

## BAB II

### LANDASAN TEORI

#### 2.1 Sistem Pakar

Menurut Budiharto & Suhartono (2016) “Sistem pakar merupakan salah satu dari beberapa domain masalah atau area dari *Artificial Intelligence* (AI) dan merupakan sebuah program computer pintar (*intelligent computer program*) yang memanfaatkan pengetahuan (*knowledge*) dan prosedur inferensi (*inference procedure*) untuk memecahkan masalah yang cukup sulit hingga membutuhkan keahlian khusus dari manusia”.

Menurut Rosnelly (2016) “Sistem pakar adalah sistem komputer yang ditujukan untuk meniru semua aspek (*emulates*) kemampuan pengambilan keputusan (*decision making*) seorang pakar. Sistem pakar memanfaatkan secara maksimal pengetahuan khusus selayaknya seorang pakar memecahkan masalah”.

Dalam buku Perancangan Sistem Pakar (2012:1), Nita dan Rahmat mengungkapkan beberapa definisi sistem pakar menurut beberapa ahli yaitu sebagai berikut:

1. Menurut Durkin: Sistem pakar adalah suatu program komputer yang dirancang untuk memodelkan kemampuan penyelesaian masalah yang dilakukan seorang pakar.
2. Menurut Ignizo: Sistem pakar adalah suatu model dan prosedur yang berkaitan dalam suatu domain tertentu, yang mana tingkat keahliannya dapat dibandingkan dengan keahlian seorang pakar.

3. Menurut Giarratano dan Riley: Sistem pakar adalah suatu sistem komputer yang bisa menyamai atau meniru kemampuan seorang pakar.

### 2.1.1 Tujuan Sistem Pakar

Sistem pakar adalah suatu sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang dilakukan oleh para ahli. Sistem pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja dan membandingkan tingkat keahliannya dari para ahli (Sri Hartati dan Sari Iswanti, 2010:2)

Sistem pakar termasuk kedalam kelompok kecerdasan buatan yang mempunyai kemampuan khusus untuk menyelesaikan kondisi permasalahan yang ada (Gede & Divayana, 2014). Selain itu, sistem pakar pernah diimplementasikan untuk menganalisa kerusakan *hardware* CISCO dengan menggunakan inferensi *Forward Chaining* (Widjaja & Susilo, 2017).

Tujuan dari sebuah sistem pakar adalah memindahkan kepakaran dari seorang pakar ke dalam komputer, kemudian ditransfer kepada orang lain yang bukan pakar (*nonexpert*). Proses ini melibatkan empat kegiatan, yaitu:

1. Akuisisi pengetahuan (dari pakar atau sumber lainnya)
2. Representasi pengetahuan (ke dalam komputer)
3. Inferensi pengetahuan
4. Pemindahan pengetahuan ke pengguna.

## 2.1.2 Kemampuan Sistem Pakar

### 2.1.2.1 Kelebihan Memakai Sistem Pakar

1. Menghimpun data dalam jumlah yang besar
2. Menyimpan data tersebut dalam jangka waktu yang lama dalam bentuk tertentu
3. Mengerjakan perhitungan secara cepat dan tepat serta mencari Kembali data yang tersimpan dengan kecepatan tinggi
4. Meningkatkan produktifitas
5. Membuat seorang yang awam bekerja seperti layaknya seorang pakar
6. Meningkatkan kualitas dengan memberi nasehat yang konsisten dan mengurangi kesalahan
7. Mampu menangkap pengetahuan dan kepakaran seseorang
8. Dapat beroperasi dilingkungan yang bahaya
9. Memudahkan akses pengetahuan seorang pakar
10. Andal
11. Meningkatkan kapabilitas sistem komputer
12. Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti, selama konsultasi dengan sistem pakar tetap akan memberikan jawabannya.
13. Bisa digunakan sebagai media pelengkap dalam pelatihan. Pengguna pemula yang bekerja dengan sistem pakar akan menjadi

lebih berpengalaman karena adanya fasilitas penjelas yang berfungsi sebagai guru

14. Meningkatkan kemampuan untuk menyelesaikan masalah karena mengambil dari sumber pengetahuan dari banyak pakar.

#### **2.1.2.2 Keuntungan Sistem Pakar**

1. Menjadikan pengetahuan lebih mudah didapat.
2. Meningkatkan *output* dan produktifitas.
3. Menyimpan kemampuan dan keahlian para pakar.
4. Meningkatkan penyelesaian permasalahan.
5. Meningkatkan realibitas
6. Memberikan *respons* (jawaban) yang cepat.
7. Merupakan panduan yang *intelegene* (cerdas)
8. Dapat bekerja dengan informasi yang kurang lengkap dan mengandung ketidakpastian
9. Dapat digunakan untuk mengakses basis data dengan cara cerdas

#### **2.1.2.3 Kelemahan Sistem Pakar**

1. Pengetahuan tidak selalu bisa didapat dengan mudah. Karena pendekatan yang dibuat oleh satu pakar dengan pakar lainnya berbeda.
2. Untuk membuat suatu sistem yang berkualitas sangat sulit dan memerlukan biaya yang tinggi.

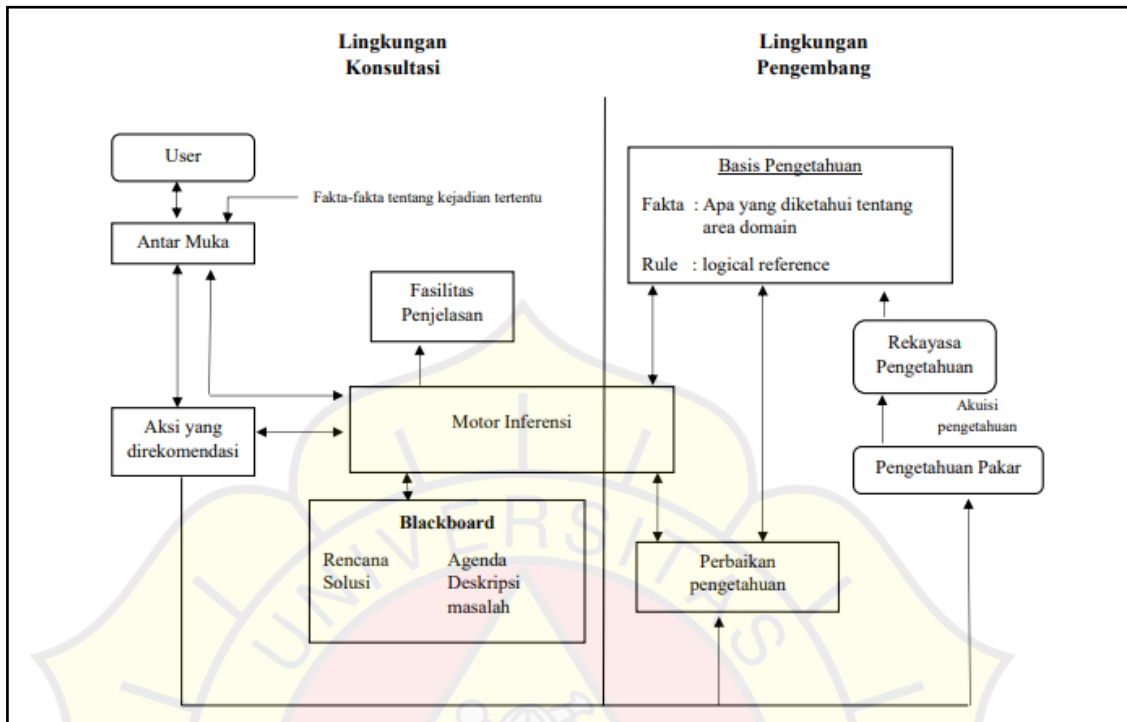
3. Sistem pakar tidak 100% benar, perlu diuji ulang sebelum digunakan. Dalam hal ini peranan manusia merupakan factor dominan.

