



**BAB II**

**TEKNOLOGI INFORMASI**

**UNIVERSITAS DARMA PERSADA**

## BAB II

### LANDASAN TEORI

#### 2.1 Kajian Penelitian Terkait

Penelitian mengenai sistem prediksi dengan metode *Naïve Bayes* dan *Decision Tree* telah banyak dikembangkan. Seperti penelitian mengenai prediksi Mahasiswa yang mendapatkan Beasiswa. Pada penelitian ini, sistem prediksi dikembangkan dengan metode *Naïve Bayes* dan *Decision Tree C4.5*. Dengan metode ini, predikat kelulusan Mahasiswa akan diprediksi serta diklasifikasi berdasarkan nilai tes masuk berupa nilai Matematika, Bahasa Indonesia, Bahasa Inggris, dan juga pilihan Program Studi. Berdasarkan penelitian ini, sistem mampu memprediksi perolehan predikat kelulusan Mahasiswa dan mengetahui atribut yang paling berpengaruh terhadap tingkat perolehan predikat kelulusan mahasiswa.

#### 2.2 Analisis Prediktif

Prediksi adalah suatu proses memperkirakan secara sistematis tentang sesuatu yang potensi tertinggi terjadinya suatu hal di masa depan berdasarkan kejadian yang telah terjadi dimasa lalu (Suntoro, 2019). Dalam ilmu statistik diketahui terdapat salah satu metode dalam pemecahan suatu permasalahan yang akan datang yaitu *Predictive Analytics* atau Analisis Prediktif.

Analisis Prediktif merupakan kemampuan untuk membuat perkiraan ataupun prediksi yang memungkinkan seseorang mengetahui tentang peristiwa atau kejadian dimasa yang akan datang berdasarkan pola masa lalu atau sejarah (Swamynathan, 2019). Analisis Prediktif dapat diimplementasikan ke berbagai model seperti *data mining*, statistika, *machine learning*, dan *artificial intelligence*.

#### 2.3 Kelulusan Mahasiswa

Pada Buku Standar Pendidikan Universitas Darma Persada (2017) Seorang mahasiswa S1 dinyatakan lulus program studi apabila telah menyelesaikan minimal SKS paling sedikit 144 (seratus empat puluh empat) dengan Indeks Prestasi Kumulatif (IPK) minimal 2.00, menyelesaikan Tugas Akhir atau Skripsi, serta memiliki masa studi paling lama 7 (tujuh) tahun, setelah itu Ketua Program Studi melaporkan kepada Dekan mahasiswa yang telah memenuhi persyaratan untuk lulus untuk diajukan pengesahan kelulusannya kepada Rektor. Rektor menerbitkan ijazah bagi mahasiswa yang telah dinyatakan lulus. Seorang mahasiswa dinyatakan lulus berhak memakai gelar sarjana untuk Program Strata Satu.

Sesuai dengan Pasal 27, Peraturan Kementerian Pendidikan dan Kebudayaan No.3 tahun 2020 tentang Standar Nasional Pendidikan Tinggi, Indeks Prestasi Kumulatif sebagai dasar penentuan predikat kelulusan ditentukan sebagai berikut:

1. IPK 2.76–3.00: Lulus dengan predikat memuaskan.
2. IPK 3.01–3.50: Lulus dengan predikat sangat memuaskan.
3. IPK 3.51-4.00: Lulus dengan predikat pujian.

#### **2.4 Data Mining**

Menurut John Naisbitt dalam Bhatia (2019) kita terbenam di dalam data, namun kita kekurangan informasi dan pengetahuan. Hal tersebut menjadi gambaran apa yang terjadi saat ini dimana kita mempunyai data yang sangat banyak tetapi data tersebut hanya menjadi sampah di memori penyimpanan apabila tidak diolah menjadi sebuah informasi yang berguna.

