

## BAB II

### LANDASAN TEORI

#### 2.1 Kajian Penelitian Terkait

Penelitian mengenai sistem penentuan kenaikan jabatan dengan metode *Naive Bayes* dan *Decision Tree* telah banyak dikembangkan. Seperti penelitian mengenai penentuan Kenaikan Jabatan Karyawan. Pada penelitian ini, sistem penentuan dikembangkan dengan metode *Naive Bayes* dan *Decision Tree C4.5*. Dengan metode ini, penentuan Kenaikan Jabatan Karyawan akan diklasifikasi berdasarkan lama bekerja dan keahlian dibidang tersebut. Berdasarkan penelitian ini, sistem mampu memprediksi penentuan Kenaikan Jabatan Karyawan yang layak untuk perusahaan.

#### 2.2 Analisis Kinerja

Kinerja adalah suatu keadaan yang menunjukkan kemampuan seseorang pegawai dalam menjalankan tugasnya sesuai standar yang ditetapkan oleh organisasi kepada pegawai sesuai dengan *job discription* (Siagian, 2019).

Kemudian menurut pendapat lain kinerja yaitu kemampuan menjalankan tugas dan pencapaian standar keberhasilan yang telah ditentukan oleh instansi kepada pegawai sesuai dengan pekerjaan yang telah diberikan kepada masing-masing pegawai (Kartono, 2020).

#### 2.3 Kenaikan Jabatan

Ukuran secara kualitatif dan kuantitatif yang menunjukkan tingkatan pencapaian sesuatu sasaran atau tujuan yang telah ditetapkan adalah merupakan sesuatu yang dapat dihitung dan diukur serta digunakan sebagai dasar untuk

menilai atau melihat bahwa kinerja setiap hari dalam instansi organisasi dan perseorangan terus mengalami peningkatan sesuai dengan rencana yang telah ditetapkan.

## **2.4 Data Mining**

*Data Mining* adalah serangkaian proses untuk menggali nilai tambah berupa informasi yang selama ini tidak diketahui secara manual dari suatu basis data. Informasi yang dihasilkan diperoleh dengan cara mengekstraksi dan menggali pola yang penting atau menarik data yang terdapat pada basis data.

Bhatia (2019) mengemukakan pendapat pribadinya mengenai *data mining*, menurutnya *data mining* adalah kumpulan metode atau teknik dengan tujuan untuk membuat alat penghasil informasi yang tidak diketahui sebelumnya. *Data Mining* digunakan untuk membuat pola berdasarkan data yang terdapat pada *database* dan digunakan untuk pengambilan keputusan suatu perusahaan.

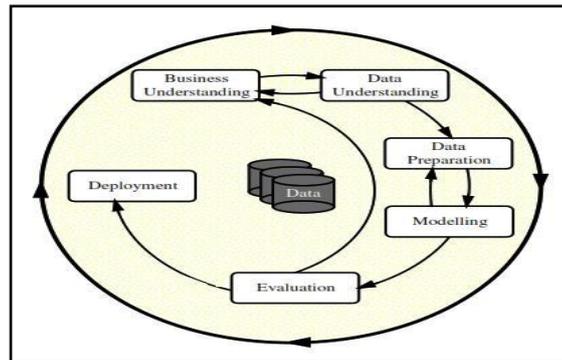
Secara umum terdapat 5 (lima) peranan dalam *data mining*, yaitu estimasi, prediksi, klasifikasi, clustering, dan asosiasi (Suntoro, 2019). Dimana masing-masing mempunyai definisi dan penggunaan algoritma sebagai berikut:

1. Estimasi, yang mirip dengan klasifikasi, tetapi variabel target estimasi lebih mengarah ke arah numerik daripada ke arah kategori. Algoritma yang banyak digunakan adalah *Linear Regression*, *Neural Network*, dan lain-lain.
2. Prediksi, prediksi menerka sebuah nilai yang belum diketahui dan juga memperkirakan nilai untuk masa mendatang. Algoritma yang banyak digunakan adalah *Support Vector Machine*, *Naive Bayes*, dan lain-lain.

