

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Preferensi**

Menurut Dhewi (2005), preferensi adalah suatu perangkat mental yang terdiri dari suatu campuran dari perasaan, harapan, pendirian, prasangka, rasa takut, atau kecenderungan lain yang mengarahkan individu kepada pilihan tertentu. Preferensi dapat diartikan seperangkat objek yang dinilai sesuai atau mendekati kesesuaian dengan persyaratan yang dikehendaki oleh para individu. Konsep utamanya adalah konsepnya mengasumsikan bahwa seperangkat stimulasi yang diterima agar dapat disajikan dalam bentuk suatu ruang multidimensi.

#### **2.2 Sistem Informasi**

Menurut Geovanne Farell dkk (2018), Pengertian sistem informasi sebuah sistem yang dapat didefinisikan dengan pendekatan prosedur yang mempunyai tujuan tertentu. Sebagai kumpulan dan komponen yang saling berhubungan satu dengan yang lainnya membentuk kesatuan yang memiliki karakteristik.

Menurut Mihadi Yananto tahun (2019), Sistem Informasi yaitu kumpulan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan untuk mengintegrasikan data, memproses dan menyimpan serta mendistribusikan informasi. Sistem informasi memiliki komponen-komponen yang terdapat didalamnya yaitu terdiri dari blok masukan, blok model, blok keluaran, blok teknologi, blok basis data, dan blok kendali. mengartikan sistem informasi yaitu suatu sistem yang di dalamnya mempunyai suatu informasi, yang memiliki kebutuhan pengolahan sistem. Agar dapat membantu fungsi sistem informasi tersebut dan dapat menyediakan kepada pihak-pihak tertentu, dengan informasi yang diperlukan untuk pengambilan keputusan.

#### **2.3 Machine Learning**

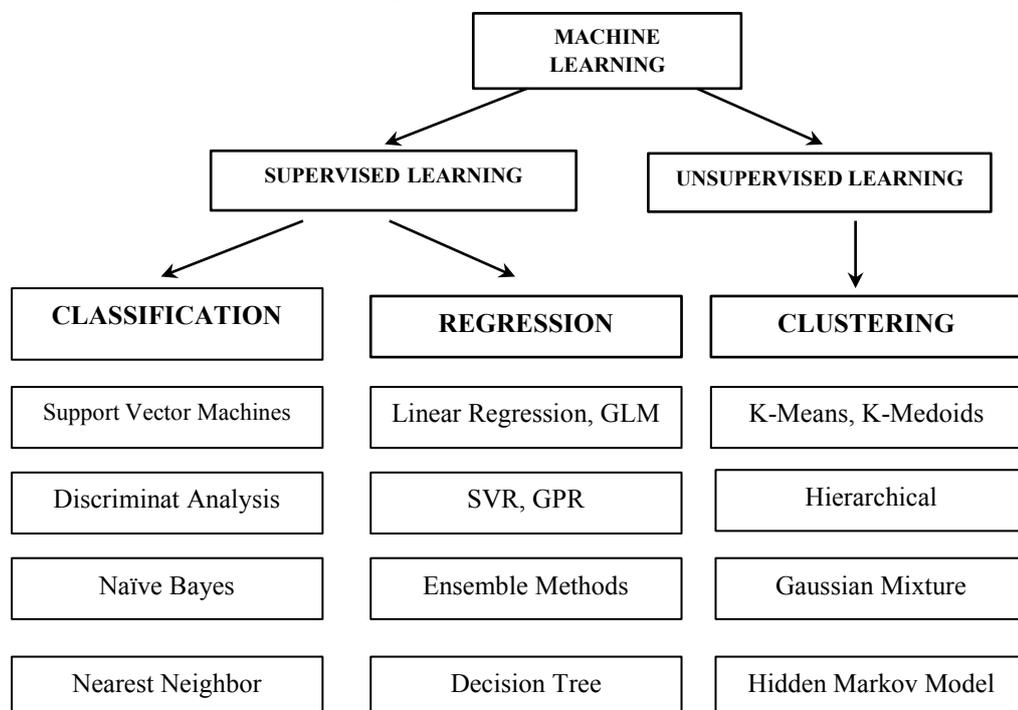
##### **2.3.1 Definisi Machine Learning**

