

## BAB II

### LANDASAN TEORI

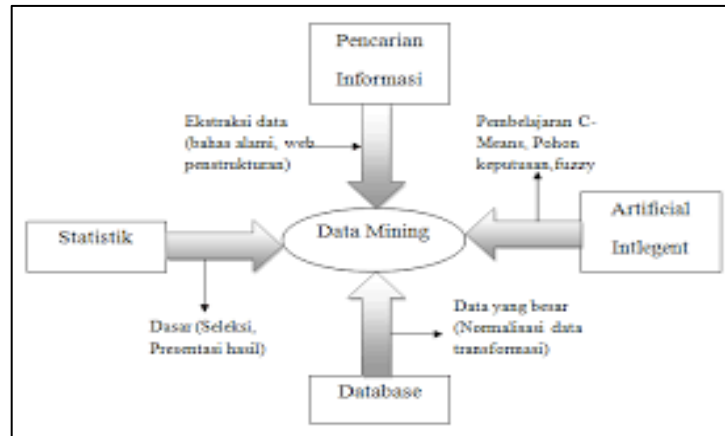
#### 2.1 Data Mining

(Kurniawan, 2018) menyatakan data mining adalah proses maupun tahapan dalam menemukan sebuah struktur data. Struktur data tersebut dapat mengambil banyak bentuk, termasuk aturan, grafik atau jaringan, pohon (*tree*) maupun persamaan, serta beberapa yang lain. Dengan menggunakan data mining, maka sebuah kasus dapat dilihat trend, struktur maupun prediksinya di masa mendatang. Data mining sendiri memiliki banyak tahapan dan teknik yang dapat diimplementasikan dalam kehidupan nyata

(Parteek Bhatia, 2019) Data mining adalah proses yang memperkerjakan satu atau lebih Teknik pembelajaran *computer (machine learning)* untuk menganalisa dan mengekstraksi pengetahuan (*knowledge*) secara otomatis.

*Data Mining* adalah serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual (Amalia, 2018).

*Data Mining* suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam *database*. *Data Mining* adalah proses yang menggunakan Teknik Statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai *database* besar (Amalia, 2018).

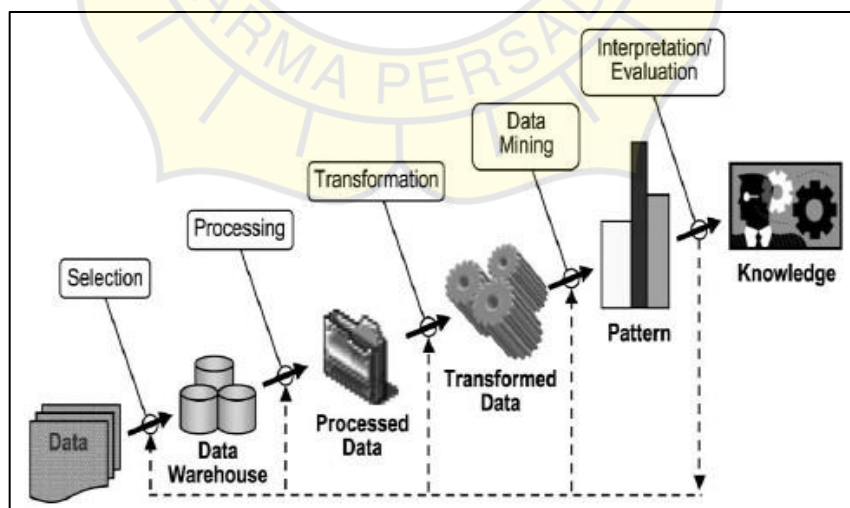


Gambar 2. 1 Bidang Ilmu Data Mining

(Sumber: Larose, 2006)

## 2.2 Tahapan Proses Dalam Data Mining

Ada beberapa proses tahap-tahapan yang ada di dalam data mining. Fase yang awal yaitu dimulai dari data-data sumber dan berakhir dengan adanya informasi yang dihasilkan dari beberapa tahapannya, yaitu:



Gambar 2. 2 Fase-fase Dalam Data Mining

(Sumber: Muchlisin Riadi, 2017)

Tahapan dalam Data Mining dapat diartikan sebagai berikut:

1. *Seleksi Data*

Pemilihan (seleksi) data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam KDD dimulai. Data hasil seleksi yang digunakan untuk proses data mining, disimpan dalam suatu berkas, terpisah dari basis data operasional.

2. *Pre-processin / cleaning*

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

3. *Transformation*

Coding adalah proses transformasi pada data yang telah dipilih, sehingga data tersebut sesuai untuk proses data mining. Proses coding dalam KDD merupakan proses kreatif dan sangat tergantung pada jenis atau pola informasi yang akan dicari dalam basis data.