#### **2.1.2.4 Ciri – Ciri Sistem Pakar**

1. Terbatas pada domain keahlian tertentu
2. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti
3. Dapat menjelaskan alasan-alasan dengan cara yang dapat dipahami
4. Bekerja berdasarkan pada kaidah atau *rule* tertentu
5. Mudah dimodifikasi
6. Basis pengetahuan dan mekanisme inferensi terpisah
7. Output bersifat anjuran
8. Sistem dapat mengaktifkan kaidah secara searah yang sesuai

#### **2.1.3 Struktur Sistem Pakar**

Ada dua bagian penting dari sistem pakar, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembang digunakan oleh pembuat sistem pakar untuk membangun komponen-komponennya dan memperkenalkan pengetahuan ke dalam *knowledge base*.



Gambar 2. 5 Struktur Sistem Pakar

Sumber : Buku Pengantar Sistem Pakar dan Metode. Z. Azmi & Verdi Yasin (2017).

Adapun penjelasan gambar 2.1 adalah sebagai berikut:

### 1. Akuisi Pengetahuan

Sub sistem ini digunakan untuk memasukkan pengetahuan dari seorang pakar dengan cara merekayasa pengetahuan agar bisa diproses oleh komputer dan meletakkannya ke dalam basis pengetahuan dengan format tertentu dalam bentuk representasi pengetahuan. Sumber-sumber pengetahuan bisa diperoleh dari pakar, buku, dokumen, multimedia, basis data, laporan riset khusus dan informasi yang terdapat diweb.

## **2. Basis Pengetahuan (*Knowledge Base*)**

Basis pengetahuan mengandung pengetahuan yang diperlukan untuk memahami, memformulasikan dan menyelesaikan masalah. Basis pengetahuan terdiri dari dua elemen dasar, yaitu:

- a. Fakta, misalnya situasi, kondisi atau permasalahan yang ada
- b. Rule (Aturan), untuk mengarahkan penggunaan pengetahuan dalam memecahkan masalah.

## **3. Mesin Inferensi (*Inference Engine*)**

Mesin inferensi adalah sebuah program yang berfungsi untuk memandu proses penalaran terhadap suatu kondisi berdasarkan pada basis pengetahuan yang ada, manipulasi dan mengarahkan kaidah, model, dan fakta yang disimpan dalam basis pengetahuan untuk mencapai solusi atau kesimpulan. Dalam prosesnya, mesin inferensi menggunakan strategi pengendalian, yaitu strategi yang berfungsi sebagai panduan arah dalam melakukan proses penalaran.

## **4. Daerah Kerja (*Blackboard*)**

Untuk merekam hasil sementara yang akan dijadikan sebagai keputusan dan untuk menjelaskan sebuah masalah yang sedang terjadi, sistem pakar membutuhkan *Blackboard*, yaitu area pada memori yang berfungsi sebagai basis data. Tiga tipe keputusan yang dapat direkam pada *Blackboard*, yaitu:

- a. Rencana: bagaimana menghadapi masalah
- b. Agenda: aksi-aksi potensial yang sedang menunggu untuk dieksekusi

c. Solusi: calon aksi yang akan dibangkitkan.

#### **5. Antarmuka Pemakai (*user interface*)**

Digunakan sebagai media komunikasi antara pengguna dan sistem pakar. Komunikasi disajikan dalam Bahasa alami (*natural language*) dan dilengkapi dengan grafik, menu dan formular elektronik. Pada bagian ini akan terjadi dialog antara sistem pakar dan pengguna.

#### **6. Sub Sistem Penjelasan (*Explanation Subsystem/Justifier*)**

Berfungsi memberi penjelasan kepada pengguna, kesimpulan dapat diambil. Kemampuan seperti ini sangat penting bagi pengguna untuk mengetahui proses pemindahan keahlian pakar maupun dalam pemecahan masalah.

#### **7. Sistem Perbaikan Pengetahuan (*Knowledge Refining System*)**

Kemampuan memperbaiki pengetahuan dari seorang pakar diperlukan untuk menganalisis pengetahuan, belajar dari kesalahan masa lalu, kemudian memperbaiki pengetahuannya sehingga dapat dipakai pada masa mendatang. Kemampuan evaluasi diri seperti itu diperlukan oleh program agar dapat menganalisis untuk mengambil kesimpulan.

#### **8. Pengguna (*User*)**

Pada umumnya pengguna sistem pakar bukanlah seorang pakar (*non-expert*) yang membutuhkan solusi, saran atau pelatihan (*training*) dari berbagai permasalahan yang ada.



### 2.1.4 Representasi Pengetahuan

Pengetahuan yang diperoleh melalui proses akuisisi pengetahuan tidak dapat diterapkan kedalam sistem pakar. Pengetahuan harus direpresentasikan dalam format tertentu dan akan dikompliasi menjadi suatu basis pengetahuan.