Menurut Indriyanti dan Subekti (2018) Menyatakan, *Machine learning* yaitu metode analisis yang mengerjakan data besar (*big data*) dengan cara mengembangkan algoritma komputer. Dengan menggunakan data, pada

pembelajaran mesin yang memungkinkan komputer menemukan wawasan tersembunyi tanpa diprogram secara eksplisit saat mencarinya.

Sementara pada algoritma *machine learning*, kemampuannya secara otomatis menggunakan perhitungan matematis yang kompleks ke data besar dan dapat lebih efektif pada perkembangan terakhir. Model ini juga dapat menganalisis data yang lebih besar dan lebih kompleks dengan memberikan hasil yang lebih cepat dan akurat dalam skala yang sangat besar. Pada prediksi nilai tinggi bisa mengarah pada keputusan dan tindakan cerdas secara *real-time* tanpa campur tangan manusia.

Pada model ini kita dapat melakukan prediksi yang mempunyai dua macam, tipe keluarannya. Jika hasil prediksi bersifat diskrit, maka hal ini dinamakan proses klasifikasi. Salah satu teknik pengklasifikasian *machine learning* adalah *supervised learning*. Seperti yang dibahas sebelumnya dan penelitian terdahulu, *machine learning* tidak akan bisa bekerja tanpa adanya data. Oleh karena itu hal utama yang disiapkan adalah data. Data biasanya akan dibagi menjadi 2 kelompok, yaitu data *training* dan data *testing*. Berikut Beberapa Metode dan algoritma pada *machine learning* digambar 2.1 :



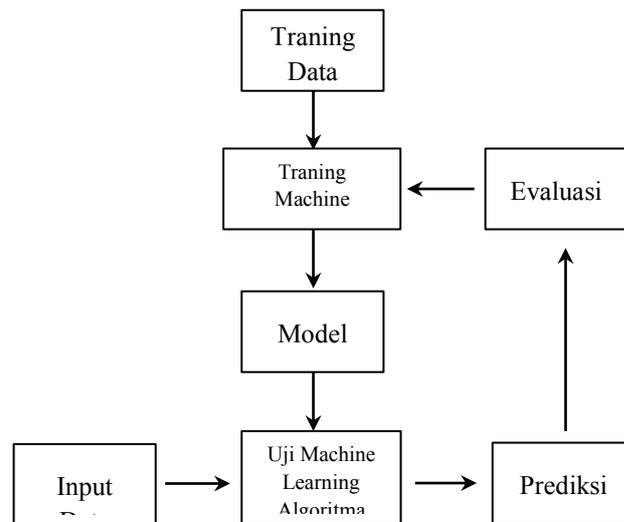
**Gambar 2.1.** Metode Algoritma Pada *Machine Learning*

### 2.3.2 Cara Kerja Machine Learning

Menurut Indriyanti dan Subekti (2018, p. 227) Menyatakan, cara kerja *machine learning* yaitu belajar seperti manusia dengan menggunakan contoh-contoh dan setelah itu barulah dapat menjawab suatu pertanyaan terkait. Proses belajar ini menggunakan data yang disebut *train dataset*. Berbeda dengan program *statis*, *machine learning* diciptakan untuk membentuk program yang dapat belajar sendiri.

Dari data tersebut, komputer akan melakukan proses belajar (*training*) dan dapat menghasilkan suatu model. Pada proses penelitian ini penulis menggunakan salah satu algoritma *machine learning* sebagai penerapan dalam prediksi. Model inilah yang menghasilkan suatu informasi dan dapat dijadikan sebagai pengetahuan dalam memecahkan suatu permasalahan pada proses *input* dan *output*. Model yang didapatkan bisa melakukan klasifikasi atau pun prediksi kedepannya. Maka untuk memastikan efisiensi suatu model yang terbentuk, dari data yang akan dibagi menjadi data pembelajaran (*train dataset*) dan data pengujian (*test dataset*).