4. *Data Mining*

*Data mining* adalah proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu. Teknik, metode, atau algoritma dalam data mining sangat bervariasi. Pemilihan metode atau algoritma yang tepat sangat bergantung pada tujuan dan proses KDD secara keseluruhan.

5. *Interpretation / Evaluation*

Pola informasi yang dihasilkan dari proses data mining perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini merupakan bagian dari proses KDD yang disebut *interpretation*. Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesis yang ada sebelumnya.

### 2.3 Tugas Data Mining

Menurut Anjar Wanto (2020: 6), Ada beberapa tugas yang dilakukan oleh *Data Mining* dalam memproses pemecahan masalah dan mencari pengetahuan baru, diantaranya adalah sebagai berikut:

a. **Klastering (*Clustering*)**

Digunakan untuk mengelompokan atau mengidentifikasi data yang memiliki karakteristik tertentu. Contoh algoritma: *K-Means*, *K-Medoids*, dan lainnya.

b. **Klasifikasi (*Classification*)**

Klasifikasi berguna untuk menemukan model/fungsi yang menjelaskan atau membedakan konsep dan juga kelas data, yang bertujuan untuk memperkirakan kelas dari suatu objek yang labelnya tidak diketahui. Contoh algoritmanya adalah: *C4.5*, *K-Nearest Neighbor (KNN)*, dan lain-lain.

c. **Asosiasi (*Association*)**

Asosiasi berguna mengatasi masalah bisnis yang tidak umum, yaitu dengan cara menganalisa tabel transaksi penjualan dan mengidentifikasi produk-produk yang sering dibeli bersamaan oleh pembeli, contohnya bila orang membeli sambal biasanya juga membeli kecap. Contoh algoritmanya adalah:

*Apriori*, *Frequent Pattern Growth* atau *FP-Growth* yang dipakai di dalam penulisan ini.

d. Estimasi (*Estimation*)

Estimasi berguna untuk memperkirakan sesuatu hal yang belum pernah ada pada sebelumnya, dan disajikan dalam bentuk hasil kuantitatif atau angka. Contoh algoritmanya adalah: *Regresi Linier*, *Confidence Interval Estimations*, dan lainnya.

e. Prediksi (*Prediction*)

Prediksi berguna untuk memperkirakan atau meramalkan suatu kejadian yang belum pernah terjadi. Contoh algoritmanya adalah: *Decision Tree*, *K-Nearest Neighbor* dan lainnya.

## 2.4 Metode Data Mining

Data mining model dibuat berdasarkan salah satu dari dua jenis pembelajaran *supervised* dan *unsupervised*. Fungsi pembelajaran *Supervised* digunakan untuk memprediksi suatu nilai. Fungsi pembelajaran *Unsupervised* digunakan untuk mencari struktur intrinsik, relasi dalam suatu data yang tidak memerlukan class atau label sebelum dilakukan proses pembelajaran.

Contoh dari algoritma pembelajaran *unsupervised*, diantaranya *K-Means Clustering* dan *Apriori Association Rules*. Contoh dari algoritma pembelajaran *supervised* yaitu *Naïve Bayes* untuk klasifikasi.

Metode data mining dapat diklasifikasikan berdasarkan fungsi yang dilakukan atau berdasarkan jenis aplikasi yang menggunakannya:

- Klasifikasi (*supervised*)
- Clustering (*unsupervised*)
- Association Rules (*unsupervised*)
- Attribute Importance (*supervised*)

#### **2.4.1 Klasifikasi (*supervised*)**

Pada persoalan klasifikasi, kita memiliki sejumlah kasus (sampel data) dan ingin memprediksi beberapa *class* yang ada pada sampel data tersebut. Tiap instan data berisi banyak atribut, dimana masing-masing atribut memiliki satu dari beberapa kemungkinan nilai.

Hanya satu atribut diantara banyak atribut tersebut yang disebut dengan atribut target, sedangkan atribut yang lain disebut sebagai atribut prediktor. Tiap kemungkinan nilai yang dimiliki oleh atribut target menunjukkan *class* yang diprediksi berdasarkan nilai-nilai dari atribut prediktor

Klasifikasi digunakan untuk segmentasi customer, pemodelan bisnis, analisa kartu kredit, dan banyak aplikasi yang lain. Sebagai contoh, perusahaan kartu kredit ingin memprediksi customer berdasarkan tipe pembayaran.

#### **2.4.2 Clustering (*unsupervised*)**

Clustering adalah teknik yang berguna untuk mengeksplorasi data. Digunakan pada saat banyak kasus dan tidak memiliki pengelompokan secara alami. Dalam hal ini algoritma data mining dapat digunakan untuk mencari pengelompokan yang ada pada data..

