

## **BAB II**

### **LANDASAN TEORI**

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

*Artificial intelligence* (AI) memiliki beberapa domain masalah atau area. Bidang sistem pakar merupakan penyelesaian pendekatan yang sangat berhasil dan bagus untuk permasalahan AI klasik dari pemrograman *intelligent* (cerdas).

Sistem pakar (*expert system*) merupakan solusi AI bagi masalah pemrograman pintar (*intelligent*). Profesor Edward Feigenbaum dari Stanford University yang merupakan pionir dalam teknologi sistem pakar mendefinisikan sistem pakar sebagai sebuah program komputer 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.

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 (Rosnelly, 2013 : 2).

### II.1.1 Kelebihan Sistem Pakar

Sistem pakar memiliki beberapa fitur menarik yang merupakan kelebihanannya (Rosnelly, 2013 : 5-6), seperti :

1. Meningkatkan ketersediaan (*increased availability*). Kepakaran atau keahlian menjadi tersedia dalam sistem komputer. Dapat dikatakan bahwa sistem pakar merupakan produksi kepakaran secara masal (*massproduction*).
2. Mengurangi biaya (*reduce cost*). Biaya yang diperlukan untuk menyediakan keahlian per satu orang *user* menjadi berkurang.
3. Mengurangi bahaya (*reduce danger*). Sistem pakar dapat digunakan di lingkungan yang mungkin berbahaya bagi manusia.
4. Permanen (*permanence*). Sistem pakar dan pengetahuan yang terdapat di dalamnya bersifat lebih permanen dibandingkan manusia yang dapat merasa lelah, bosan dan pengetahuannya hilang saat pakar meninggal dunia.
5. Keahlian multiple (*multiple expertise*). Pengetahuan dari beberapa pakar dapat dimuat ke dalam sistem dan bekerja secara simultan dan kontinu menyelesaikan suatu masalah setiap saat. Tingkat keahlian atau pengetahuan yang digabungkan dari beberapa pakar dapat melebihi pengetahuan satu orang pakar.
6. Meningkatkan kehandalan (*increased reliability*). Sistem pakar meningkatkan kepercayaan dengan memberikan hasil yang benar sebagai alternatif pendapat dari seorang pakar atau sebagai penengah
7. jika terjadi konflik antara beberapa pakar. Namun hal tersebut tidak berlaku jika sistem dibuat oleh salah seorang pakar, sehingga akan selalu sama

dengan pendapat pakar tersebut kecuali jika sang pakar melakukan kesalahan yang mungkin terjadi pada saat tertekan.

8. Penjelasan (*explanation*). Sistem pakar dapat menjelaskan detail proses penalaran (*reasoning*) yang dilakukan hingga mencapai suatu kesimpulan. Seorang pakar mungkin saja terlalu lelah, tidak bersedia atau tidak mampu melakukannya setiap waktu. Hal ini akan meningkatkan tingkat percayaan bahwa kesimpulan yang dihasilkan adalah benar.

9. Respon yang cepat (*fast response*). Respon yang cepat atau *real time* diperlukan pada beberapa aplikasi. Meskipun bergantung pada *hardware* dan *software* yang digunakan, namun sistem pakar relatif memberikan respon yang lebih cepat dibandingkan seorang pakar.

10. Stabil, tidak emosional dan memberikan respon yang lengkap setiap saat (*steady, unemotional and complete response at all times*). Karakteristik ini diperlukan pada situasi *real time* dan keadaan darurat (*emergency*) ketika seorang pakar mungkin tidak berada pada kondisi puncak disebabkan oleh stres atau kelelahan.

11. Pembimbing pintar (*intelligent tutor*). Sistem pakar dapat berperan sebagai *intelligent tutor* dengan memberikan kesempatan pada *user* untuk menjalankan contoh program dan menjelaskan proses *reasoning* yang dilakukan.