3. Klasifikasi, terdapat target yang bertipe kategori, misalnya penggolongan pendapatan dapat dipisahkan dalam 3 kategori yaitu tinggi, sedang dan rendah. Algoritma yang banyak digunakan adalah *k-Nearest Neighbor (k-NN)*, *Naive Bayes*, *ID3*, dan lain-lain.
4. Clustering, merupakan pengelompokkan record, pengamatan, atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan. Algoritma yang banyak digunakan adalah *K-Means*, *Fuzzy C-Means*, dan lain-lain.
5. Asosiasi, bertugas untuk menemukan atribut yang muncul dalam satu waktu. Dalam dunia bisnis ini disebut dengan *Market Basket Analysis* (analisis kerangka belanja). Algoritma yang banyak digunakan adalah *FP-Growth*, *A-Priori*, dan lain-lain.

## **2.5 CRISP-DM**

Model *CRISP-DM* (*Cross-Industry Standard Process for Data Mining*) memberikan gambaran umum tentang tahapan dalam data mining, berisi harapan, tugas dan keluarannya. Siklus dalam *CRISP-DM* dipecah dalam enam fase, panah hanya menunjukkan yang penting dan ketergantungan antara setiap tahap (Swamynathan, 2019).



**Gambar 2.1** CRISP-DM (Cross-Industry Standard Process for Data Mining)

Sumber: *Mastering machine learning with python in six steps: A practical implementation guide to predictive data analytics using python*. Swamynathan (2019).

Terdapat 6 fase utama dalam kerangka kerja CRISP-DM, tahapannya adalah sebagai berikut (Swamynathan, 2019):

1. *Business Understanding*, fase awal ini berfokus pada pemahaman tujuan dan persyaratan dari perspektif bisnis, kemudian mengubah pengetahuan menjadi definisi masalah dalam *data mining* untuk mencapai tujuan yang diinginkan.
2. *Data Understanding*, dimulai dengan pengumpulan data dan dilanjutkan dengan kegiatan pengenalan data untuk mengidentifikasi kualitas data.
3. *Data Preparation*, mencakup kegiatan untuk membangun kumpulan data akhir yang akan dimasukkan ke dalam algoritma pemodelan. Persiapan data kemungkinan akan dilakukan beberapa kali dengan kegiatan seperti pemilihan atribut, pembersihan data, konstruksi atribut baru, dan transformasi data untuk alat pemodelan.

4. *Modelling*, pada fase ini teknik pemodelan dipilih dan diterapkan sesuai dengan data yang diperoleh dan parameter yang akan dihasilkan.
5. *Evaluation*, pada tahap ini penting untuk mengevaluasi model secara lebih menyeluruh serta meninjau langkah-langkah yang akan dijalankan untuk membangun model agar dipastikan model tersebut mencapai tujuan bisnis dengan benar.
6. *Deployment*, pada tahap ini dilakukan pembuatan kedalam alat sehingga dapat digunakan oleh pengguna, apabila melakukan perbandingan metode maka tahap *deployment* menggunakan model yang memiliki tingkat akurasi tertinggi.

## 2.6 Naive Bayes

Menurut Santosa (2019), teori keputusan *Bayes* adalah pendekatan statistika yang fundamental dalam *data mining*. Pendekatan ini didasarkan pada kuantifikasi *trade-off* antara berbagai keputusan klasifikasi dengan menggunakan probabilitas.

Klasifikasi *Naive Bayes* dapat digunakan untuk memprediksi probabilitas kenaikan jabatan karyawan. Algoritma *Naive Bayes* adalah salah satu algoritma klasifikasi berdasarkan *Teorema Bayes* pada ilmu statistika (Suntoro, 2019).