Bhatia (2019) mengemukakan pendapat pribadinya mengenai *data mining*, menurutnya *data mining* adalah kumpulan metode atau teknik dengan tujuan untuk

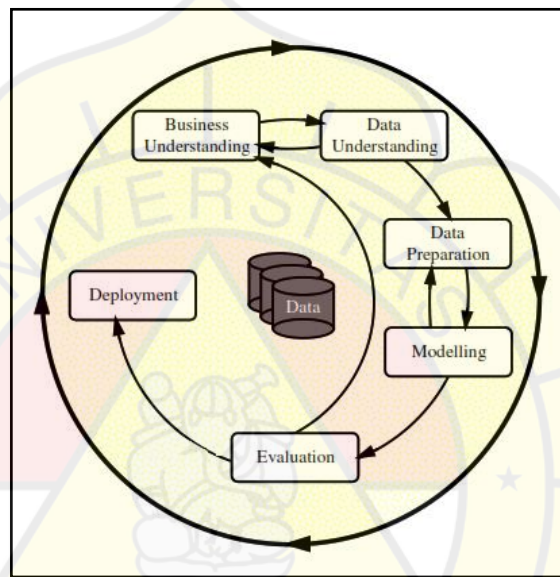
membuat alat penghasil informasi yang tidak diketahui sebelumnya. *Data Mining* digunakan untuk membuat pola berdasarkan data yang terdapat pada *database* dan digunakan untuk pengambilan keputusan suatu perusahaan.

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 dari pada 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*, *Naïve 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)*, *Naïve Bayes*, *ID3*, dan lain-lain.
4. *Clustering*, merupakan pengelompokan 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 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 tahapan, 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. *Modeling*, 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 *Naïve Bayes*

Klasifikasi *Naïve Bayesian* dapat digunakan untuk memprediksi probabilitas keanggotaan suatu kelas. Algoritma *Naïve Bayes* adalah salah satu algoritma klasifikasi berdasarkan *teorema Bayesian* pada ilmu statistika (Suntoro, 2019).

Teori *Bayesian* 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.

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

Suntoro (2019) mengatakan bahwa performa algoritma *naïve 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 rumusnya adalah sebagai berikut:

1. Rumus *Distribusi Gaussian*

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi} \cdot \sigma} \exp \frac{-(x-\mu)^2}{2\sigma^2}$$

Keterangan:

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

$\sigma$  : Standar Deviasi (Populasi).

$\mu$  : Rata-rata.

2. Perhitungan Rata-rata

$$\mu = \frac{\sum_i^n x_i}{n}$$

Keterangan:

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

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

3. Perhitungan Standar Deviasi



$$\sigma = \sqrt{\frac{\sum_i^n (x_i - \mu)^2}{n - 1}}$$

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*

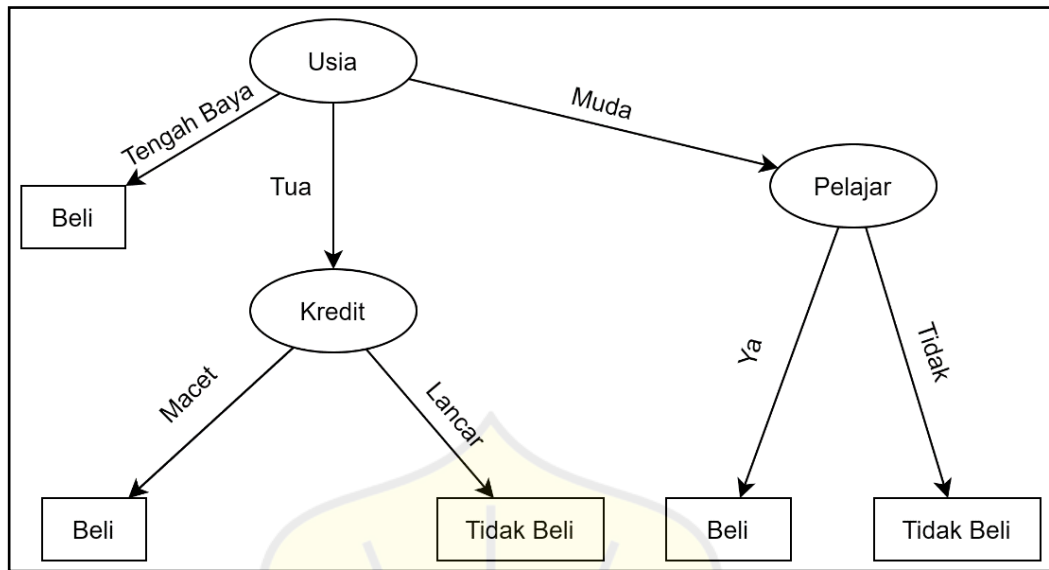
*Decision Trees* dan *Decision Rules* merupakan metodologi dalam *Data Mining* yang banyak diaplikasikan di dunia nyata sebagai solusi yang kuat untuk permasalahan klasifikasi. Algoritma *Decision Tree* membangun pohon keputusan dari sebuah *data training* berupa *record-record* dalam basis data (Bhatia, 2019).

