkat lunak yang bersifat incremental terutama untuk waktu pengerjaan yang pendek. Model RAD adalah adaptasi dari model air terjun untuk pengembangan setiap komponen perangkat lunak (Rosa & Shalahuddin, 2018).

2.18.4 Model Iteratif

Model iteratif (iterative model) mengkombinasikan proses-proses pada model air terjun dan iteratif pada model prototipe. Model inkremental akan menghasilkan versi-versi perangkat lunak yang sudah mengalami penambahan fungsi untuk setiap pertambahannya (inkremen/increment) (Rosa & Shalahuddin, 2018).

2.18.5 Model Spiral

Model spiral (spiral model) memasangkan iteratif pada model prototipe dengan kotrol dan aspek sistematik yang diambil dari model air terjun. Model spiral menyediakan pengembangan dengan cara cepat dengan perangkat lunak yang memiliki versı yang terus bertambah fungsinya (increment) (Rosa & Shalahuddin, 2018).

2.19 Metode Pengujian Sistem

Metode pengujian sistem adalah proses sistematis untuk mengevaluasi fungsi-fungsi perangkat lunak guna memastikan bahwa perangkat lunak tersebut beroperasi sesuai dengan harapan. Tujuan utama pengujian perangkat lunak adalah untuk mengidentifikasi bug, memastikan keamanan, dan memastikan bahwa perangkat lunak berfungsi secara konsisten dan dapat diandalkan. Berikut dua jenis pendekatan utama pada pengujian perangkat lunak.

2.19.1 Pengujian *Black Box*

Pengujian *black box* yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan (Rosa & Shalahuddin, 2018).

Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus proses login maka kasus uji yang dibuat adalah.

1. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang benar
2. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah.

2.19.2 Pengujian *White Box*

Pengujian *white box* yaitu menguji perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi-fungsi, masukan, dan keluaran yang sesuai dengan spesifikasi kebutuhan. Pengujian kotak putih dilakukan dengan memeriksa lojik dari kode program. Pembuatan kasus uji bisa mengikuti standar pengujian dari standar pemrograman yang seharusnya. Contoh dari pengujian kotak putih misalkan menguji alur (dengan menelusuri) pengulangan (looping) pada logika pemrograman (Rosa & Shalahuddin, 2018).