Teori *Bayes* menghitung nilai *posterior probability*  $P(H|X)$  menggunakan probabilitas  $P(H)$ ,  $P(X)$ , dan  $P(H|X)$ , dimana nilai  $X$  adalah *data testing* yang kelasnya belum diketahui. Nilai  $H$  adalah hipotesis data  $X$  yang merupakan suatu kelas yang lebih spesifik. Nilai  $P(X|H)$  atau *likelihood*, adalah probabilitas hipotesis data  $X$  berdasarkan kondisi  $H$ . Nilai  $P(H)$  atau disebut dengan *prior*

*probability* adalah probabilitas hipotesis H. Sedangkan nilai P(X) yang disebut sebagai prediktor *prior probability*, adalah probabilitas X.

$$P(H|X) = \frac{P(H|X)P(H)}{P(X)}$$

Suntoro (2019) mengatakan bahwa performa algoritma *naive bayes* sangat baik untuk melakukan *klasifikasi* pada dataset bertipe nominal atau kategorial, tetapi apabila *dataset* bertipe numerik atau angka maka digunakan perhitungan *distribusi gaussian*. Penghitungan *distribusi gaussian* dilakukan setelah perhitungan *rata-rata*  $\mu$  dan *standar deviasi*  $\sigma$ , dimana rumus nya adalah sebagai berikut:

#### 1. Rumus *Distribusi Gaussian*

Keterangan:

$x$  : Data dengan *class*  $x$  (belum diketahui).

$\sigma$  : Standar Deviasi (Populasi).

$\mu$  : Rata-rata.

#### 2. Perhitungan Rata-rata

Keterangan:

$x_i$  : Semua atribut dengan *class*  $x$  (belum diketahui).

$n$  : Jumlah atribut dengan *class*  $x$  (belum diketahui).

### 3. Perhitungan Standar Deviasi

Keterangan:

$x_i$  : Seluruh atribut dengan *class*  $x$  (belum diketahui).

$\mu$  : Rata-rata.

$n$  : Jumlah atribut dengan *class*  $x$  (belum diketahui).

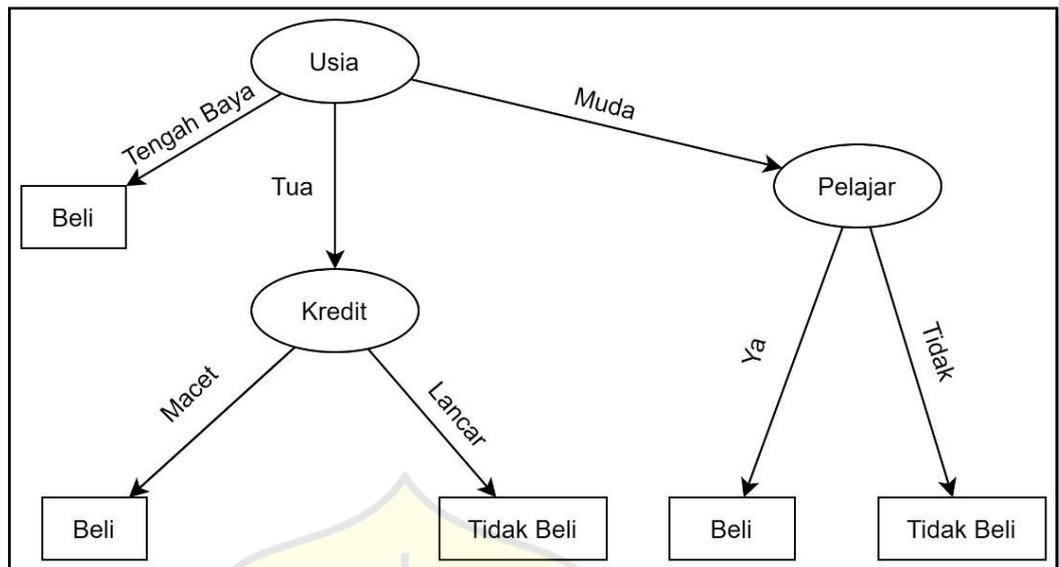
#### 2.7 Decision Tree

*Decision Tree* adalah salah satu metode klasifikasi yang populer dan banyak digunakan secara praktis. Metode ini berusaha menemukan model klasifikasi yang tahan terhadap derau. Salah satu metode *decision tree* yang sangat populer adalah *Iterative Dychotomizer version 3 (ID3)*. Dua varian lain yang populer adalah *C4.5* dan *ASSISTANT*.