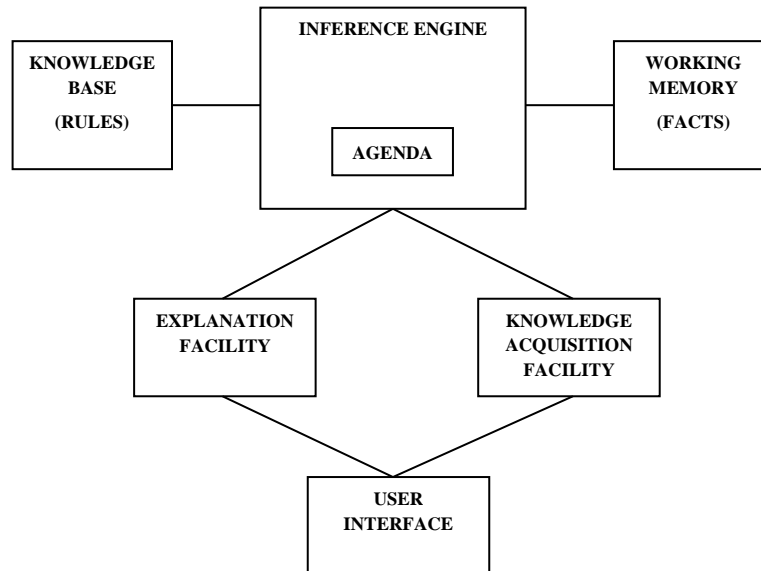
12. Basis data cerdas (*intelligent database*). Sistem pakar dapat digunakan untuk mengakses basis data secara cerdas.

### II.1.2. Konsep Umum Sistem Pakar

Sistem pakar umumnya dirancang dengan cara yang berbeda dengan sistem konvensional lain, terutama karena masalah yang dihadapi umumnya tidak memiliki solusi algoritmik dan bergantung pada inferensi untuk mendapatkan solusi yang terbaik yang paling mungkin. Oleh karena itu sistem pakar harus mampu menjelaskan inferensi yang dilakukannya sehingga hasil yang diperoleh dapat diperiksa. *Explanation facility* merupakan bagian terintegrasi dari sebuah sistem pakar. Sebuah *explanation facility* yang detail dirancang untuk memungkinkan *user* mengeksplorasi *rules* melalui tipe pertanyaan “*what if*” yang disebut *hypothetical reasoning* dan bahkan menerjemahkan *natural language* ke dalam *rules* dengan cara induksi dari tabel data. (Rosnelly, 2013 : 8).

### II.1.3. Struktur Sistem Pakar

Adapun struktur sistem pakar dapat dilihat pada gambar II.1. berikut ini :



**Gambar II.1. Struktur Sistem Pakar**  
(Sumber : Rika Rosnelly, 2013 : 13)

Komponen yang terdapat dalam struktur sistem pakar ini adalah *knowledge base (rules)*, *inference engine*, *working memory*, *explanation facility*, *knowledge acquisition facility*, *user interface*. (Rosnelly, 2013 : 13-15).

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

Basis pengetahuan mengandung pengetahuan untuk pemahaman, formulasi dan penyelesaian masalah. Komponen sistem pakar disusun atas dua elemen dasar, yaitu fakta dan aturan. Fakta merupakan informasi tentang objek dalam area permasalahan tertentu, sedangkan aturan merupakan informasi tentang cara bagaimana memperoleh fakta baru dari fakta yang telah diketahui.

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

Mesin inferensi merupakan otak dari sebuah sistem pakar dan dikenal juga dengan sebutan *control structure* (struktur kontrol) atau *rule interpreter* (dalam sistem pakar berbasis kaidah). Komponen ini mengandung mekanisme pola pikir dan penalaran yang digunakan oleh pakar dalam menyelesaikan suatu masalah. Mesin inferensi disini adalah *processor* pada sistem pakar yang mencocokkan bagian kondisi dari *rule* yang tersimpan di dalam *knowledge base* dengan fakta yang tersimpan di *working memory*.

## 3. *Working Memory*

Berguna untuk menyimpan fakta yang dihasilkan oleh *inference engine* dengan penambahan parameter berupa derajat kepercayaan atau dapat juga dikatakan sebagai global database dari fakta yang digunakan oleh rule-rule yang ada.

## 4. *Explanation Facility*

Menyediakan kebenaran dari solusi yang dihasilkan kepada user (*reasoning chain*).

## 5. *Knowledge Acquisition Facility*

Meliputi proses pengumpulan, pemindahan dan perubahan dari kemampuan pemecahan masalah seorang pakar atau sumber pengetahuan terdokumentasi pemecahan ke program komputer, yang bertujuan untuk memperbaiki atau mengembanglan basis pengetahuan.