Menurut Suntoro (2019) Algoritma *Decision Trees* 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 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 Algoritma C4.5

*Algoritma C4.5* adalah salah satu algoritma *Data Mining* yang secara khusus diterapkan pada algoritma *Decision Tree*. *Algoritma C4.5* ini merupakan penyempurnaan dari *algoritma ID3* yang dibentuk oleh Ross Quinlan pada tahun 1986. Menurut Bhatia (2019) hal yang paling penting dalam *Algoritma C4.5* adalah proses dalam pembentukan *Decision Tree* dari kumpulan *data training*.

*Algoritma C4.5* menggunakan perhitungan *entropy*, *information gain*, *split info*, dan *gain ratio* untuk pemilihan atribut menjadi *node*, dimana masing masing perhitungan adalah sebagai berikut (Suntoro, 2019):

1. Perhitungan *Entropy*

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

Keterangan:

$S$  : Himpunan Kasus.

$p_i$  : Proporsi himpunan kasus ke- $i$  terhadap himpunan kasus.

$n$  : Jumlah partisi  $S$ .

2. Perhitungan *Gain*

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

Keterangan:

$S$  : Himpunan Kasus.

$A$  : Atribut.

$n$  : Jumlah partisi atribut  $A$ .

$|S_i|$  : Proporsi  $S_i$  terhadap  $S$ .

$|S|$  : Jumlah kasus dalam  $S$ .

3. Perhitungan *Split Info*

$$SplitInfo_A(D) = - \sum_{j=1}^v \frac{|D_j|}{|D|} * \log_2 \left( \frac{|D_j|}{|D|} \right)$$

Keterangan:

$D$  : Ruang data sampel yang digunakan untuk training.

$D_j$  : Jumlah sampel pada atribut  $j$ .

4. Perhitungan *Gain Ratio*

$$GainRatio = \frac{Gain(S, A)}{SplitInfo_A(D)}$$

Proses diulang untuk masing-masing cabang sampai semua cabang memiliki kelasnya masing-masing.

## 2.9 *Data Training dan Data Testing*

Data yang akan digunakan dalam pengujian klasifikasi dibagi menjadi dua yaitu *data training* dan *data testing*. Data yang sudah diketahui sebelumnya untuk label kelas dan digunakan untuk membangun model *classification* disebut dengan

*data training*. Data yang sudah diketahui (dianggap belum diketahui) sebelumnya untuk label kelas dan digunakan untuk membangun model *classification* disebut dengan *data testing* (Swamynathan, 2019).

Model klasifikasi kemudian dibangun berdasarkan *data training* dan kemudian kinerjanya diukur berdasarkan *data testing*. Proporsi pembagian *data training* dan *data testing* biasanya diskrit, misal 90:10 (yang berarti 90% digunakan sebagai *data training* dan 10% digunakan sebagai *data testing*). Jumlah *data training* dan *data testing* dapat dihitung menggunakan persamaan sebagai berikut:

$$\text{Jumlah } data \text{ training} = \text{Proporsi } data \text{ training} \times N$$

$$\text{Jumlah } data \text{ testing} = N - \text{Jumlah } data \text{ training}$$

Pembagian *data training* dan *data testing* dengan metode proporsi adalah metode yang paling sederhana namun memiliki beberapa keterbatasan. Jumlah *data training* lebih sedikit yang tersedia untuk pelatihan karena sebagiannya digunakan sebagai *data testing*. Akibatnya, model yang dibangun kemungkinan tidak sebagus apabila menggunakan semua data sebagai *data training*.