Menurut Suntoro (2019) Algoritma *Decision Tree* terdiri dari sekumpulan *node* (simpul) yang dihubungkan oleh cabang, dimana cabang tersebut bergerak ke bawah dari *root* (akar) dan berakhir di *leaf* (daun) *node*.

*Leaf node* adalah *node* yang sudah tidak dapat dipecah lagi, *leaf node* merupakan prediksi jawaban dari masalah yang dihadapi. Sebagaimana yang kita ketahui pohon mempunyai akar dibawah, tetapi *Decision Tree* berbentuk terbalik. Dimana *root node* berada di paling atas, sedangkan *leaf node* berada di paling bawah.

Berikut merupakan gambaran dari Pohon Keputusan *Decision Tree* dengan contoh studi kasus klasifikasi keputusan pembelian komputer (Suntoro, 2019).



**Gambar 2.2** Contoh *Decision Tree*

Sumber: Data Mining: Algoritma dan Implementasi dengan Pemrograman php. (Suntoro, 2019).

## 2.8 *Confusion Matrix dan Performa Klasifikasi*

*Confusion Matrix* adalah alat yang berguna untuk menganalisa seberapa baik model klasifikasi yang telah dibuat (Swamynathan,2019). *Confusion Matrix* digunakan untuk mengevaluasi model klasifikasi yang akan digunakan untuk tahapan deployment. Berikut adalah contoh tabel *Confusion Matrix* untuk hasil

**Tabel 2.1** Contoh *Confusion Matrix*

<i>Actual Class/Predicted Class</i>	Benar	Salah
Benar	TP	FN
Salah	FP	TN

Sumber: Data Mining: Algoritma dan Implementasi dengan Pemrograman php. Suntoro (2019).

## **2.9 Split Dataset**

*Split Dataset* adalah teknik pengecekan performa model yang membagi data yang telah melalui tahap *data preparation* menjadi dua bagian secara acak, sebagian digunakan sebagai *data training* dan sebagian lainnya digunakan untuk *data testing*.

Swamynathan (2019) mengatakan bahwa untuk menguji performa dari model yang buat cara yang paling umum adalah dengan membagi *dataset* menjadi dua bagian salah satunya sebagai *data training* dan lainnya sebagai *data testing*. Secara umum teknik *split dataset* adalah dengan membagi *dataset* menjadi 80/20 atau 70/30 untuk *train/set*.

## **2.10 Website**

Website merupakan kumpulan halaman digital yang berisi informasi berupa teks, animasi, gambar, suara, dan video dari semuanya yang terkoneksi pada jaringan internet (Sunarti dkk, 2019). Website mempunyai fungsi utama yaitu menyampaikan informasi kepada pengunjung dari halaman website tersebut.

## **2.11 Script Pada Aplikasi Web**

### **2.11.1 HTML**

Proses tampilnya sebuah halaman website di browser melibatkan *HTML*. *HTML* bahasa pemrograman yang digunakan untuk menampilkan *website* (Sunarti dkk, 2019). *Hypertext Markup Language* tergolong dalam salah satu format yang digunakan dalam pembuatan dokumen yang terbaca oleh *web*.

### 2.11.2 CSS

*CSS (Cascading Style Sheets)* merupakan bahasa yang digunakan untuk mengatur tampilan desain sebuah halaman *HTML* yang ditulis dalam bahasa *markup/markup language* (Sunarti dkk, 2019).

### 2.11.3 JavaScript

*Javascript* merupakan bahasa *script* yang digunakan dalam dunia teknologi terutama internet, bahasa ini dapat digunakan di banyak *web browser* (Sunarti dkk, 2019). *Javascript* (js) ialah suatu bahasa *scripting* yang digunakan sebagai fungsionalitas dalam membuat suatu *website*.