Analisa *Clustering* mengidentifikasi cluster yang ada pada data. Cluster adalah kumpulan obyek data yang mirip satu sama lain. Metode clustering yang bagus menghasilkan cluster yang berkualitas untuk memastikan kesamaan pada data yang ada dalam satu cluster. *Clustering* model berbeda dari model prediktif dikarenakan pada clustering tidak perlu ada atribut target. *Clustering* yang diorganisasi ke dalam struktur hirarkikal akan mendefinisikan taksonomi dari data. Dalam ODM, suatu cluster dikarakterisasi oleh centroid, attribute histograms, dan clustering model hierarchical tree. ODM membentuk hierarchical clustering dengan menggunakan versi perbaikan dari algoritma *K-Means* dan *O-Cluster*.

#### **2.4.3 Association Rules (*unsupervised*)**

Fungsi Association Rules seringkali disebut dengan "market basket analysis", yang digunakan untuk menemukan relasi atau korelasi diantara himpunan item-item. Fungsi ini paling banyak digunakan untuk menganalisa data dalam rangka keperluan strategi pemasaran, desain katalog, dan proses pembuatan keputusan bisnis.

Tipe Association rule bisa dinyatakan sebagai misal : "70% dari orang-orang yang membeli mie, juice dan saus akan membeli juga roti tawar". Aturan asosiasi mengcapture item atau kejadian dalam data berukuran besar yang berisi data transaksi. Dengan kemajuan teknologi, data penjualan dapat disimpan dalam jumlah besar yang disebut dengan "basket data." Aturan asosiasi yang didefinisikan pada basket data, digunakan untuk keperluan promosi, desain katalog, segmentasi customer dan target pemasaran.

#### **2.4.4 Attribute Importance (*supervised*)**

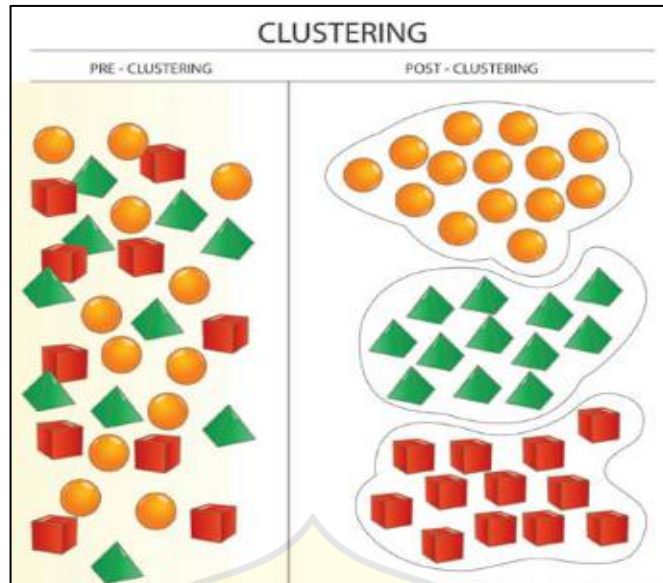
*Attribute Importance*, disebut juga dengan feature selection, menyediakan solusi otomatis untuk meningkatkan kecepatan dan akurasi dari model klasifikasi yang dibangun pada table data yang memiliki jumlah atribut yang sangat banyak. *Attribute Importance* meranking atribut *prediktif* dengan melakukan eliminasi nilai yang redundant, tidak relevant atau tidak informative dan mengidentifikasi atribut predictor yang banyak paling berpengaruh dalam pengambilan keputusan.

Dengan menggunakan atribut yang lebih sedikit akan mereduksi waktu untuk membangun suatu model, juga dapat meningkatkan akurasi dari kemampuan prediksi. Jika terlalu banyak atribut yang dilibatkan maka akan banyak pula noise yang terlibat yang akan berpengaruh terhadap model karena dapat menurunkan performansi dan akurasi.

#### **2.5 Teknik *Clustering***

Teknik *Clustering* yaitu teknik yang membagi data menjadi beberapa kelompok (Kusuma dan Andriani, 2020). Pembagian kelompok data disebut sebuah proses untuk mengelompokkan data ke dalam beberapa *cluster* sehingga cluster memiliki tingkat kemiripan yang sangat maksimum dan data antar cluster memiliki kemiripan yang minimum.





Gambar 2. 3 Penggambaran Teknik *Clustering*

(Sumber: Rizka Yolanda, 2019)

Pada Gambar 2.3 Mengilustrasikan data yang belum dikelompokan dan sudah dikelompokan yang berdasarkan bentuk dan kriterianya. *Cluster* yang baik ketika jarak antara objek data yang ada pada cluster yang sama dapat diminimalkan dan jarak antara objek yang ada pada cluster yang berbeda dapat dimaksimalkan.