Pada pembagian data dapat bervariasi tergantung algoritma yang digunakan. Umumnya *train dataset* lebih banyak dari *test dataset*, dengan rasio 3:1. *Test dataset* yang digunakan untuk menghitung seberapa efisien model yang dihasilkan dalam melakukan prediksi kedepannya, dan semakin banyak data yang digunakan maka yang dihasilkan semakin baik. Inilah yang membuat seseorang selalu bergantung pada teknologi tersebut, karena sesuatu yang ingin dikerjakan. Bahkan saat seseorang memberikan instruksi yang jelas, biasanya masih saja salah. Maka seseorang harus begitu eksplisit dengan sistem ini sehingga kesempatan itu dapat menjadi lebih mudah. Berikut cara kerja machine learning pada gambar 2.2 :



**Gambar 2.2.** Cara Kerja *Machine Learning*.

### 2.3.3 Metode Algoritma Machine Learning

Menurut Indriyanti dan Subekti (2018) Menyatakan, Metode Algoritma *Machine Learning* diantaranya adalah :

1. *Supervised machine learning algorithms*

*Supervised machine learning* yaitu algoritma yang ada pada metode *machine learning* dapat mendapatkan sebuah informasi yang ada pada data dengan memberikan label tertentu, contohnya pada data yang sudah di klasifikasi sebelumnya dengan terarah.

2. *Unsupervised machine learning algorithms*

*Unsupervised machine learning* yaitu suatu algoritma *machine learning* yang dapat digunakan pada suatu data yang tidak memiliki sebuah informasi, dengan hanya dapat diterapkan secara langsung (tidak terarah). Maka algoritma ini diharapkan sebagai struktur tersembunyi pada data yang tidak berlabel.

3. *Semi-supervised machine learning algorithms*

*Semi-supervised machine learning* yaitu algoritma yang dapat digunakan hanya untuk melakukan data berlabel dan tanpa label. Sistem yang dapat menggunakan metode ini dengan meningkatkan suatu efisiensi *output* yang dihasilkan.

#### 4. *Reinforcement machine learning algorithms*

*Reinforcement machine learning* yaitu algoritma yang mempunyai kemampuan dalam proses pembelajar yang dilakukan, algoritma ini hanya mendapatkan model yang diberikan semakin baik atau mengurangi kesalahan saat model yang dihasilkan tidak baik. Salah satunya penerapan pada mesin pencari.

### **2.4 Metode Supervised Learning**

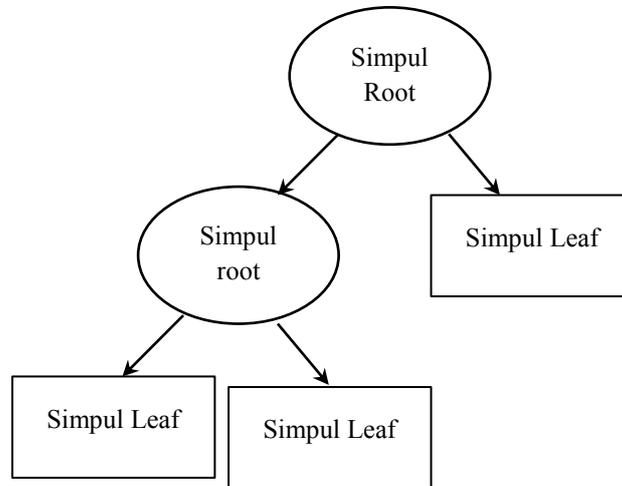
Dalam penulisan skripsi ini penulis menggunakan salah satu algoritma dari metode *machine learning* yaitu *Supervised Learning* yang merupakan algoritma yang membangkitkan suatu fungsi yang memetakan input ke output yang diinginkan. Terdapat banyak metode yang ada dalam metode algoritma *supervised learning* diantaranya adalah diantaranya Regresi Logistik, *K-nearest Neighbor*, *Super Vector Machine*, *Naive Bayes*, *Decision Tree* dan *Random Forest*. Pada metode regresi yaitu *decision tree* atau pohon keputusan yang menurut penulis sesuai dengan kebutuhan dalam menyelesaikan studi kasus dengan memprediksi aktivitas berdasarkan preferensi pengguna yang ada di dalam gedung.

