

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Sistem**

Sistem merupakan kumpulan dari unsur atau elemen-elemen yang saling berkaitan/berinteraksi dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu (Asbon Hendra,2012:157).

#### **II.2.Pakar**

Pakar adalah seorang individu yang memiliki pengetahuan khusus, pemahaman, pengalaman, dan metode-metode yang digunakan untuk memecahkan persoalan dalam bidang tertentu (Rika Rosnelly,2012:10). Seorang pakar memiliki kemampuan kepakaran yaitu:

1. Dapat mengenali dan merumuskan suatu masalah.
2. Menyelesaikan masalah dengan cepat dan tepat.
3. Menjelaskan solusi dari suatu masalah.
4. Restrukturisasi pengetahuan.

Pakar atau ahli (*expert*) didefinisikan sebagai seseorang yang memiliki pengetahuan atau keahlian khusus yang tidak dimiliki oleh kebanyakan orang. Seorang pakar dapat memecahkan masalah yang tidak mampu dipecahkan kebanyakan orang. Dengan kata lain, dapat memecahkan suatu masalah dengan lebih

efisien namun bukan berarti lebih murah (Rika Rosnelly,2012:3).

### **II.3. Sistem Pakar**

Sistem pakar merupakan penyelesaian pendekatan yang tepat dan bagus untuk permasalahan AI klasik dari pemrograman *intelligent* (cerdas).Sistem pakar (*expert system*) merupakan solusi AI bagi masalah pemrograman cerdas (*Intelligent*).

Dengan kata lain, 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 untuk memecahkan masalah (Rika Rosnelly;2012:2).

Pengetahuan yang dimuat dalam sistem pakar dapat berasal dari seorang pakar atau pun pengetahuan yang berasal dari buku, jurnal, majalah, dan dokumentasi yang dipublikasikan lainnya, serta orang yang memiliki pengetahuan meski bukan ahli. Istilah sistem pakar (*expert system*), sering disinonimkan dengan sistem berbasis pengetahuan (*knowledge-based system*) atau sistem pakar berbasis pengetahuan (*knowledge based expert system*) (Rika Rosnelly,2012:3)

#### **II.3.1. Karakteristik Sistem Pakar**

Sistem pakar umumnya dirancang untuk memenuhi beberapa karakteristik umum berikut ini :

1. Kinerja Sangat Baik (*High Performance*). Sistem harus mampu memberikan respon berupa saran (*advice*) dengan tingkat kualitas yang sama dengan seorang pakar atau melebihinya.

2. Waktu respon yang baik (*Adequate response time*). Sistem juga harus mampu bekerja dalam waktu yang sama baiknya (*reasonable*) atau lebih cepat dibandingkan dengan seorang pakar dalam menghasilkan keputusan. Hal ini sangat penting terutama pada sistem waktu nyata (*real-time*).
3. Dapat diandalkan (*Good reliability*). Sistem harus dapat diandalkan dan tidak mudah rusak / crash.
4. Dapat dipahami (*Understanable*). Sistem harus mampu menjelaskan langkah-langkah penalaran yang dilakukannya seperti seorang pakar.
5. Fleksibel (*Flexibility*). Sistem harus menyediakan mekanisme untuk menambah, mengubah, dan menghapus pengetahuan (Rika Rosnelly,2012:20-21).

### **II.3.2. Kelebihan dan Kekurangan Sistem Pakar**

Menurut Soetoyo, Sistem yang dibuat dan dikembangkan oleh manusia memiliki kelebihan yang dapat digunakan untuk membantu dalam pemecahan suatu masalah, sebaliknya sistem juga memiliki kekurangan (Raditya Pratama, 2014). Adapun kelebihan sistem pakar antara lain .

1. Membantu orang awam untuk menyelesaikan masalah tanpa bantuan para pakar.
2. Meningkatkan kualitas dan produktivitas.
3. Memiliki kemampuan untuk mengakses pengetahuan dan keahlian para ahli baik yang biasa maupun yang langka.
4. Sebagai asisten para ahli sehingga meningkatkan pekerjaan para ahli.