## 2.6 Website

### 2.6.1 CSS (*Cascading Style Sheets*)

*CSS (Cascading Style Sheets)* adalah salah satu istilah teknis dalam pemrograman, *CSS* merupakan *styling language* (Bahasa desain), bagian dari markup language yang dapat mewarnai atau mendesain suatu halaman *website* (Khairina F. Hidayati, 2021).

Menurut (Wahyudi, 2017), CSS adalah suatu bahasa pemrograman *website* yang digunakan untuk mengendalikan dan membangun berbagai komponen dalam *website* sehingga tampilan akan lebih rapi, terstruktur dan seragam.

### **2.6.2 Javascript**

Menurut (Siahaan dan Rismon, 2020), yaitu "Javascript adalah sebuah bahasa script dinamis yang dapat dipakai untuk membangun interaktifitas pada halaman-halaman *HTML* statis. Ini dilakukan dengan menamakan blok-blok kode *Javascript* di semua tempat pada halaman *website*." *PHP (Hypertext Preprocessor)*.

### **2.6.3 PHP**

Rohi Abdulloh (2018, h. 3) mengatakan bahwa "*PHP* berperan sebagai proses data pada sisi *server*, sesuai yang diminta oleh *client* menjadi informasi yang siap ditampilkan, juga sebagai penghubung aplikasi *website* dengan *database*."

### **2.6.4 Bootstrap**

Zaenal A Rozi & SmitDev Community (2015, h. 1) mendiskusikan topik *bootstrap*. Dan mereka mengatakan bahwa "*Bootstrap* adalah paket aplikasi siap pakai untuk membuat *front-end* sebuah *website*. Bisa dikatakan, *bootstrap* adalah template desain *website* dengan fitur plus. *Bootstrap* diciptakan untuk mempermudah proses desain *website* bagi berbagai tingkat pengguna, mulai dari level pemula hingga yang sudah berpengalaman."

### **2.6.5 HTML (*Hypertext Markup Language*)**

Rohi Abdulloh (2018, h. 7) mengatakan bahwa “Bahasa standar *website* yang dikelola penggunaannya oleh W3C (*World Wide Website Consortium*) berupa tag-tag yang menyusun setiap elemen dari *website* HTML. Berperan sebagai penyusun struktur halaman *website* yang menempatkan setiap elemen *website* sesuai *layout* yang diinginkan.”.

### **2.6.6 Visual Studio Code (*VS Code*)**

Ummy Gusti Salamah (2021, h. 1) mendefinisikan bahwa “*Visual Studio Code (VS Code)* ini adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi multiplatform, artinya tersedia juga untuk Linux, Mac, dan Windows. Teks editor *VSCode* juga bersifat open source, yang mana kode sumbernya dapat kalian lihat dan kalian dapat berkontribusi untuk pengembangannya. Hal ini juga yang membuat *VSCode* menjadi favorit para pengembang aplikasi, karena para pengembang aplikasi bisa ikut serta dalam proses pengembangan *VSCode* ke depannya.

## **2.7 Database**

Menurut (Hesananda et al., 2017), *Database* adalah suatu wadah untuk menampung sebuah data yang ada pada sistem. *Database* juga bisa diartikan sebagai kumpulan data. *Database* juga bisa dikenal sebagai formal dan tegas. *Database* juga bisa diartikan dengan kumpulan data yang terintegrasi yang dapat dimanipulasi, diambil dan dicari secara cepat.

## 2.8 MySql

Yoga Ananda Putra, Sumijan, & Mardison. (2019) mendiskusikan bahwa “*MySQL* adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan *MySQL*, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial.

*MySQL* sebenarnya merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya; *SQL* (*Structured Query Language*). *SQL* adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

## 2.9 XAMPP

Darman Umagapi & Arisandy Ambarita (2018) mendiskusikan bahwa “XAMPP adalah perangkat lunak (*free software*) bebas, yang mendukung untuk banyak sistem operasi, yang merupakan kumpulan dari beberapa program.”.

## 2.10 Algoritma Sistem

### 2.10.1 Algoritma *Naïve Bayes*

*Naïve Bayes* adalah suatu klasifikasi kemungkinan sederhana yang dapat menghitung seluruh kemungkinan dengan menggabungkan sejumlah kombinasi dan

frekuensi suatu nilai dari basis data yang didapatkan. Suatu algoritma memanfaatkan teorema *bayes* dan memperkirakan seluruh atribut yang bebas dan saling lepas yang dapat diberikan oleh suatu nilai pada kelas variabel.