#### **2.4.1 Decision Tree**

Menurut Suyanto (2018), *Decision tree* dapat diartikan juga sebagai *top-down induction of decision trees* yaitu salah satu model supervised learning yang membangun representasi aturan klasifikasi berstruktur dengan cara mempartisi himpunan data latih secara rekursif. Teknik ini menghasilkan pohon keputusan yang berupa *n-ary branching tree* yang merepresentasikan suatu aturan klasifikasi. Beberapa teknik dalam *decision tree* yaitu *classification and regression tree (CART)*, *iterative dichotomiser (ID3)*, *C4.4 Quinlan*, *C4.5*, *C5.0 Quinlan*, *Cubist Quinlan*, *Assistant Cestnik*. *Decision tree* dapat digunakan untuk pengenalan pola dan termasuk dalam pengenalan pola secara statistik, *decision tree* dibentuk dari 3 tipe simpul :

1. Simpul leaf memuat suatu keputusan akhir atau kelas target untuk suatu *decision tree*.
2. Simpul root adalah titik awal dari *decision tree*.

3. Setiap simpul perantara berhubungan dengan suatu pertanyaan atau pengujian.



**Gambar 2.3.** Pola *Decision Tree*

Penulis melakukan klasifikasi yang diusulkan untuk dataset pelatihan (*training*) dengan menggunakan algoritma C4.5 yang berasal dari keluarga pohon keputusan (*decision Tree*).

#### **2.4.2 Algoritma C4.5**

Menurut Aldino Ramadhan dkk (2019), Algoritma C4.5 adalah salah satu algoritma yang ada di metode *machine learning*. Dengan algoritma ini, komputer akan diberikan data untuk dipelajari yang disebut *learning dataset*. Kemudian hasil dari pembelajaran selanjutnya untuk mengolah data-data yang baru yang disebut *tes dataset*. Karena algoritma C4.5 digunakan untuk melakukan klasifikasi, jadi hasil dari pengolahan *test dataset* berupa pengelompokkan data ke dalam kelas-kelas Umumnya, tahapan pada algoritma C4.5 yang digunakan untuk membentuk pohon keputusan (*Decision Tree*) adalah :

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

Untuk memilih atribut sebagai root, didasarkan pada nilai gain tertinggi dari atribut yang tersedia. Sementara itu, untuk mendapat nilai gain tertinggi kita harus menghitung nilai entropy dari semua nilai di dalam atribut. Entropy berperan sebagai parameter untuk mengukur varian dari data sampel. Setelah nilai entropy dalam data sampel diketahui, atribut yang paling berpengaruh akan menjadi pengukur dalam pengklasifikasian data, ukuran ini disebut sebagai Information gain.

Rumus menghitung entropy pada algoritma C4.5 :

$$\sum_{i=1}^k -p_i * \log_2 p_i$$

Keterangan :

- S adalah Himpunan (dataset) kasus.
- k adalah banyaknya partisi S.
- Pi adalah probabilitas yang didapat dari Sum (Ya) atau Sum (Tidak) dibagi total kasus.