5. Memiliki rehabilitas.
6. Dapat menghemat waktu dan pengambilan keputusan.

Sedangkan kelemahan sistem pakar diantaranya adalah:

1. Tidak ada jaminan bahwa sistem pakar memuat 100% kepakaran yang di perlukan.
2. Pengembangan sistem pakar tergantung ada tidaknya pakar dibidang sehingga pengembangannya dapat terkendali.
3. Biaya untuk mendesain, mengimplementasikan dan memeliharanya cukup mahal tergantung seberapa lengkap dan kemampuannya.

#### **II.4.Sarung Tangan**

Lateks adalah produk yang di dapat dari getah pohon *Hevea brasiliensis* yang berasal dari hutan amazon. Getah karet alam merupakan gabungan partikel yang mengandung 35% cis 1,4 polysoprene (karet), 55-60% air, 5-10% bahan lain. Protein yang terdapat dalam getah karet antara 1-1,8%, berisi bahan karet *cis-1, 4 polyisoprene*. Bahan ini terutama terdiri dari cis -1,4-polyisoprene, polimer organik yang memberikan sebagian besar kekuatan dan elastisitas lateks. Juga terkandung berbagai macam gula, lipid, asam nukleat, dan protein yang sangat alergi.

Dalam proses pembuatannya, ditambahkan bahan dan zat-zat penstabil seperti ammoniak dan bahan-bahan kimia lain pada sarung tangan lateks. Penambahan bahan-bahan sarung tangan NRL mengubah lateks dari bentuknya yang awalnya cair menjadi lapisan yang sangat tipis, elastis dan kuat.

### 1. Antigen Kimia

Bahan kimia yang utama ditambahkan dalam proses pembuatan karet lateks yaitu akselator dan antioksidan yang mencapai lebih dari 90%. Akselator yang ditambahkan pada NRL terdiri dari *Thiuram-mix*, *Carba-mix*, *Mercapto-mix*. Akselator merupakan bahan kimia yang digunakan untuk mempercepat proses vulkanisasi yang bekerja sebagai katalisator.

### 2. Antigen Protein

Untuk menganalisa alergen protein yang terdapat pada NRL menggunakan 2-D elektroforesis. Pada NRL ditemukan lebih dari 250 jenis protein / polipeptida dan hanya kira-kira 30 jenis yang dapat berkaitan dengan antibodi IgE serum penderita alergi NRL. Alergen protein NRL yang umumnya dijumpai pada pekerja kesehatan yaitu Hev b 5 : 62%, Hev b 6: 65% dan Hev b 7 : 41%.

### 3. Serbuk Sarung Tangan

Serbuk sarung tangan adalah tepung jagung yang sudah dimodifikasi dan digunakan untuk membantu dalam mengenakan sarung tangan. Serbuk digunakan dalam pembuatan sarung tangan terutama untuk mencegah bloking atau lekatnya permukaan NRL (Rachmat Andy Nursecha, 2013).

## **II.5. Metode Teorema Bayes**

*Teorema Bayes* merupakan kaidah yang memperbaiki atau merevisi suatu probabilitas dengan cara memanfaatkan informasi tambahan. Maksudnya, dari probabilitas awal (*prior probability*) yang belum diperbaiki yang dirumuskan

berdasarkan informasi yang tersedia saat ini, kemudian dibentuklah probabilitas berikutnya (*posterior probability*). (Hartatik, dkk, 2015 : 28).

Rumus teorema Bayes dapat dilihat sebagai berikut :

$$P(H|E) = \frac{P(E|H) * P(H)}{P(E)} \dots \dots \dots (1)$$

Dimana :

$P(H | E)$  = probabilitas hipotesis H jika diberikan *evidence* E.

$P(E | H)$  = probabilitas munculnya *evidence* E jika diketahui hipotesis H.

$P(H)$  = probabilitas H tanpa mengandung *evidence* apapun.

$P(E)$  = probabilitas *evidence* E.