(Hayuningtyas, 2019) Metode *Naïve Bayes* merupakan metode yang memanfaatkan metode probabilitas dan statistic yang dikemukakan oleh ilmuwan Inggris *Thomas Bayes*. *Naïve Bayes* merupakan metode pengklasifikasikan yang sangat sederhana dengan mengasumsikan klasifikasi atribut. Dengan metode *Naïve Bayes* terlebih dahulu mencari Nilai Probabilitas dan likelihood maksimum dari setiap atribut untuk masing-masing kelas.

$$P(H\setminus X) = \frac{P(X\setminus H).P(H)}{P(X)}$$

Dimana:

X = Kelas data yang belum diketahui

H = Hipotesa data X adalah kelas spesifik

P(H|X) = Kemungkinan Hipotesa H berdasarkan keadaan X (posteriori prob.)

P(H) = Kemungkinan Hipotesa H (prior prob.)

P(X|H) = Kemungkinan X berdasarkan keadaan tersebut

P(X) = Kemungkinan dari X

### 2.10.2 Keuntungan dan Kerugian *Naïve Bayes*

Kerugian dari Metode Bayes termasuk bahwa Metode Bayes hanya dapat digunakan untuk masalah klasifikasi dengan pembelajaran yang diawasi dan data

kategorikal, Metode Bayes membutuhkan pengetahuan awal untuk dapat membuat keputusan. Tingkat keberhasilan metode ini tergantung pada pengetahuan awal yang diberikan.

Masalah lain dengan klasifikasi Naïve Bayes adalah 'frekuensi nol' yang berarti bahwa jika variabel kategori memiliki kategori tetapi tidak diamati dalam data pelatihan set, maka model Naive Bayes akan menetapkan probabilitas nol untuk itu dan itu tidak akan dapat untuk membuat prediksi.

Kelebihan dari Metode Bayes termasuk adalah Interpolasi: Metode Bayes memiliki pilihan tentang berapa banyak waktu dan upaya yang dilakukan oleh manusia vs komputer; Bahasa: Metode Bayes memiliki bahasa sendiri untuk menentukan hal-hal sebelumnya dan posterior; Intuisi: Melibatkan sebelum dan integrasi, dua kegiatan yang secara luas bermanfaat.

### **2.10.3 Algoritma *Decision Tree***

*Decision tree model is a method most frequently used in data mining. The purpose is to create a model that predicts the resultant of a target variable based on several input variables given by the user as training data set. (Suneetha, KR Krishnamoorthi, R 2011)*

*Decision Tree* merupakan algoritma yang umum digunakan untuk pengambilan keputusan dengan membentuk cabang-cabang dari setiap keputusan. (Sartika et al. 2017)

Menurut Anief Rufianto, M. Rochcham dan Abdul Rohman dalam buku yang berjudul “Penerapan Algoritma C4.5 Untuk Prediksi Kepuasan Mahasiswa

Tahun 2020” mendiskusikan bahwa “*Decision tree* atau pohon keputusan adalah metode atau algoritma klasifikasi data mining dengan membentuk pola pohon keputusan yang digunakan untuk mendapatkan jawaban dari masalah yang dimasukan. Dengan pohon keputusan, dapat dengan mudah mengidentifikasi hubungan antar faktor-faktor yang mempengaruhi masalah dan mencari solusi yang baik dengan memperhitungkan faktor-faktor tersebut.”.

#### **2.10.4 Algoritma C.45**

Menurut N. Iriadi dajnN. Nuraeni (2016), Algoritma *C4.5* adalah bagian dari algoritma untuk klasifikasi dalam pembelajaran machine learning dan data mining. *C4.5* merupakan algoritma yang cocok digunakan untuk masalah klasifikasi pada machine learning dan data mining.

Menurut Damanik, S. F., Wanto, A., & Gunawan, I. (2022), Algoritma *C4.5* adalah salah satu metode klasifikasi dari *data mining* yang digunakan untuk mengkonstruksikan pohon keputusan (*Decision Tree*). Algoritma *C4.5* adalah program yang memberi kontribusi satu set data berlabel dan menghasilkan pohon keputusan sebagai keluaran. Pohon keputusan tindak lanjut ini kemudian diverifikasi terhadap data uji berlabel yang tidak terlihat untuk menghitung generalisasinya.

*C4.5* adalah program yang digunakan untuk menghasilkan peraturan taksonomi dengan menggunakan pohon keputusan dari sekumpulan data yang diberikan.

Algoritma *C4.5* merupakan perpanjangan dari algoritma ID3 dasar dan dirancang oleh Quinlan. *C4.5* adalah salah satu algoritma pembelajaran yang



banyak digunakan. Algoritma *C4.5* membangun pohon keputusan dari serangkaian data pelatihan yang mirip dengan Algoritma ID3, dengan menggunakan konsep entropi informasi. *C4.5* juga dikenal sebagai klasifikasi statistik.