## 2.2 Metode

### 2.2.1 Metode Teorema Bayes (*Bayesian*)

Probabilitas Bayes merupakan salah satu cara yang baik untuk mengatasi ketidakpastian data dengan menggunakan formula bayes yang dinyatakan dengan

$$\text{rumus: } P(H | E) = \frac{P(E | H) \cdot (P(H))}{P(E)}$$

Keterangan:

$P(H | E)$  = Probabilitas hipotesis H jika diberikan

*Evidence* E

$P(E | H)$  = Probabilitas munculnya *evidence* apapun

$P(E)$  = Probabilitas *evidence* E

**Contoh kasus penyakit gagal ginjal akut:** (Sumber: Rahayu, 2013)

Aulia melakukan diagnose dengan menjawab pertanyaan sesuai dengan gejala berikut:

$$G1 = 0.4 = P(E|H1) \quad G5 = 0.2 = P(E|H5) \quad G9 = 0.4 = P(E|H9)$$

$$G2 = 0.2 = P(E|H2) \quad G6 = 0.2 = P(E|H6)$$

$$G3 = 0.4 = P(E|H3) \quad G7 = 0.2 = P(E|H7)$$

$$G4 = 0.6 = P(E|H4) \quad G8 = 0.4 = P(E|H8)$$

Kemudian mencari nilai semesta dengan menjumlahkan dari hipotesa di atas:

$$\begin{aligned} \sum_{k=1}^9 G_k &= G_1 + G_2 + G_3 + G_4 + G_5 + G_6 + G_7 + G_8 + G_9 \\ &= 0.4 + 0.2 + 0.4 + 0.6 + 0.2 + 0.2 + 0.2 + 0.4 + 0.4 \\ &= 3 \end{aligned}$$

Setelah hasil penjumlahan di atas diketahui, maka didapatkan rumus untuk menghitung nilai semesta adalah sebagai berikut:

$$P(H_1) = H_1 = 0.4 = 0.133333 \quad P(H_6) = H_6 = 0.2 = 0.066667$$

$$\sum_{k=1}^3 9 \quad \sum_{k=1}^3 9$$

$$P(H_2) = H_2 = 0.2 = 0.066667 \quad P(H_7) = H_7 = 0.2 = 0.066667$$

$$\sum_{k=1}^3 9 \quad \sum_{k=1}^3 9$$

$$P(H_3) = H_3 = 0.4 = 0.133333 \quad P(H_8) = H_8 = 0.4 = 0.133333$$

$$\sum_{k=1}^3 9 \quad \sum_{k=1}^3 9$$

$$P(H_4) = H_4 = 0.6 = 0.200000 \quad P(H_9) = H_9 = 0.4 = 0.133333$$

$$\sum_{k=1}^3 9 \quad \sum_{k=1}^3 9$$

$$P(H_5) = H_5 = 0.2 = 0.066667$$

$$\sum_{k=1}^3 9$$

Setelah nilai  $P(H_i)$  didapat, probabilitas hipotesis  $H$  tanpa memandang *evidence* apapun, maka langkah berikut adalah:

$$\begin{aligned} \sum_{k=1}^9 P(H_k) &= P(H_1) * P(E|H_1) + P(H_2) * P(E|H_2) + P(H_3) * P(E|H_3) + P(H_4) * \\ &P(E|H_4) + P(H_5) * P(E|H_5) + P(H_6) * P(E|H_6) + P(H_7) * P(E|H_7) \\ &+ P(H_8) * P(E|H_8) + P(H_9) * P(E|H_9) \end{aligned}$$

$$\begin{aligned}
&= (0.133333 \cdot 0.4) + (0.066667 \cdot 0.2) + (0.133333 \cdot 0.4) + (0.200000 \cdot 0.6) \\
&+ (0.066667 \cdot 0.2) + (0.066667 \cdot 0.2) + (0.066667 \cdot 0.2) + (0.133333 \cdot 0.4) \\
&+ (0.133333 \cdot 0.4) = 0.053333 + 0.013333 + 0.053333 + 0.12 + 0.013333 + \\
&0.013333 + 0.013333 + 0.053333 + 0.053333 = 0.386667
\end{aligned}$$

Langkah selanjutnya ialah mencari nilai  $P(H_i|E)$  atau probabilitas hipotesis

$H_i$  benar jika diberikan *evidence*  $E$

$$P(H_1|E) = \frac{0.133333 \cdot 0.4}{0.386667} = 0.137931 \qquad P(H_5|E) = \frac{0.034483 \cdot 0.2}{0.386667} = 0.034483$$

$$P(H_2|E) = \frac{0.034483 \cdot 0.2}{0.386667} = 0.034483 \qquad P(H_6|E) = \frac{0.034483 \cdot 0.2}{0.386667} = 0.034483$$

$$P(H_3|E) = \frac{0.133333 \cdot 0.4}{0.386667} = 0.137931 \qquad P(H_7|E) = \frac{0.034483 \cdot 0.2}{0.386667} = 0.034483$$

$$P(H_4|E) = \frac{0.310345 \cdot 0.6}{0.386667} = 0.310345 \qquad P(H_8|E) = \frac{0.133333 \cdot 0.4}{0.386667} = 0.137931$$

$$P(H_9|E) = \frac{0.133333 \cdot 0.4}{0.386667} = 0.137931$$

Setelah seluruh nilai  $P(H_i|E)$  diketahui, maka jumlahkan seluruh nilai bayesnya dengan rumus sebagai berikut:

$$\begin{aligned}
\sum_{k=1}^9 \text{Bayes } k &= \text{Bayes 1} + \text{Bayes 2} + \text{Bayes 3} + \text{Bayes 4} + \text{Bayes 5} + \text{Bayes 6} + \text{Bayes 7} \\
&+ \text{Bayes 8} + \text{Bayes 9}
\end{aligned}$$

$$\begin{aligned}
&= (0.137931 \cdot 0.4) + (0.034483 \cdot 0.2) + (0.137931 \cdot 0.4) + (0.310345 \cdot 0.6) \\
&+ (0.034483 \cdot 0.2) + (0.034483 \cdot 0.2) + (0.034483 \cdot 0.2) + (0.137931 \cdot 0.4) \\
&+ (0.137931 \cdot 0.4) \\
&= 0.434488 = 43,44828
\end{aligned}$$

### 2.2.2 Metode FP-Growth

Frequent Pattern Growth (FP-Growth) adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*frequent itemset*) dalam sebuah kumpulan data. Algoritma FP-Growth merupakan pengembangan dari algoritma Apriori. Sehingga kekurangan dari algoritma Apriori diperbaiki oleh algoritma FP-Growth.

FP-Growth menggunakan konsep pembangunan tree dalam pencarian *frequent itemset*. Hal tersebutlah yang menyebabkan algoritma FP-Growth lebih cepat dari algoritma Apriori. Karakteristik algoritma FP-Growth adalah struktur data yang digunakan adalah tree yang disebut dengan FP-Tree.