Kekurangan pada metode Teorema Bayes adalah sebagai berikut :

1. Metode Bayes hanya bisa digunakan untuk persoalan klasifikasi yang data-datanya bersifat kategorikal dan dengan metode pembelajaran terawasi.
2. Diperlukan pengetahuan awal bagi Metode Bayes untuk dapat mengambil suatu keputusan. Tingkat keberhasilannya sangat tergantung pada pengetahuan awal yang diberikan. (Belutowe, 2015 : 51).

### **II.5.1. Studi Kasus Teorema Bayes**

Arman melakukan penentuan kualitas sarung tangan berbahan karet dengan menjawab pertanyaan sesuai dengan jawaban berikut:

$$K1 = 0.7 = P(E|H1)$$

$$K2 = 0.9 = P(E|H2)$$

$$K3 = 0.7 = P(E|H3)$$

$$K4 = 0.8 = P(E|H4)$$

$$K5 = 0.5 = P(E|H3)$$

$$K6 = 0.7 = P(E|H4)$$

Kemudian mencari nilai semesta dengan menjumlahkan dari hipotesa di atas :

$$K=1$$

$$= K1 + K2 + K3 + K4 + K5 + K6$$

$$= 0.7 + 0.9 + 0.7 + 0.8 + 0.5 + 0.6$$

$$= 4.3$$

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

$$P(H1) = \frac{H1}{\sum_{k=1}^6 K} = \frac{0.7}{4.3} = 0.163$$

$$P(H2) = \frac{H2}{\sum_{k=1}^6 K} = \frac{0.9}{4.3} = 0.209$$

$$P(H3) = \frac{H3}{\sum_{k=1}^6 K} = \frac{0.7}{4.3} = 0.163$$

$$P(H4) = \frac{H4}{\sum_{k=1}^6 K} = \frac{0.8}{4.3} = 0.186$$

$$P(H5) = \frac{H5}{\sum_{k=1}^6 K} = \frac{0.5}{4.3} = 0.116$$

$$P(H_6) = \frac{H_6}{k=1} = \frac{0.7}{4.3} = 0.163$$

Setelah nilai  $P(H_i)$  diketahui, probabilitashipotesis H tanpa memandang *evidence* apapun, makalangkah selanjutnya adalah :

$$\begin{aligned} &= P(H_1) * P(E|H_1) + P(H_2) * P(E|H_2) + P(H_3) * P(E|H_3) + P(H_4) * P(E|H_4) + P(H_5) * \\ &P(E|H_5) + P(H_6) * P(E|H_6) \\ &= (0.163 * 0.7) + (0.209 * 0.9) + (0.163 * 0.7) + (0.186 * 0.8) + (0.116 * 0.5) + (0.163 \\ &* 0.7) \\ &= 0.1141 + 0.1881 + 0.1141 + 0.1488 + 0.1141 \\ &= 0.7372 \end{aligned}$$

Langkah selanjutnya ialah mencari nilai  $P(H_i|E)$  atau probabilitas hipotesis  $H_i$  benar jika diberikan *evidence* E.

$$P(H_1|E) = \frac{0.7 * 0.163}{0.7372} = 0.1548$$

$$P(H_2|E) = \frac{0.9 * 0.209}{0.7372} = 0.2552$$

$$P(H_3|E) = \frac{0.7 * 0.163}{0.7372} = 0.1548$$

$$P(H_4|E) = \frac{0.7 * 0.186}{0.7372} = 0.2018$$

$$P(H_5|E) = \frac{0.5 * 0.116}{0.7372} = 0.0787$$

$$P(H_6|E) = \frac{0.7 * 0.163}{0.7372} = 0.1548$$

Setelah seluruh nilai  $P(H_i|E)$  diketahui, maka jumlahkan seluruh nilai bayesnya

$$\begin{aligned}
&= (0.7 * 0.1548) + (0.9 * 0.2552) + (0.7 * 0.1548) + (0.8 * 0.2018) + (0.5 * 0.0787) \\
&\quad + (0.7 + 0.1548) \\
&= 0.10836 + 0.22968 + 0.10836 + 0.16144 + 0.03935 + 0.10836 \\
&= 0,75555 * 100\% \\
&= 75,555\%.
\end{aligned}$$

### **Pengertian Database**

Menurut Connolly, Salah satu dari teknologi sistem informasi adalah basis data atau sering disebut *database*. Adapun definisi dari basis data yang berbunyi *database is a single, large repository of data which can be used simultaneously by many departments and users* atau dengan kata lain basis data adalah tempat penyimpanan data yang besar dan tunggal yang dapat digunakan secara bersamaan oleh beberapa departemen atau pengguna (Yanti Efandi, 2012).