Algoritma *C4.5* merupakan metode yang menjadi pilihan pertama dan sering digunakan dalam pengembangan *Data Mining* karena kecepatan dalam pengklasifikasian pohon keputusan disamping dapat mengkonstruksi pengklasifikasian dengan aturan-aturan yang lain. Algoritma ini mempunyai *input* berupa *training samples* dan *samples*. *Training samples* berupa data contoh yang akan digunakan untuk membangun sebuah tree yang telah diuji kebenarannya. Sedangkan *samples* merupakan *field-field* data yang nantinya akan digunakan sebagai parameter dalam melakukan klasifikasi data. Secara umum algoritma *C4.5* untuk membangun pohon keputusan adalah sebagai berikut.

- a. Pilih atribut sebagai akar.
- b. Buat cabang untuk tiap-tiap atribut.
- c. Bagi kasus dalam cabang.
- d. Ulangi proses untuk setiap cabang sampai semua kasus pada cabang memiliki kasus yang sama.

Untuk memilih atribut sebagai akar, didasarkan pada nilai *gain* tertinggi dari atribut-atribut yang ada. Untuk menghitung *gain* digunakan rumus seperti tertera pada persamaan 1 berikut.

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} Entropy(S_i) \dots (1)$$

Keterangan:



- S : Himpunan kasus
- A : Atribut
- N : Jumlah partisi atribut A
- |Si| : Jumlah kasus pada partisi ke-i
- S : Jumlah kasus dalam S

Entropi(S) merupakan jumlah bit yang diperkirakan dibutuhkan untuk dapat mengekstrak suatu kelas dari sejumlah data acak pada ruang sampel S. Entropy dapat dikatakan sebagai kebutuhan bit untuk menyatakan suatu kelas. Pada persamaan (2) merupakan rumus entropi :

$$Entropy(S) = \sum_{i=1}^n - p_i * \log_2 p_i \dots (2)$$

Dimana:

- S : Ruang (data) sampel yang digunakan untuk pelatihan.
- n : Jumlah partisi S
- pi : Proporsi dari Si terhadap S

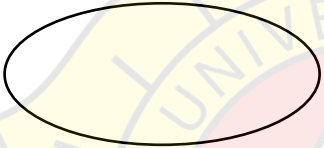


## 2.11 UML

Menurut Sulianta (2017) dalam buku “Teknik Perancangan Arsitektur Sistem Informasi” mengatakan bahwa “*Unified Modeling Language (UML)* merupakan kumpulan diagram-diagram yang sudah memiliki standar untuk membangun perangkat lunak berbasis objek.”.

Menurut Sukamto dan Shalahuddin (2018:133), mendiskusikan bahwa, “UML merupakan sebuah standar bahasa yang digunakan untuk menganalisis dan merancang serta menggambarkan arsitektur program dalam pemrograman *object oriented*.”.

### 2.11.1 Use Case

Table 2. 2 Simbol *Use Case* Diagram (Yunahar, 2018)

Simbol	Deskripsi
Use Case 	Fungsional yang disediakan system sebagai unit-unit yang saling bertukar antar unit atau actor biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case.
Actor 	Orang proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari actor adalah gambar orang, biasanya dinyatakan menggunakan kata benda di awal frase nama actor.
Asosiasi 	Komunikasi antar aktor dan use case yang berpartisipasi pada use case atau

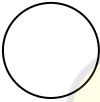



	use case memiliki interaksi dengan aktor.
<p>Extensi</p> <p>--- &lt;&lt;extend&gt;&gt; ---&gt;</p>	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan.
<p>Generalisasi</p> <p>—————&gt;</p>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya.
<p>Include</p> <p>--- &lt;&lt;include&gt;&gt; ---&gt;</p>	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.

*Use Case Diagram* merupakan diagram yang bekerja dengan cara mendeskripsikan tipikal interaksi antara user (pengguna) sebuah sistem dengan suatu sistem tersendiri melalui sebuah cerita bagaimana sebuah sistem dipakai (T. Bayu Kurniawan dan Syarifuddin, 2020).

*Use case* diagram terdiri dari aktor dan interaksi yang dilakukannya, aktor tersebut dapat berupa manusia, perangkat keras, system lain, atau mereka yang berinteraksi dengan sistem.

### 2.11.2 Activity Diagram

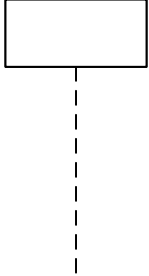


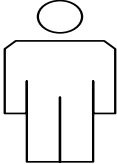
Table 2.3 Simbol *Activity* Diagram (Yunahar, 2018)

Simbol	Deskripsi
	Simbol start untuk menyatakan awal dari suatu proses
	Simbol stop untuk menyatakan akhir dari suatu proses
	Simbol desticion digunakan untuk menyatakan kondisi dari suatu proses
	Simbol action menyatakan aksi yang dilakukan dalam suatu arsitektur sistem

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

### 2.11.3 Sequence Diagram

Table 2. 4 Simbol *Sequence* Diagram (Yunahar, 2018)