Nixon (2018) menjelaskan bahwa *javascript* apabila dikombinasikan dengan *css* akan menghasilkan tampilan *website* yang dinamis, serta *javascript* merupakan salah satu bahasa pemrograman *multi-platform*.

### 2.11.4 Bootstrap

*Bootstrap* merupakan salah satu *framework CSS* yang sangat populer dikalangan pencipta pemrograman *website*. Chiaretta (2018) mengatakan bahwa *Bootstrap* adalah *framework HTML, CSS, JavaScript* yang berfungsi untuk mendesain *responsive* dengan cepat dan akurat, *bootstrap* berisi kumpulan *CSS, HTML, dan JS* untuk mengubah tampilan dari *form, elements, buttons, navigations*, dan banyak dari komponen *user interface* lainnya.

### **2.11.5 Composer**

Menurut Suntoro (2019) *Composer* adalah perangkat lunak yang digunakan untuk manajemen dependensi pada pemrograman *PHP*. *Composer* biasanya digunakan untuk membuat *library* pada *project/proyek* yang dikerjakan.

### **2.11.6 jQuery**

*jQuery* adalah *library* atau pustaka *JavaScript multi-platform* yang didesain untuk menyederhanakan *client-side scripting* pada *HTML*. Menurut Nixon (2018) *jQuery* menyediakan lapisan abstrak yang memiliki tujuan umum untuk *website scripting* bahkan dapat mencakup untuk banyak penggunaan seperti *HTML* dan *DOM manipulation*, fungsi spesial dalam interaksi secara langsung dengan *CSS*, membuat animasi, dan lainnya.

## **2.12 Database SQL**

### **2.12.1 MySQL**

*MySQL* dikategorikan sebagai perangkat lunak dan sistem pembuat *datarbase* yang bersifat terbuka (*open source*) dan berjalan di berbagai sistem operasi baik di *Windows* maupun *Linux*.

Menurut West (2018) *MySQL* adalah sebuah perangkat lunak sistem yang digunakan untuk manajemen basis data *SQL* atau *DBMS (Database Management System)* yang *multithread, multi-user*.

### **2.12.2 *PhpMyAdmin***

Menurut West (2018) *PhpMyAdmin* merupakan sebuah aplikasi *open source* yang berfungsi untuk memudahkan manajemen *MySQL*, *PhpMyAdmin* dapat memperbaharui data dengan *WUI* yang baik dan terasa mudah tanpa perlu mengetikkan perintah *SQL* secara manual.

## **2.13 Perangkat Lunak yang dibutuhkan**

### **2.13.1 *XAMPP***

Menurut Suntoro (2019) *XAMPP* adalah perangkat lunak yang bersifat *open source*, aplikasi *Apache* (web server) yang mudah dipasang dan berisikan *MariaDB*, *PHP*, dan *Perl*.

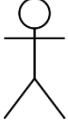
## **2.14 Pemodelan Sitem dengan *UML***

Menurut Mulyani (2017) *Unified Modeling Language* adalah sebuah teknik pengembangan sistem yang banyak digunakan di berbagai kegiatan dalam perencanaan terhadap suatu sistem, menggunakan bahasa grafis sebagai bahan untuk pedokumentasian dan melakukan spesifikasi pada sistem.

### **2.14.1 *Use Case Diagram***

Secara definisi *Use Case Diagram* adalah penggambaran yang dilakukan untuk menunjukkan hubungan antara pengguna dengan sistem yang dirancang (Mulyani, 2017). Diagram ini menggambarkan sistem secara global serta elemen yang digunakan cukup sedikit. Berikut ini elemen-elemen yang digunakan pada *use case diagram*.