### **II.6. Normalisasi Database**

Normalisasi merupakan parameter digunakan untuk menghindari duplikasi terhadap tabel dalam basis data dan juga merupakan proses mendekomposisi sebuah tabel yang masih memiliki beberapa anomali atau ketidakwajaran sehingga menghasilkan tabel yang lebih sederhana dan struktur yang bagus, yaitu sebuah tabel yang tidak memiliki *data redundancy* dan memungkinkan *user* untuk melakukan *insert*, *delete*, dan *update* pada baris (*record*) tanpa menyebabkan inkonsistensi data. (Triyono, 2012 : 19).

### 1. *First Normal Form (1 NF)*

Sudah tidak ada *repeating group* yaitu pengulangan yang terjadi pada beberapa atribut atau kolom dalam sebuah tabel, dan juga setiap atribut harus bernilai tunggal. Atribut *multivalued, composite, derive* tidak tunggal. Setiap nilai dari atribut hanya mempunyai nilai tunggal.

### 2. *Second Normal Form (2 NF)*

Untuk menjadikan tabel normal tingkat ke 2 maka sudah 1NF dan setiap atribut yang bukan *primary key* sepenuhnya secara *functional* tergantung pada semua atribut pembentuk *primary key*.

### 3. *Third Normal Form (3 NF)*

Tabel sudah 2NF dan tidak memiliki *transitive dependencies*, *Transitive dependency* adalah ketika ada atribut yang secara tidak langsung tergantung pada *primary key* dan atribut tersebut juga tergantung pada atribut lain yang bukan *primary key*.

### 4. *Boyce-codd Normal Form (BCNF)*

Tabel dalam BCNF jika sudah 3NF dan semua *determinants* adalah *candidate keys*. Perbedaan 3NF dan BCNF adalah untuk *functional dependency*  $A \rightarrow B$ , 3NF memperbolehkan ketergantungan ada dalam relasi jika B adalah *Primary Key* dan A bukan merupakan *candidate key*. Sedangkan BCNF menuntut untuk ketergantungan tetap ada dalam relasi, A harus menjadi *candidate key*.

#### 5. *Fourth Normal Form (4 NF)*

Relasi berada pada bentuk normal keempat apabila memenuhi syarat BCNF dan tidak mempunyai *multivalued dependency*.

#### 6. *Fifth Normal Form (5 NF)*

Tabel bentuk normal kelima sering disebut PJNF (*Projection Join Normal Form*), penyebutan PJNF karena untuk suatu relasi akan berbentuk normal kelima jika tabel tersebut dapat dipecah atau diproyeksikan menjadi beberapa tabel dan dari proyeksi-proyeksi itu dapat disusun kembali (*join*) menjadi tabel yang sama dengan keadaan semula. Jika penyusunan ini tidak mungkin dilakukan dikatakan pada relasi itu terdapat *join dependencies* dan dikatakan bersifat *lossy join*. (Triyono, 2012 : 20).

### **II.7. UML (*Unified Modelling Language*)**

Menurut Windu Gata, hasil pemodelan pada OOAD (*Object Oriented Analysis And Design*) terdokumentasikan dalam bentuk *Unified Modeling Language (UML)*. UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem (Gellysa Urva dan Helmi Fauzi Siregar, 2015).

Alat bantu yang digunakan dalam perancangan berorientasi objek berdasarkan UML adalah sebagai berikut :

a. *Class Diagram* (Diagram Kelas)

Diagram kelas menggambarkan jenis-jenis dari objek dalam suatu sistem dan berbagai jenis hubungan statis yang ada diantaranya. Sebuah kelas merupakan kumpulan dari objek yang memiliki karakteristik yang sama seperti atribut, operasi hubungan, dan semantik. Sebuah kelas mengimplementasikan satu atau lebih *interface*.