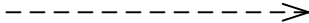
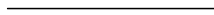
Nama Komponen	Keterangan	Simbol
Lifeline	Mengindikasikan keberadaan sebuah objek dalam basis waktu. Notasi untuk lifeline adalah garis putus-putus vertical yang ditarik dari sebuah objek	
Activation	Dinotasikan sebagai sebuah kotak segi empat Digambar pada sebuah lifeline mengidentifikasi sebuah objek yang akan melakukan sebuah aksi	
Message	Digambarkan dengan anak panah horizontal antara activation. Message mengidekasikan komunikasi antara objek-objek.	
Actor	Actor juga dapat berkomunikasi	

“*Sequence Diagram* adalah tool yang sangat populer dalam pengembangan sistem informasi secara object-oriented untuk menampilkan interaksi antar objek. Berdasarkan definisi tersebut, dapat disimpulkan bahwa *Sequence Diagram* adalah tool yang digunakan dalam pengembangan system” (Yunahar, 2018).

#### 2.11.4 *Deployment Diagram*

Table 2. 5 Simbol *Deployment Diagram* (Rosa dan M.Shalahuddin, 2018)

Simbol	Deskripsi
<p data-bbox="512 837 624 869">Package</p> 	<p data-bbox="847 837 1353 943">Package merupakan sebuah bungkus dari satu atau lebih node</p>
<p data-bbox="528 1160 608 1191">Node</p> 	<p data-bbox="847 1160 1353 1780">Biasanya mangacu pada perangkat keras (Hardware), perangkat lunak yang tidak dibuat sendiri (Software), jika di dalam node disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen</p>

Dependency 	Kebergantungan antar node, arah panah mengarah pada node yang dipakai
Link 	Relasi antar node

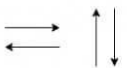











(Rosa dan M.Shalahuddin, 2018) mengemukakan bahwa “*Deployment Diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi”. *Deployment Diagram* dapat digunakan untuk menggambarkan hal-hal sebagai berikut, yaitu:

1. Sistem tambahan (embedded system) yang menggambarkan rancangan device, node dan hardware.
2. Sistem *client* atau *server*.
3. Sistem terdistribusi murni.
4. Rekayasa ulang aplikasi.

Dapat disimpulkan bahwa *Deployment Diagram* merupakan diagram yang digunakan untuk menggambarkan konfigurasi pada *system*.

### 2.11.5 *Flowchart*

Menurut Wibawanto (2017:20) “*Flowchart* adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program.”

	<p><b>Flow</b></p> <p>Simbol yang digunakan untuk menggabungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga dengan Connecting Line.</p>		<p><b>Input/output</b></p> <p>Simbol yang menyatakan proses input atau output tanpa tergantung peralatan.</p>
	<p><b>On-Page Reference</b></p> <p>Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang sama.</p>		<p><b>Manual Operation</b></p> <p>Simbol yang menyatakan suatu proses yang tidak dilakukan oleh komputer.</p>
	<p><b>Off-Page Reference</b></p> <p>Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang berbeda.</p>		<p><b>Document</b></p> <p>Simbol yang menyatakan bahwa input berasal dari dokumen dalam bentuk fisik, atau output yang perlu dicetak.</p>
	<p><b>Terminator</b></p> <p>Simbol yang menyatakan awal atau akhir suatu program.</p>		<p><b>Predefine Proses</b></p> <p>Simbol untuk pelaksanaan suatu bagian (sub-program) atau prosedur.</p>
	<p><b>Process</b></p> <p>Simbol yang menyatakan suatu proses yang dilakukan komputer.</p>		<p><b>Display</b></p> <p>Simbol yang menyatakan peralatan output yang digunakan.</p>
	<p><b>Decision</b></p> <p>Simbol yang menunjukan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu ya dan tidak.</p>		<p><b>Preparation</b></p> <p>Simbol yang menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal.</p>