FP-Tree merupakan struktur penyimpanan data yang dimampatkan. FP-Tree dibangun dengan memetakan setiap data transaksi ke dalam setiap lintasan tertentu dalam FP-tree. Karena dalam setiap transaksi yang dipetakan, mungkin ada transaksi yang memiliki item yang sama, maka lintasannya memungkinkan untuk saling menimpa. Semakin banyak data transaksi yang memiliki item yang sama, maka proses pemampatan dengan struktur data FP-tree semakin efektif. Metode FP-Growth dapat dibagi menjadi 3 tahapan utama yaitu sebagai berikut:

#### A. Tahap Pembangkitan Conditional Pattern Base

*Conditional Pattern Base* merupakan subdatabase yang berisi *prefix path* (lintasan prefix) dan *suffix pattern* (pola akhiran). Pembangkitan *conditional pattern base* didapatkan melalui FP-tree yang telah dibangun sebelumnya.

## B. Tahap Pembangkitan Conditional FP-Tree

Pada tahap ini, *support count* dari setiap item pada setiap *conditional pattern base* dijumlahkan, lalu setiap item yang memiliki jumlah *support count* lebih besar sama dengan minimum *support count*  $\xi$  akan dibangkitkan dengan *conditional FP-tree*.

## C. Tahap Pencarian Frequent-Itemset

Apabila *Conditional FP-tree* merupakan lintasan tunggal (*single path*), maka didapatkan *frequent itemset* dengan melakukan kombinasi item untuk setiap *conditional FP-tree*. Jika bukan lintasan tunggal, maka dilakukan pembangkitan *FP-Growth* secara rekursif.

## 2.3 Kerusakan Kendaraan Bermotor

Terkadang kendaraan bermotor juga menunjukkan gejala indikasi bahwa kendaraan tidak dalam kondisi kurang baik. Ketika pada kendaraan bermotor mengalami adanya penurunan performa yang berarti mesin pada kendaraan bermotor tidak dapat bekerja secara maksimal. Ada beberapa penyebab sehingga terjadinya kerusakan pada mesin kendaraan bermotor. Jika penyebab kerusakan pada kendaraan bermotor tidak segera ditangani atau diperbaiki dengan baik, akan menyebabkan kerusakan total yang lebih parah lagi nantinya.

### **2.3.1 Masalah Utama Pada Kendaraan Bermotor dan Penyebabnya**

Gejala-gejala kerusakan pada kendaraan bermotor pada umumnya dapat dirincikan sebagai berikut:

1. Aki soak

Aki berfungsi sebagai penyimpanan arus pada motor. Aki ini sangat dibutuhkan pada saat proses starting, arus dari baterai akan disalurkan ke motor stater untuk memutar poros engkol mesin. Aki soak ini terjadi saat aki tidak mampu lagi menyimpan arus listrik dengan jumlah yang besar. Beberapa penyebabnya, antara lain motor jarang digunakan atau dipanaskan, aksesoris pada kendaraan yang berlebihan, usia pemakaian aki yang sudah tua.

2. Mesin brebet

Mesin brebet adalah kondisi dimana suara mesin tidak stabil seperti kehabisan bensin. Hal ini terjadi karena adanya misfire pada beberapa siklus mesin yang terjadi. Penyebab pada kondisi ini adalah pilot jet mampet yang membuat suplay bahan bakar berkurang, busi kotor yang menyebabkan api pada busi tidak memercikan bunga api dengan maksimal, api pada busi terlalu kecil bisa juga terjadi karena kerusakan pada busi motor.

3. Tenaga hilang/ngempos

Motor ngempos juga sering dikeluhkan oleh pemilik motor. Masalah ini lebih sering dialami ketika motor kita sedang dalam

posisi menanjak. Factor usia menjadi penyebab utama berkurangnya power dan akselerasi pada kendaraan bermotor. Selain factor usia ada beberapa factor lainnya yang membuat tenaga motor hilang, seperti kondisi filter yang sudah kotor yang dapat menyebabkan aliran udara selalu terhambat karena adanya banyak kotoran yang mengendap pada permukaan filter (saringan udara), selain itu motor ngempos juga bisa disebabkan karena tekanan kompresi rendah. Kalau ini, disebabkan oleh keausan pada salah satu komponen antara ring piston dan blok silinder.

#### 4. Mesin susah hidup saat pagi hari

Ada beberapa factor yang menjadi penyebab mesin kendaraan bermotor sulit dihidupkan pada saat mesin dingin. Beberapa penyebabnya, antara lain tegangan arus listrik baterai yang berkurang atau bahkan habis (soak), campuran bensin terlalu kurus (lean AFM), kompresi mesin yang rendah, fuel pump lemah (biasanya terjadi pada motor injeksi), api busi terlalu kecil.

#### 5. Suara mesin menggelitik (*Knocking*)

Mesin yang knocking atau menggelitik juga kerap banyak dikeluhkan oleh para pemilik kendaraan bermotor, terutama saat menambah kecepatan atau akselerasi. Ada beberapa penyebab. Factor penyebab yang paling sering adalah spesifikasi bahan bakar yang tidak sesuai, tingkat Research Octane Number (RON),

alhasil, timing pengapian tidak tepat. Istilah bengkelnya dikendal karena adanya prematur ignition yaitu proses pembakaran diruang bakar antara bahan bakar dan udara dengan percikan api tidak sempurna. Sehingga ada ledakan prematur yang disebut knocking.

#### 6. Keluar asap putih

Asap yang dimaksud adalah asap putih yang keluar dari exhaust (pembuangan) kendaraan bermotor pada mesin 4tak. Asap ini berasal dari oli pada mesin yang ikut terbakar di dalam ruang bakar. Seharusnya, oli mesin memang tidak masuk kedalam ruang bakar, namun ini bisa terjadi saat ring piston mengalami keausan. Penyebab lainnya yaitu keausan blok silinder dan linier mesin, batang valve (klep) bengkok, pemasangan katup (klep) tidak benar, paking pada head silinder rusak.

#### 7. Mogok secara tiba-tiba

Motor mogok juga sering dialami oleh banyak pemilik motor. Penyebabnya bukan hanya karena kehabisan bahan bakar bensin, tetapi masih ada penyebab lainnya seperti rusak nya pada CDI (Capacitor Discharge Ignition).

CDI adalah komponen pengapian sepeda motor yang dipakai untuk mengaktifkan induksi pada ignition coil. Apabila CDI mengalami kerusakan maka induksi pada coil juga tidak terjadi sehingga busi tidak mengeluarkan percikan bunga api.