## 6. *User Interface*

Mekanisme untuk memberi kesempatan kepada user dan sistem pakar untuk berkomunikasi. Antar muka menerima informasi dari pemakai dan mengubahnya ke dalam bentuk yang dapat diterima oleh sistem. Selain itu, antar muka menerima informasi dari sistem dan menyajikannya ke dalam bentuk yang dapat dimengerti oleh pemakai.

### II.1.4. Karakteristik Sistem Pakar

Menurut Rosnelly (2013 : 20-21), sistem pakar umumnya dirancang untuk memenuhi beberapa karakteristik umum berikut ini :

- a. **Kinerja sangat baik** (*high performance*). Sistem harus mampu memberikan respon berupa saran (*advice*) dengan tingkat kualitas yang sama dengan seorang pakar atau melebihinya.
- b. **Waktu respon yang baik** (*adequate respon 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*).
- c. **Dapat diandalkan** (*good reliabilty*). Sistem harus dapat diandalkan dan tidak mudah rusak/*crash*.
- d. **Dapat dipahami** (*understandable*). Sistem harus mampu menjelaskan langkah-langkah penalaran yang dilakukannya seperti seorang pakar. Hal ini penting untuk beberapa alasan, yaitu :

- 1) Dimungkinkan bahwa sistem pakar berkaitan dengan nyawa manusia atau properti lainnya sehingga harus dapat menjelaskan mengapa dihasilkan suatu kesimpulan tertentu.
  - 2) Untuk mengkonfirmasi bahwa pengetahuan pakar telah dikumpulkan dengan benar dan digunakan oleh sistem dengan benar pula. Hal ini penting dalam proses debugging pengetahuan yang mungkin salah karena pengetikan atau pemahaman yang salah dari *knowledge engineer*.
- e. **Fleksibel** (*flexibility*). Sistem harus menyediakan mekanisme untuk menambah dan menghapus pengetahuan.

## II.2. Tanaman Anggrek *Phalaenopsis* (*Phalaenopsis amabilis*)

Anggrek sebagai tanaman hias pot sangat diminati masyarakat karena keindahan dan keunikan bunganya. Keindahan dan keunikan bunga anggrek dapat dilihat dari variasi warna, ukuran dan bentuk bunga yang dihasilkan. Hal lain juga, beberapa jenis anggrek mengeluarkan aroma wangi. Keunikan semakin bernilai tinggi dijumpai pada jenis-jenis anggrek spesies. Anggrek bulan (*Phalaenopsis amabilis*) merupakan jenis anggrek (*Orchidaceae*) yang mempunyai ciri khas kelopak bunga yang lebar dan berwarna putih. Meskipun saat ini sudah banyak anggrek bulan hasil persilangan ( anggrek bulan hibrida ) yang memiliki corak dan warna beragam jenis ( Yulia Andiani; 2008:21 ).

Anggrek *Phalaenopsis* termasuk tanaman monopodial yaitu pertumbuhan tanaman vertikal pada satu titik tumbuh dan hanya satu batang utama. Ia hidup sebagai tanaman epifit yaitu hidup menumpang pada tanaman lain tetapi tidak

mengambil makanan dari tanaman yang ditumpangi atau sebagai tanaman litofit yang tumbuh diatas batuan bukit karang yang sudah lapuk dan kadang berlumut.

Tanaman yang memiliki bunga indah ini berbatang pendek dan tidak menghasilkan umbi palsu seperti tanaman anggrek lainnya. Akar *Phalaenopsis* adalah yang paling menarik diantara anggrek lainnya, karena dalam kehidupan liarnya akan bergelayut erat pada tanaman yang di tumpangi. Akarnya seakan berdaging dan tumbuh liar.

### II.2.1. Klasifikasi dan Morfologi Tanaman Anggrek *Phalaenopsis*

#### 1. Klasifikasi Tanaman Anggrek *Phalaenopsis*

Kedudukan tanaman anggrek bulan dalam sistematika (*taksonomi*) tumbuhan dapat diklasifikasikan sebagai berikut :

Kerajaan	: <i>Plantae</i>
Divisi	: <i>Spermatophyta</i>
Sub divisi	: <i>Angiospermae</i>
Kelas	: <i>Monocotyledonae</i>
Bangsa	: <i>Orchidales</i>
Suku	: <i>Orchidaceae</i>
Marga	: <i>Phalaenopsis</i>
Jenis	: <i>Phalaenopsis</i> sp.