Setelah mendapatkan entropy dari keseluruhan kasus, lakukan analisis pada setiap atribut dan nilai-nilainya dan hitung entropynya. Langkah berikutnya yaitu dengan menghitung Gain, rumus daripada Gain adalah sebagai berikut (Aldino Ramadhan dkk, 2019, p. 14):

$$Entropy(S) - \sum_{i=1}^k \frac{|S_i|}{|S|} \times Entropy(S_i)$$

- *Confusion Matrix*

*Confusion Matrix* adalah tool yang digunakan untuk evaluasi model klasifikasi untuk memperkirakan objek yang benar atau salah. Sebuah matrik dari prediksi yang akan dibandingkan dengan kelas yang asli dari inputan atau dengan kata lain berisi informasi nilai aktual dan prediksi pada klasifikasi (Aldino Ramadhan dkk, 2019, p. 14).

**Tabel 2.1** *Confusion Matrix* 2 Kelas

| Classification | Predicted Class         |                        |
|----------------|-------------------------|------------------------|
|                | Class = Yes             | Class = No             |
| Class = Yes    | a (True Positive-TP)    | b (False Negative-FN)  |
| Class = No     | c (False Positive - FP) | d (True Negative - TN) |

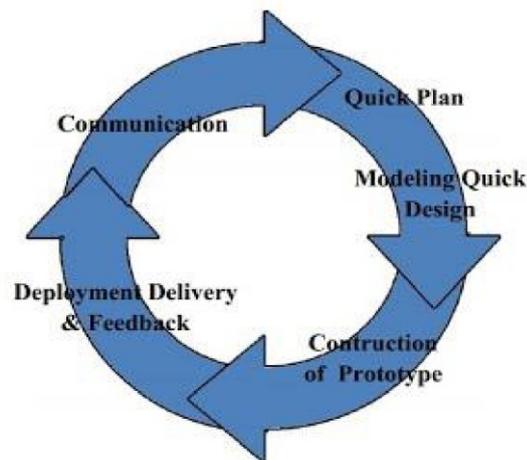
Rumus untuk menghitung tingkat akurasi pada matriks adalah:

$$Akurasi = \frac{TP + TN}{TP + FP + TN + FN} = \frac{a + d}{a + b + c + d} \times 100\%$$

## 2.5 Metode Pengembangan Perangkat Lunak

Penulis menggunakan metode *prototype*, yang merupakan metode pengembangan sistem dimana hasil analisa per bagian sistem langsung diterapkan ke dalam sebuah model tanpa menunggu seluruh sistem selesai. Dengan metode *prototype* ini akan dihasilkan *prototype* sistem sebagai perantara pengembang dan pengguna agar dapat berinteraksi dalam proses kegiatan pengembangan sistem.

Tahapan atau proses pembuatan *prototype* ini berhasil dengan mendefinisikan aturan-aturan yang ada pada tahap awal, yaitu pengembang dan pengguna harus satu pemahaman bahwa *prototype* dibangun untuk mendefinisikan kebutuhan awal. Metode *prototype* akan dihilangkan atau ditambahkan pada bagiannya sehingga sesuai dengan perencanaan dan analisis yang dilakukan pengembang sampai dengan uji coba dilakukan secara proses pengembangan, berikut 5 tahapan pada model *prototype* :



**Gambar 2.4.** Model *prototype*

Penjelasan dari gambar 2.4 diatas adalah sebagai berikut :

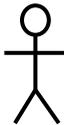
Tahap pertama adalah *communication* dan pengumpulan data awal yaitu tahap suatu perencanaan yang di lakukan, mulai dari menciptakan dan melaksanakan proses untuk memastikan bahwa perencanaan tersebut berkualitas tinggi, terpercaya, efisiensi biaya. Tahap kedua adalah *quick plan* yaitu analisis terhadap

kebutuhan pengguna. Tahap ketiga adalah *modelling quick design* yaitu pembuatan desain secara umum untuk selanjutnya dikembangkan kembali. Tahap keempat adalah *construction of prototype* adalah pembuatan perangkat *prototype* termasuk pengujian dan penyempurnaan. Tahap kelima adalah *deployment, delivery, and feedback* adalah tahap penyerahan sistem ke pengguna dan umpan balik.