#### 8. Knalpot nembak

Knalpot (exhaust) nembak adalah suara seperti letupan yang keluar dari dalam knalpot. Suara ini tidak hanya mengganggu tetapi terkadang juga mengagetkan pengendara motor itu sendiri. Factor penyebabnya adalah karena campuran bensin terlalu banyak, ketika mesin berada pada RPM tinggi maka Gerakan piston berlangsung cepat. Saat campuran bensin ini terlalu banyak, ada sejumlah bensin yang tidak terbakar. Dan saat bensin itu keluar menuju knalpot. Bensin terbakar oleh temperature knalpot yang tinggi, penyebab lainnya karena busi kotor, ini membuat api yang keluar dari busi sangat kecil sehingga ketika RPM tinggi pembakaran tidak berlangsung secara menyeluruh. Dan saat langkah buang, bensin terbakar didalam knalpot akibat temperature tinggi pada knalpot dan menyisakan suara tembakan.

#### 9. Komstir oblak

Komstir atau steering adalah komponen paling penting bagi kenyamanan berkendara sepeda motor, maupun mobil. Fungsi komstir yaitu menyeimbangkan dan menstabilkan stang motor serta meminimalisir getaran pada kecepatan tinggi. Penyebab komstir oblak adalah karena factor pemakaian (usia yang terlalu tua), motor yang sering digunakan pada jalan berlubang atau penuh bebatuan atau sering menghamtam lobang dengan

kecepatan tinggi, menyebabkan adanya tekanan yang berlebihan pada area fork suspense teleskopik.

#### 10. Rem cakram blong

Rem cakram atau disc brake adalah salah satu sistem pengereman pada motor yang konsep kerjanya memanfaatkan komponen tambahan berupa disc rotor (piringan) yang dijepitkan oleh dua buah kampas rem, agar bisa memperlambat laju kendaraan.

Penyebab biasanya yang terjadi pada rem cakram yang blong adalah kampas rem cakram yang sudah aus (tipis), penggunaan minyak rem yang tidak sesuai anjuran pabrik, udara yang masuk (masuk angin) adalah udara yang masuk dan terperangkap di sistem pengereman. Sehingga ada ruang kosong pada jalur minyak rem dari tuas (handle) menuju kaliper. Normalnya, sistem rem hanya boleh terisi dengan minyak rem sehingga tekanan tuas rem ke kaliper tersampaikan dengan baik.

#### 11. Ban bocor

Ban bocor pada kendaraan bermotor dapat mengganggu kita sebagai pengendara motor. Meski sekarang banyak yang menggunakan ban tubless, tetapi tetap saja ban tubless juga bisa bocor hanya saja keluar anginnya lebih lambat. Penyebab terjadinya ban bocor ialah pemakaian yang terlalu lama dan

keausan pada karet ban, sehingga menyebabkan ban tipis dan gampang sobek (bocor).

### **2.3.2 Gejala Kerusakan Pada Kendaraan Bermotor**

Pada usia penggunaan motor dan waktu yang tertentu motor akan mengalami penurunan kondisi. Hal ini tentu berkaitan dengan tingkat kerusakan komponen baik akibat pemakaian, kondisi dari pengguna maupun usia pemakaian itu sendiri. Namun dengan jumlah yang sangat banyak, sangat sulit jika kita harus mengecek kondisi tiap komponen. Beberapa penyakit pada motor memang bisa dideteksi langsung dengan mata. Namun banyak juga gejala kerusakan yang muncul ketika melakukan test drive (tes jalan).

Dengan melakukan test drive, maka bisa diketahui kerusakan apa saja yang terjadi. Ada beberapa kerusakan yang biasa terjadi pada kebanyakan kendaraan motor bekas. Dengan mengenali gejala penyakit umum ini, tentu bisa dijadikan pertimbangan sekaligus persiapkan dana untuk perbaikan jika ada yang rusak.

1. Gejala kerusakan: Jumlah Pelumas (*engine oil*) yang tersisa dalam mesin sedikit. Kemungkinan penyebab nya adalah:
  - a. Seal oli (gasket) mengalami kebocoran
  - b. Bosh katup (seal katup/klep bocor)
  - c. Pelumas mesin tidak sesuai anjuran pabrik.
2. Gejala kerusakan: Susah dihidupkan saat di stater. Kemungkinan penyebab nya adalah:
  - a. Bahan bakar yang kotor

- b. Kerja choke tidak benar (tidak sesuai setelan pabrik)
  - c. Putaran stasioner yang terlalu rendah
  - d. Penyetelan skrup udara pada karburator tidak tepat.
  - e. Insulator pada karburator terdapat kebocoran.
3. Gejala kerusakan: Suara mesin yang berisik. Kemungkinan penyebabnya adalah:
- a. Penyetelan katup (klep) yang tidak tepat atau tidak sesuai dengan buku panduan ketentuan motor.
  - b. Camshaft dan Rocker arm yang sudah aus
  - c. Sistem tensioner rantai mesin rusak (aus)
4. Gejala kerusakan: Mesin tidak dapat stasioner. Kemungkinan penyebabnya adalah:
- a. Katup menutup tidak rapat, perlu nya skir katup ulang agar katup dapat menutup dengan rapat
  - b. Penyetelan katup yang tidak tepat.
  - c. Penyetelan udara pada karburator yang tidak sesuai atau kebocoran pada insulator karburator
5. Gejala kerusakan: Tekanan Kompresi rendah. kemungkinan penyebabnya adalah:
- a. Penyetelan pembukaan pada katup in/ex tidak tepat atau tidak sesuai standar pabrik
  - b. Katup (klep) aus atau bengkok

- c. Pegas pada katup (klep) patah
  - d. Piston dan ring piston rusak (aus)
6. Gejala kerusakan: Roda depan goyang, kemungkinan penyebabnya adalah:
- a. Lingkar pada roda atau velg berubah bentuk
  - b. Bearing atau laher roda sudah rusak
  - c. Poros (baut) as roda tidak dikencangkan dengan baik
7. Gejala kerusakan: Daya pengereman pada rem cakram kurang (los), kemungkinan penyebabnya adalah:
- a. Piringan (disc) cakram terdapat minyak atau oli
  - b. Disc pad atau kampas rem kotor
  - c. Kaliper dan piston rem kotor
8. Gejala kerusakan: Lampu sein mati, kemungkinan penyebabnya adalah:
- a. Flasher rusak (tidak bekerja normal)
  - b. Aki (battery) lemah
  - c. Rangkaian kelistrikan pada lampu sein putus
  - d. Bohlam lampu sein putus

#### **2.4 FP-Growth (Frequent Pattern Growth)**

*FP-Growth (Frequent Pattern Growth)* adalah algoritma untuk menemukan frequent itemset tanpa melalui proses penggenerasian kandidat seperti pada algoritma apriori. Algoritma *FP Growth* merupakan pengembangan dari algoritma

apriori yang mengadopsi prinsip divide-and-conquer yaitu dengan cara mengkompresi database dan merepresentasikan requent itemset ke dalam bentuk Frequent Pattern Tree (FP-Tree) yang menyimpan informasi assosiasi antar itemset. Kemudian membagi database terkompresi tersebut kedalam suatu set conditional dataset.