## 2. Morfologi Tanaman Anggrek *Phalaenopsis*

Susunan tubuh tanaman anggrek bulan terdiri dari bunga, buah, biji, daun, batang, dan akar.

### a. Bunga Anggrek Bulan

Bunga anggrek bulan tersusun dalam tandan dan kadang-kadang bercabang dengan panjang karangan bunga mencapai 50 cm yang tumbuh menjuntai. Setiap tangkai mendukung 10-12 kuntum bunga dengan daun penumpu 5 mm berbentuk segitiga, bunganya cukup harum dan waktu mekarnya lama. Perhiasan bunga tersusun membulat dengan diameter 6-10 cm atau lebih dan mahkotanya bertumpang tindih dengan kelopak tersusun membundar (Puspitaningtyas, 2010).

Bentuk bunga anggrek *Phalaenopsis* ada dua, yaitu bulat (*round shape*) dan bintang (*star*). Bunga anggrek terdiri dari kelopak (*sepal*), mahkota (*petal*), dan lidah (*labelum*). Sepal yang dimiliki anggrek terdiri atas tiga helai dan tiga helai petal yang salah satu petal berubah menjadi bibir bunga atau labelum. Selain itu, terdapat bagian lain yang disebut tugu, yaitu perpanjangan gagang bunga (bakal buah), dibentuk oleh penyatuan putik dan benang sari (Kencana, 2007).

Warna bunga putih bersih dengan sedikit variasi kuning dan bintik kemerahan di bibir bunga. Bunga anggrek *Phalaenopsis* juga memiliki motif yang beragam diantaranya motif titik-titik, garis-garis, blok dan sembur (*splash*). Susunan bunganya sangat artistik, tersusun rapi, menjuntai ke bawah, dan berselang-seling (Setiawan, 2005). Bibir kedua cuping samping

tegak melebar dan bagian tepi depannya berwarna kuning dengan garis kemerahan. Buah berbentuk bulat lonjong, berukuran 7,5 x 1,3 cm (Puspitaningtyas, 2010).

b. Buah dan Biji Anggrek Bulan (*Phalaenopsis*)

Bentuk buah anggrek merupakan lentera atau *capsular* yang memiliki 6 rusuk. Tiga diantaranya merupakan rusuk sejati dan tiga rusuk yang lain merupakan tempat melekatnya dua tepi daun buah yang berlainan. Buah anggrek yang mencapai besarnya jari kelingking memiliki ratusan ribu bahkan jutaan biji anggrek yang sangat lembut dalam ukuran yang sangat kecil di dalamnya. Biji-biji anggrek tidak memiliki endosperm sebagai cadangan makanan, sehingga untuk perkecambahannya dibutuhkan nutrisi yang berfungsi membantu pertumbuhan biji.

c. Daun dan Batang Anggrek Bulan (*Phalaenopsis*)

Anggrek bulan termasuk anggrek epifit monopodial yang tumbuh menjuntai. Batangnya sangat pendek dan terbungkus oleh seludang daun. Daunnya berjumlah kurang dari lima helai, berwarna hijau, tebal, berdaging, berbentuk lonjong bulat telur sungsang atau jorong, melebar di bagian ujungnya, berujung tumpul, atau sedikit meruncing, dengan panjang 20-30 cm dan lebar 5-8 cm.

d. Akar Anggrek Bulan (*Phalaenopsis*)

Akar anggrek bulan berbentuk bulat memanjang serta berdaging, bercabang, berwarna putih dan hijau di bagian ujungnya (Puspitaningtyas, 2010). Menurut Rukmana (2008), akar tanaman anggrek bulan terdiri dari dua