**Tabel II.1. Multiplicity Class Diagram**

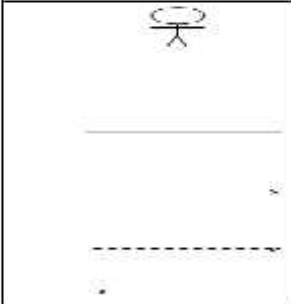
<b>Multiplicity</b>	<b>Penjelasan</b>
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar,2015)

b. *Usecase Diagram*

*Usecase* adalah rangkain atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah actor.*Usecase* digunakan untuk membentuk tingkah laku benda dalam sebuah model serta direalisasikan oleh sebuah kolaborasi.

Tabel II.2. Simbol *Use Case*




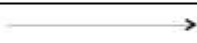
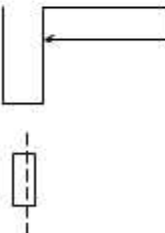

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit unit yang terdapat pesan untuk unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p> <p>Aktor adalah <i>personality</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditunjukkan pembagian tenaga kerja dan tugas tugas yang berkaitan dengan peran pada banyak target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Peran dicatat bahwa aktor berinteraksi dengan <i>use case</i>, tetapi tidak memiliki control terhadap <i>use case</i>.</p> <p>Asosiasi antara aktor dan <i>use case</i> digambarkan dengan garis lurus panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.</p> <p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.</p> <p><i>Sub-use</i> merupakan di dalam <i>use case</i> lain (<i>parent</i>) atau penerangan <i>use case</i> oleh <i>use case</i> lain, umumnya adalah penggantian sebuah fungsi yang sama. <i>Extend</i>, merupakan perlakuan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.</p>

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar,2015)

### c. *Sequence Diagram*

*Sequence diagram* menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem (Hamim Tohari; 2014:101).

Tabel II.3. Simbol *Sequence Diagram*





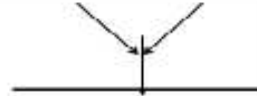
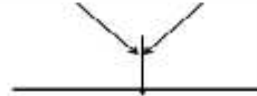

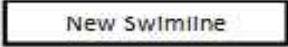
Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.  <i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar,2015)

#### d. *Activity Diagram*

*Activity Diagram* memodelkan *workflow* proses bisnis dan urutan aktivitas dalam sebuah proses. Diagram ini sangat mirip dengan flowchart karena memodelkan *workflow* dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke status.

Tabel II.4. Simbol *Activity Diagram*

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas
	<i>End point</i> , akhir aktifitas.
  	<i>Activitas</i> , menggambarkan suatu proses/kegiatan bisnis. <i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i>
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar,2015)

## II.8. Visual Basic 2010

Visual Basic 2010 merupakan versi perbaikan dan pengembangan dari versi sebelumnya, yaitu VisualBasic 2008 yang termasuk ke dalam produk pemrograman terbaru yang dikeluarkan oleh Microsoft, yaitu Microsoft Visual Studio 2010 yang termasuk juga ke dalam bagian dari Microsoft .NET. Beberapa pengembangan yang terdapat di dalamnya antara lain dukungan terhadap *library* terbaru

dari Microsoft, yaitu dukungan terhadap pengembangan aplikasi menggunakan Microsoft SilverLight, dukungan terhadap aplikasi berbasis *Cloud Computing*, serta perluasan dukungan terhadap *database* yang *stand alone* maupun *server* (Santi dkk, 2015).

## **II.9. SQL Server 2008**

*SQL Server 2008* adalah sebuah terobosan baru dari *Microsoft* dalam bidang *database*. *SQL Server* adalah sebuah *DBMS (Database Management System)* yang dibuat oleh *Microsoft* untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti *IBM* dan *Oracle*. *SQL Server 2008* dibuat pada saat kemajuan dalam bidang *hardware* sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa *SQL Server 2008* membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data (Agustinus Setiawan dan Rika Putri Permadani, 2016)