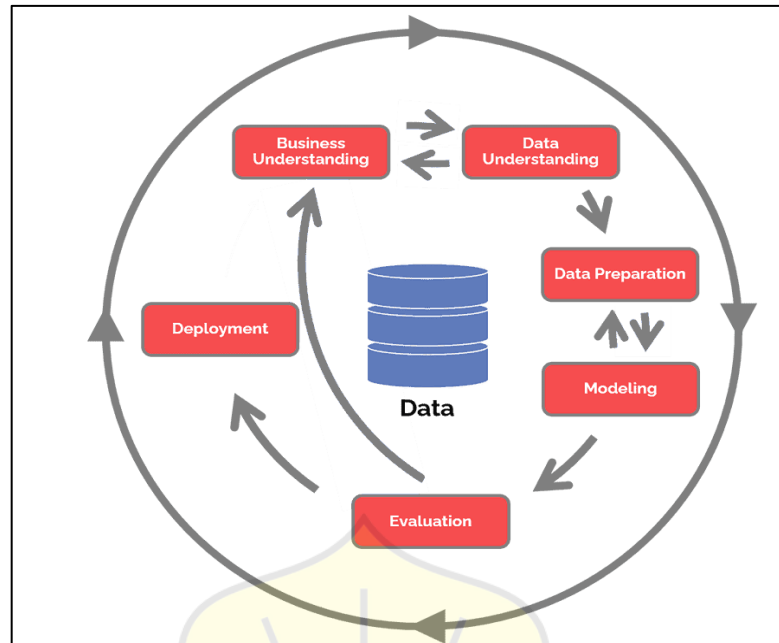
Gambar 2. 4 Simbol-simbol Flowchart

(Sumber: Rony Setiawan, 2021)

## 2.12 CRISP-DM

Menurut Bernardus Ari Kuncoro (2020) pada buku yang berjudul “Pengenalan Prinsip *Data Science* untuk Pemula” mendefinisikan bahwa “CRISP-DM merupakan singkatan dari *Cross-Industry Standard Process for Data Mining*. Sebuah metodologi yang menerapkan pendekatan terstruktur untuk perencanaan proyek data mining yang sangat ampuh dan sudah teruji dengan baik. Metode ini sangat umum digunakan karena sangat praktis, fleksibel, dan aplikatif untuk memecahkan isu bisnis yang sulit sekalipun. Metode ini merupakan metode andalan yang dapat dijalankan di hampir semua persoalan bisnis data mining.”.





Gambar 2. 5 Proses CRISP-DM

(Sumber: Bernardus Ari Kuncoro, 2020)

Metode CRISP-DM terdiri dari enam (6) tahapan yaitu: \*

1. Pemahaman Bisnis (*Business Understanding*)

Tahapan ini memfokuskan pada pemahaman tujuan proyek dan kebutuhan-kebutuhan yang diinginkan bisnis, kemudian merubahnya pengetahuan ini untuk mendefinisikan data mining dan rencana yang ingin dilakukan untuk mencapai tujuan bisnis.

2. Pemahaman Data (*Data Understanding*)

Tahapan memahami data dimulai dengan mengumpulkan data awal dan dilanjutkan dengan kegiatan-kegiatan untuk mendapatkan data yang lazim serta identifikasi data yang berkualitas, pemahaman data sangat diperlukan untuk mendeteksi bagian yang menarik dari data sehingga dapat membangun hipotesis terhadap informasi yang tersembunyi.

### 3. Persiapan Data (*Data Preparation*)

Fase *data preparation* ini memerlukan pemikiran matang dan upaya tinggi untuk memperbaiki masalah dalam data dan dibuat *variable derived* serta memastikan apakah data sudah tepat untuk algoritma yang digunakan. Tahap ini sering mengalami peninjauan ulang ketika menemukan kendala pada pembangunan model, sehingga dilaksanakan iterasi hingga menemukan hal yang sesuai dengan data yang dimaksud.

### 4. Pemodelan (*Modeling*)

Pada tahap ini, dilakukan metode statistika dan *machine learning* untuk menentukan teknik, alat bantu serta algoritma data mining yang akan diterapkan. Kemudian langkah selanjutnya adalah menerapkan teknik dan algoritma tersebut pada data dengan alat bantu. Yang perlu digaris bawahi disini, beberapa teknik memungkinkan untuk digunakan pada data mining yang memiliki permasalahan yang sama. Jika diperlukan penyesuaian data terhadap metode data mining, kita dapat kembali ke tahapan *data preparation*.

### 5. Evaluasi (*Evaluation*)

Tahapan *evaluation* ini merupakan tahap evaluasi dengan melaksanakan interpretasi terhadap *output* dari data mining yang dihasilkan dalam tahapan sebelumnya. Evaluasi disini bertujuan agar model yang sudah ditentukan dapat sesuai dengan tujuan yang ingin dipenuhi pada fase pertama.

### 6. Penyebaran (*Deployment*)

Tahap *deployment* atau rencana penggunaan model merupakan fase yang penting dalam proses CRISP-DM. Perencanaan untuk tahap *deployment* dimulai sejak proses *Business Understanding* dilakukan. Fase *deployment* ini tidak hanya menghasilkan suatu model, tapi juga mengonversi skor putusan serta menggabungkan keputusan dalam sistem operasional.

Pada akhirnya, rencana sistem *deployment* mengakui bahwa tidak ada model yang statis. Model tersebut dibangun dari data yang diwakili data pada waktu tertentu, sehingga perubahan waktu dapat menyebabkan berubahnya karakteristik data. Model pun harus dipantau dan mungkin diganti dengan model yang sudah diperbaiki.