### 2.5.1 Use Case Diagram

Menurut Rosa Tahun (2015) menyatakan, bahwa “*use case diagram* suatu pemodelan untuk mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara garis besar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

**Tabel 2.2** simbol-simbol use case diagram

| Simbol   | Deskripsi   |
|--|---|
| <p><i>use case</i></p>              | <p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau faktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i>.</p>      |
| <p>Aktor/<i>actor</i></p>           | <p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.</p> |
| <p>Asosiasi/<i>association</i></p>  | <p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>   |

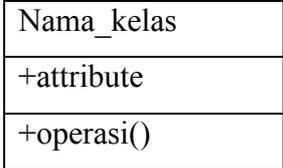
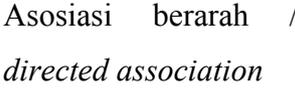
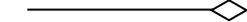
|   |   |
|---|---|
| <p>Ekstensi/<i>extend</i></p> <p>.....&gt;</p> <p>&lt;&lt;<i>extend</i>&gt;&gt;</p>                       | <p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dan <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek.</p> |
| <p>generalisasi/<i>generalizati</i><br/><i>on</i></p> <p>—————▶</p>                                       | <p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang lebih umum dari lainnya.</p>   |
| <p>menggunakan/<i>include/us</i><br/><i>es</i></p> <p>.....&gt;</p> <p>&lt;&lt;<i>include</i>&gt;&gt;</p> | <p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dan <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.</p>                              |

## 2.5.2 Class Diagram

Menurut rosa (2013) menyatakan, bahwa setiap *class* diagram merupakan gambaran dari struktur sistem dari segi pendefinisian kelas-kelas yang dibuat untuk membangun sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

- a. Kelas main, yaitu kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.
- b. Kelas yang menangani tampilan sistem, yaitu kelas yang mendefinisikan dan mengatur tampilan ke pemakai.
- c. Kelas yang diambil dari pendefinisian *use case*, yaitu kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*.
- d. Kelas yang diambil dari pendefinisian data, yaitu kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

**Tabel 2.3** simbol-simbol *class* diagram

| Simbol  | Deskripsi  |
|---|--|
| <p>Kelas</p>   | Kelas pada struktur sistem.  |
| <p>Antarmuka/<i>interface</i></p>                      | Sama dengan konsep interface dalam pemrograman berorientasi objek.   |
| <p>Asosiasi / <i>association</i></p>                   | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .   |
| <p>Asosiasi berarah / <i>directed association</i></p>  | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> . |
| <p>generalisasi</p>                                  | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status.   |
| <p>keberuntungan / <i>dependency</i></p>             | Relasi antar kelas dengan makna kebergantungan antar kelas.  |
| <p>Agregasi / <i>aggregation</i></p>                 | Relasi antar kelas dengan makna semua-bagian ( <i>whole-part</i> ).  |

Sumber : rosa a.s (2015:146)

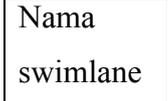
### 2.5.3 Activity Diagram

Menurut rosa (2011) menyatakan, bahwa *activity* diagram merupakan diagram aktivitas yang menggambarkan suatu sistem apa yang dilakukan actor, jadi aktivitas yang dapat dilakukan oleh sistem. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

- a. Rancangan proses bisnis dimana setiap urusan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.

- b. Urutan atau pengelompokan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
- c. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

**Tabel 2.4** simbol-simbol *activity diagram*

| Simbol  | Deskripsi  |
|---|--|
| Status awal<br>                  | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.        |
| aktivitas<br>                    | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.             |
| Percabangan/ <i>decision</i><br> | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.                    |
| Penggabungan/ <i>join</i><br>  | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.           |
| status akhir<br>               | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |
| <i>Swimlane</i><br>            | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.       |

Sumber : rosa a.s (2011:162)