macam yaitu akar lekat dan akar udara (*aerial*). Akar lekat berfungsi untuk melekat dan menahan keseluruhan tanaman agar tetap berada pada posisinya. Bagian ujung akar meruncing dan sedikit lengket, dalam keadaan kering, akar tampak berwarna putih keperak-perakan dan hanya bagian ujung akar saja berwarna hijau atau tampak keunguan. Akar yang sudah tua akan berwarna coklat tua dan kering. Akar udara atau akar aerial merupakan akar yang keluar dari batang atas. Akar udara atau akar aerial yang tidak melekat pada batang pohon tidak ditumbuhi rambut akar. Akar aerial yang masih aktif ujungnya berwarna hijau, hijau keputihan atau kuning kecoklatan, licin dan mengkilat. Akar aerial ini mempunyai lapisan sel atau jaringan yang disebut *velamen* yang bersifat *spongy* (berongga). Jaringan tersebut berfungsi untuk memudahkan akar menyerap air hujan yang jatuh pada kulit pohon inang dan membasahi akar udara. Akar udara berperan dalam proses pertumbuhan dan perkembangan tanaman karena berkemampuan menyerap unsur hara dan sebagai alat pernafasan anggrek (Utami *dkk*, 2007).

Anggrek bulan memiliki karakter tumbuh monopodial, sehingga tidak menghasilkan anakan ke samping. Dalam hal ini, perbanyakan *Phalaenopsis* akan lebih efektif dilakukan secara generatif daripada vegetatif. Proses perkecambahan biji dilakukan di laboratorium, yaitu dalam medium agar buatan yang dilakukan secara steril (Puspitaningtyas, 2010).

## II.2.2. Penyakit pada Tanaman Anggrek *Phalaenopsis*

Tabel II.1. Penyakit pada Tanaman Anggrek *Phalaenopsis*

No	Nama Penyakit	Penyebab	Gejala
1	Busuk Pucuk (Crown Rot)	<i>Cendawan Phytophthora Palmivora</i>	<ul style="list-style-type: none"> <li>- Perubahan warna pangkal daun.</li> <li>- Akar mudah membusuk.</li> <li>- Pangkal daun mudah merambat kebawah.</li> <li>- Pangkal daun mudah mati.</li> <li>- Daun mudah menguning.</li> <li>- Umbi batang mudah berkeriput.</li> <li>- Daun mudah rontok dan mati.</li> </ul>
2	Busuk Lunak	<i>Bakteri Erwinia Caratovora</i>	<ul style="list-style-type: none"> <li>- Muncul bintik kecil berwarna pada daun.</li> <li>- daun menjadi lunak.</li> <li>- daun berair</li> <li>- turgornya hilang</li> <li>- munculnya bintik berwarna kuning dan mudah mati</li> </ul>
3	Busuk Hitam	<i>Cendawan Phytophthora Omnivora</i>	<ul style="list-style-type: none"> <li>- Muncul warna kehitaman pada daun.</li> <li>- Muncul bercak kuning pada daun.</li> <li>- Pangkal daun mudah membusuk dan lunak.</li> <li>- Muncul cendawan berupa air pada pucuk daun.</li> <li>- Akar pada daun mudah keriput dan putus.</li> </ul>
4	Petal Blight	<i>Cendawan Botrytis</i>	<ul style="list-style-type: none"> <li>- Muncul bundaran-bundaran kecil berwarna kecoklatan pada petal bunga.</li> <li>- titik-titik Hitam muncul di sepal bunga.</li> </ul>
5	Bercak Coklat	<i>Pseudomonas Cattleya</i>	<ul style="list-style-type: none"> <li>- Muncul bercak kecil bening pada pucuk pucuk.</li> <li>- Mudah berkecambah menjadi lunak dan mati.</li> <li>- Ada nya cairan pada pucuk daun.</li> </ul>

			- Mudahnya pucuk daun mengalami penularan pada akar daun.
--	--	--	---

( Sumber Haryani,dkk ; 1991 )

### II.3. Metode Dempster Shafer

Teori *Dempster-Shafer* adalah teori yang mampu menangani berbagai kemungkinan yang mengkombinasikan satu kemungkinan dengan fakta yang ada. Dalam *Dempster-Shafer Theory* (DST) ada berbagai konflik yang dipersatukan untuk mengkombinasikan dari berbagai informasi yang ada. Kumpulan informasi yang bersifat berbeda dan menyeluruh dalam teori ini dikenal dengan *frame discernment* yang dinotasikan dengan  $q$  ( $\theta$ ) (Hartati, 2008).