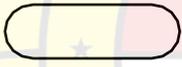
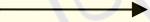
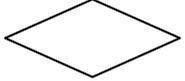
**Tabel 2.2** Notasi *UML Use Case Diagram*

Komponen <i>Use Case</i>	Penjelasan
 <p>nama aktor</p>	<p><i>Actor</i> : mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>.</p>
 <p>Use case</p>	<p><i>Use case</i> : abstraksi dan interaksi antara sistem dan <i>actor</i>.</p>
 <p>Relasi <i>Association</i></p>	<p><i>Association</i> : abstraksi dari penghubung antara <i>actor</i> dan <i>use case</i>.</p>
 <p>Relasi <i>Generalization</i></p>	<p><i>Generalization</i> : menunjukkan spesialisasi <i>actor</i> untuk dapat berpartisipasi dengan <i>use case</i>.</p>
<p>&lt;&lt;<i>include</i>&gt;&gt;</p> 	<p>Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.</p>
<p>&lt;&lt;<i>extends</i>&gt;&gt;</p> 	<p>menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi.</p>

### 2.14.2 Activity Diagram

Menurut Mulyani (2017) *Activity Diagram* merupakan diagram yang digunakan sebagai penggambaran aktivitas pada proses, logika, dan hubungan antar aktor dengan alur-alur kerja *use case*. Berikut ini elemen-elemen yang digunakan pada *activity diagram*. *Activity diagram* mempunyai peran seperti hanya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak.

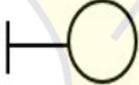
**Tabel 2.3** Notasi UML *Activity Diagram*

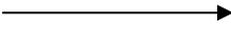
<i>Activity Diagram</i>	Penjelasan
 <i>Start State</i>	Sebuah diagram aktivitas memiliki sebuah status awal.
 <i>Activity</i>	<i>Activity</i> , memiliki fungsi aktifitas pada proses sistem.
 <i>State Transition</i>	<i>State Transition</i> , berfungsi untuk menunjukkan aliran atau urutan dari <i>event</i> atau aktivitas pada diagram.
 <i>Horizontal Synchronization</i>	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
 <i>Decision</i>	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
 <i>End State</i>	Status akhir yang di lakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

### 2.14.3 Sequence Diagram

Menurut Mulyani (2017) *Sequence Diagram* merupakan suatu penyajian perilaku yang disusun seperti rangkaian langkah-langkah percontohan dari waktu ke waktu. *Sequence Diagram* digunakan untuk menggambarkan alur pekerjaan, pesan yang disampaikan dan menggambarkan elemen-elemen di dalamnya bekerja sama untuk mencapai suatu hasil. Berikut ini elemen-elemen yang digunakan pada *sequence diagram*.

**Tabel 2.4** Notasi UML *Sequence Diagram*

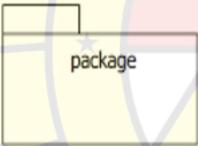
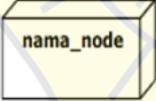
<i>Sequence Diagram</i>	Penjelasan
 nama aktor	Menggambarkan orang yang sedang berinteraksi dengan sistem.
 <i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan.
 <i>Boudary Class</i>	Menggambarkan sebuah gambaran dari form.
 <i>Control Class</i>	Menggambarkan penghubung antara boundary dengan tabel.
 <i>A focus of control &amp; A life line</i>	Menggambarkan tempat mulai dan berakhirnya message.

 <i>Object Message</i>	Menggambarkan pengiriman pesan.
--	---------------------------------

#### 2.14.4 Deployment Diagram

*Deployment Diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi dan juga digunakan untuk memodelkan hal-hal seperti sistem tambahan dan sistem *client/server* (Mulyani, 2017). Berikut ini elemen-elemen yang digunakan pada *deployment diagram*.

**Tabel 2.5** Notasi *UML Deployment Diagram*

Simbol	Deskripsi
 Package	Package merupakan sebuah bungkus dari satu atau lebih node.
 Node	Biasanya mengacu 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 di definisikan sebelumnya pada diagram komponen.
 Keberuntungan/dependency	Keberuntungan antar node, arah panah mengarah pada node yang dipakai.

<hr/> Link	Relasi antar node.
---------------	--------------------