## 2.10 *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 dengan kelas benar dan salah:

**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).

*Actual Class* adalah kelas yang sebenarnya pada *test set*. *Predicted class* adalah kelas hasil prediksi dari model yang dihasilkan oleh *classifier*. Suntoro (2019) mengatakan bahwa *akurasi* merupakan persentase baris *test set* yang diklasifikasikan dengan benar. Terdapat pula beberapa hal yang perlu diketahui dalam melihat kinerja model klasifikasi seperti *Error Rate*, *Recall (Sensitivity)*, *Precision*, dan *F1-score*. Berikut adalah penjelasan dari masing-masing perhitungan kinerja klasifikasi (Swamynathan, 2019):

**Tabel 2. 2** Matriks Performa Klasifikasi

<i>Metric</i>	Keterangan	Perhitungan
<i>Accuracy</i>	Berapa % yang diprediksi benar.	$\frac{TP + FP}{TP + TN + FP + FN}$
<i>Misclassification Rate</i>	Berapa % yang diprediksi salah.	$\frac{FN + FP}{TP + TN + FP + FN}$
<i>Recall</i>	Berapa % perbandingan antara TP dengan banyaknya data yang diprediksi salah tetapi mempunyai nilai aktual benar (FN).	$\frac{TP}{(TP + FN)}$
<i>Precision</i>	Berapa % perbandingan antara TP dengan banyaknya data yang diprediksi benar tetapi mempunyai nilai aktual salah (FP).	$\frac{TP}{(TP + FP)}$
<i>F1-Score</i>	Berapa % rata-rata harmonik dari nilai <i>recall</i> dan <i>precision</i> .	$2 * \frac{precision * recall}{precision + recall}$

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

## 2.11 *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/test*.

## **2.11 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.

## **2.13 Script pada aplikasi Web**

### **2.13.1 HTML**

Proses tampilnya sebuah halaman website di browser melibatkan *HTML*. *HTML* adalah 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*.

### **2.13.2 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).

### **2.13.3 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*.

### **2.13.4 PHP**

*PHP (Hypertext Preprocessor)* merupakan bahasa *script server-side* (berbasis *server*) yang dapat menjalankan kode dengan ekstensi *.php*, sehingga menghasilkan tampilan website yang dinamis (Sunarti dkk, 2019). Sedangkan menurut Nixon (2018) *PHP* adalah bahasa pemrograman yang menggunakan server untuk menghasilkan *output* secara dinamis. *PHP* banyak dipakai oleh *programmer* untuk membuat situs *web* yang bersifat dinamis karena gratis dan mudah dipelajari.

### **2.13.5 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 website *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.

### **2.13.6 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.

### **2.13.7 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.

## **2.14 Database SQL**

### **2.14.1 MySQL**

*MySQL* dikategorikan sebagai perangkat lunak dan sistem pembuat *database* 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*.



### **2.14.2 *PhpMyAdmin***

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

## **2.15 Perangkat Lunak yang dibutuhkan**

### **2.15.1 *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*.

### **2.15.2 *Visual Studio Code***

Menurut Sole (2019) *Visual Studio Code* merupakan perangkat lunak *teks editor multi-platform* untuk mengembangkan aplikasi, dengan menggunakan *visual studio code* pengguna dapat menulis kode-kode program, menjalankan kode program, melakukan pengujian, *debugging*, dan banyak lagi.




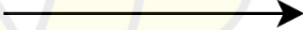
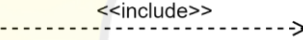
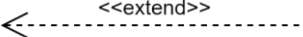
## **2.16 Pemodelan Sistem dengan *UML***

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 pendokumentasian dan melakukan spesifikasi pada sistem.