## 2.6 Bahasa Pemrograman Yang Digunakan

Pada bahasa pemrograman penulis menggunakan bahasa pemrograman dengan *database* yang sering digunakan untuk menunjukkan semua program yang dapat dipakai dalam sistem komputer. Dalam istilah ini menunjuk pada sebuah program yang dapat mempermudah pemakai dari berbagai jenis komputer untuk mendayagunakan *hardware* dengan baik. Untuk merancang dan membangun sistem ini pembuat membutuhkan perangkat lunak yang dapat menunjang untuk memaksimalkan antara lain :

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

Menurut r.h sianipar (2015), html5 merupakan suatu markup buat menstrukturkan serta menunjukkan isi dari taman website. Html yang awal kali diciptakan pada tahun 1990 serta tipe keempatnya, html4, pada tahun 1997, serta sampai bulan juni 2011 senantiasa dalam proses pengembangan. Tujuan utama pengembangan html merupakan buat membetulkan teknologi html supaya menunjang teknologi multimedia terkini, gampang dibaca manusia serta pula gampang dipahami oleh mesin.

### **2.6.2 PHP (*Hypertext Preprocessor*)**

Php merupakan bahasa pemrograman yang digunakan secara luas buat penindakan pembuatan serta pengembangan suatu website serta dapat digunakan pada html. Php singkatan dari *hypertext preprocessor* yang digunakan bagaikan bahasa script server- side dalam pengembangan website yang disisipkan pada dokumen html. Pemakaian php membolehkan website bisa terbuat dinamis sehingga *maintenance* sistem website tersebut jadi lebih gampang serta efektif. Php adalah aplikasi *open-source* yang disebarkan serta dilisensikan secara *free* dan bisa didownload secara leluasa dari web resminya [http:// www.php.net](http://www.php.net)( peranganing, 2006).

Menurut edy winarno, ali zaki serta smitdev *community* (2014). Php (*hypertext preprocessor*) merupakan bahasa pemrograman website berbasis server (*server slide*) yang sanggup memparsing kode php dari kode dengan ekstensi php sehingga menciptakan tampilan web yang dinamis di sisi *client*.

### **2.6.3 Mysql**

Penelitian ini menggunakan *database* mysql dikarenakan *database* mysql merupakan sesuatu program *database* server yang sanggup menerima serta mengirimkan informasi dengan kilat memakai perintah-perintah sql. Mysql mempunyai dua wujud lisensi, ialah *free software* serta *freeware*. Mysql *free software* dibawah lisensi gnu/ gpl (*general public license*).

Menurut rachmad hakim s.Dab amp ir.Sutarto, meter.sang (2009). Database adalah mekanisme pengelolaan informasi dalam jumlah yang besar secara terstruktur. Database mempermudah program buat mengambil serta menaruh informasi. Bila informasi yang diolah banyak serta membutuhkan penindakan

spesial, jangan memakai file buat menaruh informasi. Namun, pakai database yang banyak diterapkan dikala ini merupakan database bertipe relasional (*relational database*), semacam oracle, microsoft sql server, mysql, dan lain-lain.

#### **2.6.4 Rapid Miner**

*Rapid miner* yaitu tools software dalam data ilmu pengetahuan yang dikembangkan oleh perusahaan dengan nama yang sama dapat menyediakan lingkungan terpadu untuk mempelajari mesin *machine learning*. Aplikasi ini dapat melakukan pelatihan pembuatan model *prototype* dengan cepat, dan pengembangan aplikasi serta mendukung semua langkah proses pembelajaran mesin termasuk persiapan data, visualisasi hasil, validasi dan pengoptimalan.

#### **2.7 Pengujian Black Box Testing**

Menurut rosa dan shalahuddin (2016) menyatakan bahwa, *blackbox testing* yaitu pengujian perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian ini dimaksudkan untuk memahami apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak atau *software* sesuai dengan spesifikasi yang dibutuhkan. Studi kasus yang dibuat untuk melakukan pengujian *black box testing* harus dibuat dengan kasus benar dan kasus salah.