Adapun langkah-langkah dari Algoritma FP-Growth adalah sebagai berikut:

1. Scan database dengan cara yang sama seperti algoritma apriori
2. Langkah berikutnya adalah pembentukan FP-Tree. FP Tree dibuat dengan cara seperti berikut:
  - a. Buat Root dari Tree dan di label dengan “null”
  - b. Item pada setiap transaksi diproses dalam bentuk list order, ★ selanjutnya pembentukan simpul untuk setiap transaksi
3. Menambang FP-Tree dengan cara sebagai berikut:
  - a. Membentuk conditional Pattern Base (“sub database” yang terdiri dari suatu set prefix path didalam FP-Tree yang terbentuk bersama suffix pattern)
  - b. Membentuk conditional FP-Tree
  - c. Frequently Pattern Growth di dapat dari penggabungan antara suffix pattern dengan Frequently Pattern yang dihasilkan dari Conditional FP-Tree.

## 2.5 Basis Data (Database)

Asal kata basis data (*database*) adalah basis (*base*) dan data (*data*). Menurut Fred R. McFadden et al (1999) dalam *Modern Database Management* menyebutkan bahwa data adalah fakta-fakta tentang segala sesuatu di dunia nyata yang dapat direkam dan disimpan pada media komputer.

Menurut Jefferey A. Hoffer, Mary B. Prescott dan Fred R. McFadden (2005:5) data merupakan gambaran objek peristiwa yang mempunyai arti dan penting di lingkungan pemakai. Menurut Kamus Besar Bahasa Indonesia (KBBI) basis adalah asas, dasar, himpunan, pangkalan, kumpulan

Menurut Raghu Ramakrishnan dan Johanes Gehrke (2003:3), basis data adalah sekumpulan data, umumnya mendeskripsikan aktivitas suatu organisasi yang berhubungan atau lebih. Misalnya, database universitas mungkin berisi informasi mengenai hal berikut:

- a. Entitas seperti mahasiswa, fakultas, mata kuliah dan ruang kuliah.
- b. Hubungan antara entitas, seperti registrasi mahasiswa dalam mata kuliah, fakultas yang mengajarkan mata kuliah dan penggunaan ruang untuk kuliah.

Tujuan umum dalam merancang penyimpanan data adalah:

1. Meyakini pengambilan Kembali data tujuan
2. Menyediakan penyimpanan data yang efisien
3. Ketersediaan data
4. Mendukung pengambilan data yang efisien.

## 5. Menjamin integritas data. (Kendall, 2010:25)

Secara sederhana basis data dapat diungkapkan sebagai suatu pengorganisasian data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan cepat. Dalam hal ini, pengertian akses dapat mencakup pemerolehan data maupun manipulasi data seperti menambah serta menghapus data.

Sebagaimana diketahui, manajemen modern mengikutsertakan informasi sebagai sumber daya penting yang setara dengan sumber daya manusia, uang, mesin dan material. Menurut Hoffer, Prescott dan McFadden (2005: 5) informasi merupakan data yang telah diproses dengan cara meningkatkan pengetahuan seseorang yang menggunakan data tersebut. Informasi merupakan bentuk penyajian data melalui mekanisme pemrosesan, berguna bagi pihak tertentu, misalnya seorang manajer. Bagi pihak manajemen, informasi merupakan bahan untuk pengambilan keputusan.

Dengan adanya komputer, data dapat disimpan dalam media penyimpanan yang disebut *hard disk*. Dengan menggunakan media ini, kehadiran kertas yang digunakan untuk menyimpan data dapat dikurangi. Selain itu, data menjadi lebih cepat untuk diakses terutama kalau dikemas dalam bentuk *database*.

## 2.6 PHP (Hypertext Preprocessor)

PHP telah menjadi Bahasa pemrograman web yang digunakan secara luas untuk membuat halaman web yang dinamis. Dalam hal ini diperkuat oleh Solichin (2016) yang menyatakan bahwa PHP menjadi salah satu Bahasa pemrograman yang digunakan sebagai pengembang website. PHP awalnya mulai dikembangkan pada



akhir tahun 1994 oleh Rasmus Lerdorf, namun sekarang di ambil oleh *The PHP Group*. Semula PHP merupakan singkatan dari *Personal Home Page*, namun dalam perkembangannya diubah menjadi (*Hypertext Preprocessor*). Menurut Kadir (2017) ciri dari Bahasa pemrograman Interpreter adalah bekerja menerjemahkan instruksi pada saat program mulai di eksekusi. Dengan kian banyaknya programmer atau peminat dalam menggunakan Bahasa pemrograman PHP, hingga PHP terus berkembang dari masa ke masa yang dimanfaatkan sesuai dengan kebutuhan pengguna dan hingga kini PHP telah masuk pada versi 7.0, berikut table perkembangan PHP dari masa ke masa:

*Tabel 2. 1 Perkembangan PHP*

<b>Versi PHP</b>	<b>Tahun</b>
1. PHP 1.0	1994
2. PHP 2.0	1996
3. PHP 3.0	1998
4. PHP 4.0	2000
5. PHP 5.0	2004
6. PHP 6.0	2005
7. PHP 7.0	2015

PHP juga dikenal dengan Bahasa pemrograman *open source* yang berarti dapat digunakan secara gratis, PHP juga dapat disatukan dengan Bahasa HTML. Prasetio (2012) menyatakan bahwa PHP merupakan Bahasa pemrograman yang memiliki kemampuan dalam memisahkan kode PHP dari HTML

## 2.7 Website

### 2.7.1 Pengertian Website

Website merupakan sebuah media yang memiliki banyak halaman yang terhubung (*Hyperlink*), dimana website memiliki fungsi dalam memberikan informasi berupa teks, gambar, video, suara dan animasi atau penggabungan dari semuanya. Website pada saat sekarang ini umumnya telah bersifat dinamis, meskipun dahulu juga ada website yang bersifat statis, namun website statis telah jarang dan bahkan hamper tidak ada lagi ditemukan. Karakteristik utama yang dimiliki oleh website adalah halaman-halaman yang saling terhubung, dan dilengkapi dengan domain sebagai alamat (url) atau *World Wide Web* (www) dan juga hosting sebagai media yang menyimpan banyak data. Website dapat diakses menggunakan jaringan internet dengan platform yang disebut browser, seperti chrome, mozilla firefox, internet explorer (IE), opera dan sebagainya.