### **2.16.1 *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. 3** Notasi *UML Use Case Diagram*

Penjelasan	Notasi <i>UML</i>
<b>Actor:</b> Elemen yang menjadi pemicu sistem.	 Actor
<b>Use Case:</b> Potongan proses yang merupakan bagian dari sistem.	
<b>Association:</b> Menggambarkan interaksi antara <i>use case</i> dan aktor.	
<b>Generalization:</b> Menggambarkan pewarisan antara dua aktor atau <i>use case</i> .	
<b>Dependency:</b> Menggambarkan relasi antara dua <i>use case</i> . Terdapat 2 tipe yaitu, <i>include</i> menggambarkan kebutuhan <i>use case</i> satu dengan yang lainnya sementara <i>extends</i> tergantung pada kondisi.	 

Sumber: Analisis dan Perancangan Sistem Informasi Manajemen Keuangan




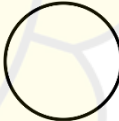

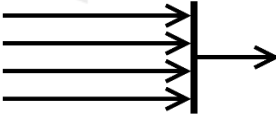
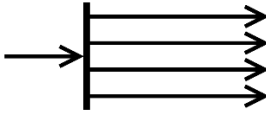
Daerah: Notasi Pemodelan Unified Modeling Language (UML).

Sri Mulyani (2017).

### 2.16.2 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*.

**Tabel 2. 4** Notasi *UML Activity Diagram*


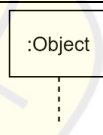
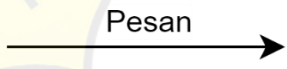
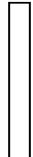
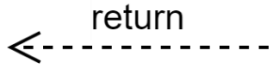
Penjelasan	Notasi UML
<b>Activities:</b> Elemen yang digunakan untuk menggambarkan aktivitas.	
<b>Transitions:</b> Menggambarkan transisi dari elemen yang satu ke elemen lainnya.	
<b>Decisions:</b> Elemen yang digunakan untuk percabangan logika.	
<b>Start Point:</b> Elemen yang digunakan untuk memulai <i>activity diagram</i> .	
<b>End Point:</b> Elemen yang digunakan untuk mengakhiri <i>activity diagram</i> .	
<b>Join:</b> Penggabungan lebih dari satu aktivitas menjadi satu.	
<b>Fork:</b> Pemisah dari satu aktivitas menjadi banyak aktivitas.	

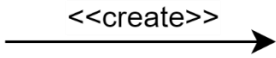
Sumber: Analisis dan Perancangan Sistem Informasi Manajemen Keuangan  
 Daerah: Notasi Pemodelan Unified Modeling Language (UML).  
 Sri Mulyani (2017).

### 2.16.3 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. 5** Notasi UML *Sequence Diagram*

Penjelasan	Notasi UML
<b>Aktor:</b> Merupakan elemen pemicu sistem.	
<b>Lifeline:</b> Merupakan objek yang berinteraksi.	
<b>Pesan:</b> Merupakan komunikasi antar objek partisipan.	
<b>Waktu Aktivasi:</b> Menyatakan objek dalam keadaan aktif dan sedang berinteraksi.	
<b>Pesan Return:</b> Menyatakan objek telah menjalankan suatu operasi.	

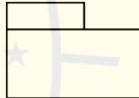
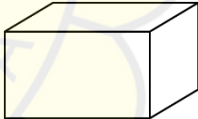


<b>Pesan <i>Create</i>:</b> Menyatakan objek membuat objek yang lain.	
---	---

Sumber: Analisis dan Perancangan Sistem Informasi Manajemen Keuangan  
Daerah: Notasi Pemodelan Unified Modeling Language (UML).  
Sri Mulyani (2017).

#### 2.16.4 *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. 6** Notasi *UML Deployment Diagram*

<b>Penjelasan</b>	<b>Notasi UML</b>
<b><i>Package</i>:</b> Merupakan sebuah bungkusan dari satu <i>node</i> atau lebih.	
<b><i>Node</i>:</b> Berisi <i>hardware</i> , <i>software</i> dan komponen rancangan.	
<b>Ketergantungan/<i>dependency</i>:</b> Ketergantungan antar <i>node</i> .	
<b><i>Link</i>:</b> Relasi antar <i>node</i> .	

Sumber: Analisis dan Perancangan Sistem Informasi Manajemen Keuangan  
Daerah: Notasi Pemodelan Unified Modeling Language (UML).  
Sri Mulyani (2017).