Teori *dempster-Shafer* ini melakukan pembuktian berdasarkan *belief function and plausible reasoning* (fungsi kepercayaan dan pemikiran yang masuk akal) yang digunakan untuk mengkombinasikan potongan informasi yang terpisah untuk mengkalkulasikan kemungkinan dari suatu peristiwa. Teori ini dikembangkan oleh Arthur P. Demster dan Glenn Shafer. Secara umum teori *Dempster-Shafer* ditulis dalam suatu interval :

#### [*Belief, Plausibility*]

*Belief* (bel) adalah ukuran kekuatan *evidence* (bukti) dalam mendukung suatu himpunan proposisi. Jika bernilai 0 maka mengidentifikasi bahwa tidak ada *evidence* dan jika bernilai 1 menunjukkan adanya kepastian.

*Plausibility* (PI) dinotasikan sebagai :

$$PI(s) = 1 - Bel (\sim s) \dots\dots\dots [2.1]$$

*Plausibility* juga bernilai 0 sampai 1. Jika kita yakin  $\sim s$ , maka dapat dikatakan bahwa

$$\text{Bel}(s) = 1 \text{ dan } \text{PI}(\sim s) = 0.$$

$$\text{Misalkan } q = \{A, F, D, B\}$$

Dengan :

A = Alergi

F = Flue

D = Demam

F = Bronkitis

Tujuannya adalah untuk mengkaitkan ukuran kepercayaan elemen-elemen dari  $q$ . Tidak semua evidence secara langsung mendukung tiap-tiap elemen. Untuk itu perlu adanya probabilitas fungsi densitas ( $m$ ). Nilai  $m$  tidak hanya mendefinisikan elemen-elemen  $q$  saja, tetapi juga semua himpunan bagianya (sub-set). Sehingga jika  $q$  berisi  $n$  elemen, maka sub-set dari  $q$  berjumlah  $2^n$ .

Selanjutnya harus ditunjukkan bahwa jumlah semua densitas ( $m$ ) dalam sub-set  $q$  sama dengan 1. Andaikan tidak ada informasi apapun untuk memilih keempat hipotesis tersebut, maka nilai dari :

$$m\{q\} = 1, 0$$

Jika kemudian diketahui bahwa panas merupakan gejala dari Flue, Demam dan Bronkitis dengan  $m = 0,8$  maka :

$$m\{F, D, B\} = 0,8$$

$$m\{q\} = 1 - 0,8 = 0,2$$

Andaikan diketahui  $X$  adalah sub-set dari  $q$  dengan  $m_1$  sebagai fungsi densitasnya, dan  $Y$  juga merupakan sub-set dari  $q$  dengan  $m_2$  sebagai fungsi densitasnya, maka dapat dibentuk suatu fungsi kombinasi  $m_1$  dan  $m_2$  sebagai  $m_3$ , yaitu :

Dimana :

$$m_3(Z) = \frac{\sum_{x \cap y = z} m_1(X) \cdot m_2(Y)}{1 - K} \quad \dots\dots\dots[2.2]$$

Keterangan :

1.  $m_1, m_2, m_3$  = densitas gejala
2.  $X, Y, Z$  = himpunan penyakit
3.  $K$  = jumlah *conflict evidence*

Contoh :

Vany mengalami gejala panas badan. Dari diagnosa dokter kemungkinan Vany menderita Flue, Demam atau Bronkitis. Tunjukkan kaitan ukuran kepercayaan dari

elemen-elemen yang ada !

- a. Gejala 1: panas

Apabila diketahui nilai kepercayaan setelah dilakukan observasi panas sebagai gejala Flue, Demam dan Bronkitis adalah :

$$m_1\{F,D,B\} = 0,8$$

$$m_1\{q\} = 1 - 0,8 = 0,2.$$

Sehari kemudian Vany datang ke dokter lagi dengan gejala hidung buntu.

- b. Gejala 2: hidung buntu

Setelah observasi diketahui bahwa nilai kepercayaan hidung buntu sebagai gejala Alergi, Flue dan Demam adalah :

$$m_2\{A, F, D\} = 0,9$$

$$m_2\{q\} = 1 - 0,9 = 0,1$$

Munculnya gejala baru maka harus dihitung densitas baru untuk beberapa kombinasi ( $m_3$ ). Untuk memudahkan perhitungan maka himpunan-himpunan bagian dibawa ke bentuk table berikut ini :

**Tabel II.2. Aturan Kombinasi untuk  $m_3$**

	{A,F,D} (0,9)	q (0,1)
{F,D,B} (0,8)	{F,D} (0,72)	{F,D,B} (0,08)
q (0,2)	{A,F,D} (0,18)	q (0,02)

Keterangan :

- Kolom pertama berisikan himpunan bagian pada gejala pertama (panas) dengan  $m_1$  sebagai fungsi densitasnya.
- Baris pertama berisikan semua himpunan bagian gejala kedua (hidung buntu), dengan  $m_2$  sebagai fungsi densitasnya
- Baris kedua dan ketiga pada kolom kedua merupakan irisan dari kedua himpunan.