Website dapat dibangun dalam mode localhost, yang artinya website dapat dirancang, dibangun dan dimodifikasi tanpa menggunakan jaringan internet. Dalam pembangunan sebuah website pada mode publikasi ke internet ada beberapa aplikasi yang dibutuhkan, diantaranya adalah database (MySQL, Oracle) *etc*, Web Server Apache, PHP Editor (Macromedia, Notepad++) *etc*, dan browser. Website atau aplikasi berbasis web umumnya dibangun menggunakan Bahasa pemrograman seperti *Hypertext Preprocessor* (PHP) dan *Active Server Pages* (ASP) , yang dikombinasikan dengan *Hypertext Markup Language* (HTML), Cascading Style Sheet (CSS), dan Javascript. Pada tingkatan *expert* seorang programmer juga

membutuhkan aplikasi tambahan lain seperti Photoshop, Coreldraw dan lain sebagainya. Setelah aplikasi tersebut di instalasi pada komputer maka barulah dilakukan proses perancangan, desain dan pengkodean (coding) terhadap website yang akan dibangun.

## **2.8 UML**

### **2.8.1 Pengertian UML**

UML (*Unified Modelling Language*) adalah sebuah Teknik pengembangan sistem yang menggunakan Bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem. UML pertama kali dipopulerkan oleh Grady Booch dan James Rumbaugh pada tahun 1994 untuk mengkombinasikan dua metodologi terkenal yaitu Booch dan OMT, kemudian Ivar Jacobson yang menciptakan *Object Oriented Software Engineering* (OOSE) ikut bergabung standar UML yang dikelola oleh *Object Management Group* (OMG).

Dari penelitian (Rizkita et al., 2018), bahwa UML atau biasa dibilang *Unified Modelling Language* merupakan Teknik yang dapat mengembangkan sistem dengan menggunakan salah satu Bahasa yaitu Bahasa grafis sebagai alat pendokumentasian dan juga dalam melakukan spesifikasi sistem. UML memiliki banyak diagram dan diagram itu digunakan untuk melakukan pemodelan data maupun sistem.

### **2.8.2 Use Case Diagram**

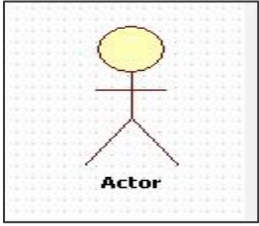
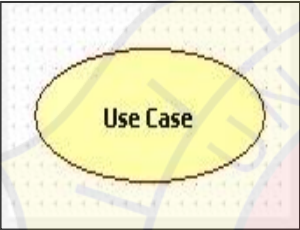
Alasan paling umum beberapa proyek pengembangan perangkat lunak dinyatakan gagal adalah kurangnya komunikasi antara pengembang perangkat

lunak dengan *stakeholder* kunci. Komunikasi yang kurang memadai dengan *stakeholder* kunci menyebabkan kurangnya *alignment* antara pengembang perangkat lunak dengan jenis usaha organisasi. Pelaku bisnis umumnya mengetahui bahwa mereka memiliki masalah tertentu yang perlu diselesaikan. Namun pengembangan organisasi bisa jadi hanya memiliki gambaran umum tentang apa yang diinginkan oleh bisnis organisasi, dengan beberapa persyaratan tertentu.

Menurut (Setiawan & Khairuzzaman, 2017) “Diagram *Use Case* menyajikan interaksi antara *Use Case* dan *actor*. Dimana *actor* dapat berupa orang, peralatan atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun.

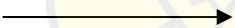

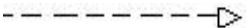
*Use case diagram* erat kaitannya dengan kejadian-kejadian (*scenario*). Kejadian merupakan contoh apa yang terjadi ketika *user* atau seseorang berinteraksi dengan sistem. *Use case diagram* berguna dalam tiga hal, yaitu:

Tabel 2. 2 Komponen Use Case

Komponen <i>Use Case</i>	Penjelasan
	<p>Merupakan sebuah komponen yang menggambarkan seseorang atau sesuatu (seperti perangkat atau sistem lainnya) yang berinteraksi dengan sistem.</p>
	<p><i>Use case</i> adalah gambaran fungsionalitas dari suatu sistem, sehingga pengguna sistem paham atau mengerti mengenai kegunaan sistem yang akan dibangun.</p>

(Sumber : Rosa dan Shalahuddin , 2014:156)

Tabel 2. 3 Relasi Use Case

Relasi <i>use case</i>	Penjelasan
 <p>Relasi <i>Association</i></p>	<p><i>Association</i>, menghubungkan link antara element.</p>
 <p>Relasi <i>Generalization</i></p>	<p><i>Generalization</i> disebut juga <i>inheritance</i> (pewarisan), sebuah elemen dapat merupakan spesialisasi dari elemen lainnya.</p>
 <p>Relasi <i>Dependency</i></p>	<p><i>Dependency</i>, sebuah elemen bergantung dalam beberapa cara ke elemen lainnya.</p>

Tabel 2. 4 Stereotype Use Case

<b>Relasi / Stereotype</b>	<b>Penjelasan</b>
<b>&lt;&lt;include&gt;&gt;</b>	Kelakuan yang harus terpenuhi agar sebuah <i>event</i> dapat terjadi, dimana pada kondisi ini sebuah <i>use case</i> adalah bagian dari <i>use case</i> lainnya.
<b>&lt;&lt;extends&gt;&gt;</b>	Kelakuan yang hanya berjalan dibawah kondisi tertentu seperti menggerakkan alarm.
<b>&lt;&lt;Communicates&gt;&gt;</b>	Ditambahkan untuk asosiasi yang mungkin menunjukkan asosiasinya adalah <i>communicates association</i> . Ini merupakan pilihan selama asosiasi hanya tipe <i>relationship</i> yang dibolehkan antara <i>actor</i> dan <i>use case</i> .

(Sumber : Rosa dan Shalahuddin , 2014:156)

### 2.8.3 Activity Diagram



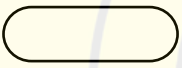
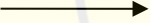


Menurut Sucipto (2011:208) diagram aktivitas menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainnya seperti *use case* atau interaksi. Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Dipakai pada *business modeling* untuk memperlihatkan urutan aktivitas proses bisnis. Struktur diagram ini mirip *flowchart* atau data *flow* diagram pada perancangan terstruktur simbol dalam *activity* diagram.

Menurut (Irmayani & Susyatih, 2017) “*Activity* diagram menggambarkan aktivitas utama dari user pada sistem informasi yang dibuat”. Disimpulkan dari dua


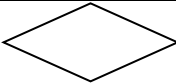

penjelasan diatas bahwa *Activity* diagram Teknik untuk mendeskripsikan logika procedural, proses bisnis dan aliran kerja dalam banyak kasus.

Berikut adalah simbol-simbol yang terdapat pada *Activity* diagram:

Tabel 2. 5 Simbol – Simbol *Activity* Diagram

<i>Activity Diagram</i>	Penjelasan
 <i>Start State</i>	<i>Start State</i> , sebagai tanda awal proses dari <i>activity</i> diagram.
 <i>State</i>	State, berfungsi menampung <i>event</i> dalam <i>activity</i> diagram.
 <i>Activity</i>	<i>Activity</i> , memiliki fungsi yang sama dengan <i>state</i> . Menampung <i>event</i> atau aktifitas pada proses sistem.
 <i>State Transition</i>	<i>State Transition</i> , berfungsi untuk menunjukkan aliran atau urutan dari <i>event</i> atau aktifitas pada diagram.
 <i>Transition to self</i>	<i>Transition to self</i> , berfungsi untuk menunjukkan transisi sebuah <i>event</i> yang mengarah ke <i>event</i> itu sendiri.
 <i>Horizontal Synchronization</i>	<i>Horizontal Synchronization</i> , berfungsi untuk mensinkronisasikan 2 cabang <i>event</i> yang posisinya horizontal.



 <i>Vertical Synchronization</i>	<i>Vertical Synchronization</i> , berfungsi untuk mensinkronisasikan 2 cabang <i>event</i> yang posisinya vertikal.
 <i>Decision</i>	<i>Decision</i> , digunakan ketika terjadi pemilihan 2 kondisi <i>event</i> pada diagram.
 End State	End State, sebagai tanda akhir dari <i>activity diagram</i> .

(Sumber : Rosa dan Shalahuddin , 2014:162)

#### 2.8.4 Sequence Diagram

Menurut Sucipto (2011:210) banyaknya diagram *sequence* yang harus Digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram *sequence* sehingga semakin banyak *use case* yang didefinisikan maka diagram *sequence* yang harus dibuat juga semakin banyak.

Menurut (Irmayani & Susyatih, 2017) *Sequence* diagram menggambarkan bagaimana sistem merespon kegiatan user. *Sequence* diagram yang dibuat yaitu yang berhubungan langsung dengan kegiatan utama dari sistem informasi anggaran pendapatan dan belanja desa berbasis objek”.

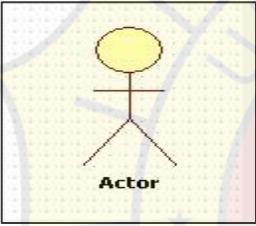
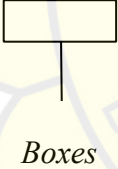


Menurut Sukanto dan Shalahuddin (2015:165) menyimpulkan bahwa diagram *sequence* menggambarkan kelakuan objek pada *use case* dengan


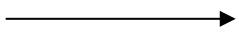


mendesripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram *sequence* juga dibutuhkan untuk melihat scenario yang ada pada *use case*.

Berikut adalah simbol-simbol yang terdapat pada diagram *sequence*:

Tabel 2. 6 Simbol-Simbol Pada Sequence Diagram

<b>Sequence Diagram</b>	<b>Penjelasan</b>
	<i>Actor</i> , menggambarkan seseorang atau sesuatu (seperti perangkat atau sistem lain) yang berinteraksi dengan sistem.
	<i>Boxes</i> , sebuah kontak yang tampil pada posisi paling atas diagram, yang mewakili <i>object</i> , <i>use case</i> , <i>class</i> , dan <i>actor</i> .
	<i>Return Message</i> , menggambarkan pesan atau hubungan antara obyek yang menunjukkan urutan kejadian yang terjadi.
	<i>Lifeline</i> , eksekusi obyek selama <i>sequence</i> ( <i>message</i> dikirim atau diterima dan aktifasinya)

 <i>Message to Self</i>	<i>Message to Self</i> , menggambarkan pesan atau hubungan obyek itu sendiri yang menunjukkan urutan kejadian yang terjadi.
 <i>Object Message</i>	<i>Object Message</i> , menggambarkan pesan atau hubungan antar obyek yang menunjukkan urutan kejadian yang terjadi.

(Sumber : Rosa dan Shalahuddin , 2014:162)

## 2.9 Sistem

Sistem adalah sekelompok elemen – elemen yang saling berinteraksi untuk mencapai tujuan tertentu. Pada dasarnya sistem adalah suatu kerangka dari prosedur – prosedur yang saling bersangkutan, yang disusun sesuai dengan skema yang menyeluruh untuk melaksanakan suatu kegiatan atau fungsi utama dari perusahaan yang dihasilkan oleh suatu proses tertentu yang bertujuan untuk menyediakan informasi untuk membantu mengambil keputusan manajemen operasi perusahaan dari hari ke hari serta menyediakan informasi yang layak untuk pihak diluar perusahaan. Berikut adalah pengertian sistem yang dikemukakan oleh para ahli, yaitu :

1. Menurut Azhar Susanto (2013, 22) dalam bukunya yang berjudul Sistem Informasi Akuntansi menyatakan sistem adalah kumpulan dari sub sistem / bagian / komponen apapun baik fisik ataupun non fisik yang saling

berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan tertentu.

2. Menurut Sutarman (2009, 5) dalam bukunya yang berjudul Pengantar Teknologi Informasi menyatakan sistem adalah kumpulan elemen yang saling berinteraksi dalam suatu kesatuan untuk menjalankan suatu proses pencapaian suatu tujuan utama.
3. Menurut Mulyadi (2010, 5) dalam bukunya yang berjudul Sistem Akuntansi menyatakan bahwa sistem adalah jaringan prosedur yang dibuat menurut pola yang terpadu untuk melaksanakan kegiatan – kegiatan pokok perusahaan.

Berdasarkan beberapa pendapat yang dikemukakan diatas dapat ditarik kesimpulan bahwa sistem adalah kumpulan bagian – bagian atau sub sistem yang disatukan dan dirancang untuk mencapai suatu tujuan.

### **2.10 Sistem Informasi**

Sesungguhnya, yang dimaksud dengan sistem informasi tidak harus melibatkan computer. Sistem informasi yang menggunakan computer biasa disebut sistem informasi berbasis computer (Computer-Based Information System atau CBIS). Istilah sistem informasi lebih sering dipakai tanpa embel-embel berbasis computer walaupun dalam kenyatannya computer merupakan bagian yang penting. Dibuku ini, yang dimaksudkan dengan sistem informasi adalah sistem informasi yang berbasis komputer.