Selanjutnya dihitung densitas baru untuk beberapa kombinasi ( $m_3$ ) dengan persamaan *Dempster-Shafer*, sebagai berikut :

$$m_3 \{ F, D \} = \frac{0,72}{1-0} = 0,72$$

$$m_3 \{ A, F, D \} = \frac{0,18}{1-0} = 0,18$$

$$m_3 \{ F, D, B \} = \frac{0,08}{1-0} = 0,08$$

$$m_3 \{ q \} = \frac{0,02}{1-0} = 0,02$$

Keterangan :

- Terlihat bahwa pada mulanya dengan hanya gejala panas,  $m\{F,D,B\} = 0,8$ . Namun setelah ada gejala baru (hidung buntu), maka nilai  $m\{F,D,B\} = 0,08$ .
- Demikian pula pada mulanya hanya dengan gejala hidung buntu,  $m\{A,F,D\} = 0,9$ . Namun setelah ada gejala baru (panas) maka  $m\{A,F,D\} = 0,18$ .
- Dengan adanya 2 gejala tersebut, maka nilai densitas yang paling kuat adalah  $m\{F,D\} = 0,72$ .

Dari hasil perhitungan di atas dapat disimpulkan bahwa dengan dua gejala yang dialami vany kemungkinan terbesar vany terkena Flue dan Demam.

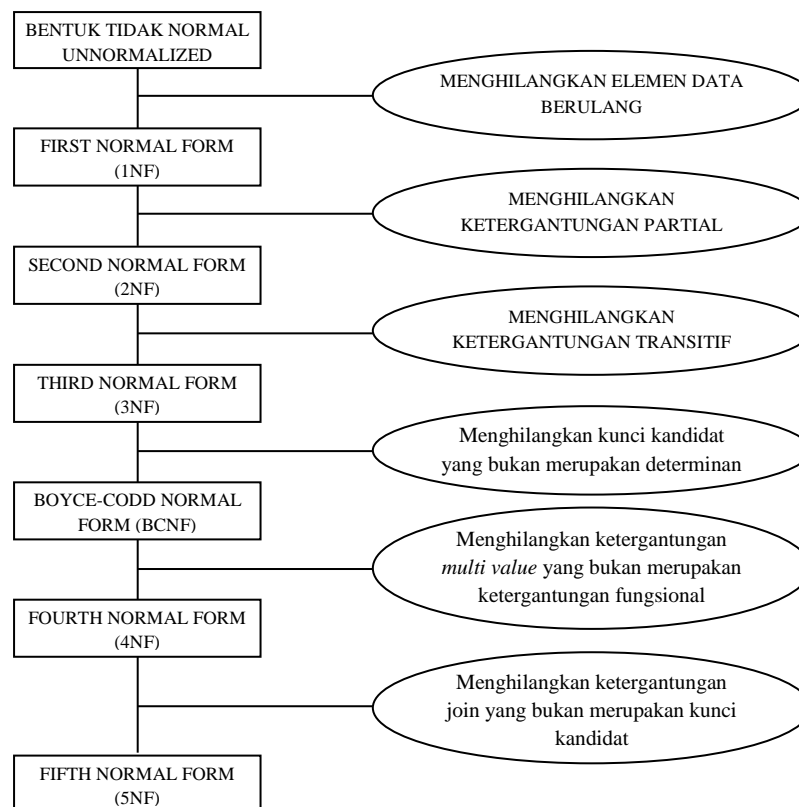
#### II.4. Normalisasi

Istilah normalisasi berasal dari E. F. Codd, salah seorang perintis teknologi basis data. Selain dipakai sebagai metodologi tersendiri untuk menciptakan struktur tabel (relasi) dalam basis data (dengan tujuan untuk mengurangi kemubaziran data), normalisasi terkadang hanya dipakai sebagai perangkat verifikasi terhadap tabel-tabel yang dihasilkan oleh metodologi lain. Normalisasi memberikan panduan yang sangat membantu bagi pengembang untuk mencegah

penciptaan struktur tabel yang kurang fleksibel atau mengurangi ketidakefisienan (Kadir, 2003 : 65).

### 1. Langkah-langkah pembentukan normalisasi

Langkah-langkah pembentukan normalisasi dapat dilihat pada gambar II.2. berikut ini beserta penjelasannya.



**Gambar II.2. Langkah-langkah pembentukan normalisasi**

*(Sumber : Linda Marlinda, 2004 : 121)*

Keterangan gambar di atas dapat dijelaskan sebagai berikut (Marlinda, 2003 : 122-125).

**a. Bentuk tidak normal (*Unnormalized Form*)**

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti suatu format tertentu. Dapat saja data tidak lengkap atau terduplikasi. Data dikumpulkan apa adanya sesuai dengan saat menginput.

**b. Bentuk normal kesatu (1 NF/*First Normal Form*)**

Suatu relasi 1 NF jika dan hanya jika sifat dan setiap relasi atributnya bersifat atomik. Atom adalah zat terkecil yang masih memiliki sifat induknya, bila dipecah lagi maka ia tidak memiliki sifat induknya.

**c. Bentuk normal kedua (2 NF/*Second Normal Form*)**

Bentuk normal kedua mempunyai syarat yaitu bentuk data telah memenuhi kriteria bentuk normal kesatu. Atribut bukan kunci haruslah bergantung secara fungsi pada *primary key*. Jadi, untuk membentuk normal kedua haruslah sudah ditemukan kunci-kunci field. Kunci field haruslah unik dan dapat mewakili atribut lain yang menjadi anggotanya.

**d. Bentuk normal ketiga (3 NF/*Third Normal Form*)**

Untuk menjadi bentuk normal ketiga maka relasi haruslah dalam bentuk normal kedua dan semua atribut bukan primer tidak punya hubungan yang transitif. Dengan kata lain, setiap atribut bukan kunci haruslah bergantung hanya pada *primary key* dan pada *primary key* secara menyeluruh.

**e. *Boyce-Codd Normal Form* (BCNF)**

BCNF mempunyai paksaan lebih kuat dan bentuk normal ketiga. Untuk menjadi BCNF, relasi harus dalam bentuk normal kesatu dan setiap atribut harus bergantung fungsi pada atribut *superkey*.

**f. Bentuk normal keempat (4 NF/*Fourth Normal Form*)**

Relasi R adalah bentuk 4 NF jika dan hanya jika relasi tersebut juga termasuk BCNF dan semua ketergantungan *multi value* adalah juga ketergantungan fungsional.

**g. Bentuk normal kelima (5 NF/*Fifth Normal Form*)**

Disebut juga PJNF (*Projection Join Normal Form*) dan 4 NF dilakukan dengan menghilangkan join yang bukan merupakan kunci kandidat.

## **II.5. PHP Hypertext Preprocessor (PHP)**

Menurut Sibero (2013 : 49), PHP adalah pemrograman *interpreter* yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan. PHP disebut sebagai pemrograman *Server Side Programming*, hal ini dikarenakan seluruh prosesnya dijalankan pada *server*. PHP adalah suatu bahasa dengan hak cipta terbuka atau yang juga dikenal dengan istilah *Open Source*, yaitu pengguna dapat mengembangkan kode-kode fungsi PHP sesuai dengan kebutuhannya.

## **II.6. Database MySQL**

MySQL atau dibaca "*My Sekuel*" dengan adalah suatu RDBMS (*Relational Database Management System*) yaitu aplikasi sistem yang menjalankan fungsi pengolahan data. MySQL pertama kali dikembangkan oleh MySQL AB yang kemudian diakuisisi Sun Microsystem dan terakhir dikelola oleh Oracle Corporation. (Sibero, 2013 : 97).

## **II.7. Adobe Dreamweaver**

*Adobe Dreamweaver* adalah suatu produk *Web Developer* yang dikembangkan oleh *Adobe Systems Inc.*, sebelumnya produk *Dreamweaver* dikembangkan oleh *Macromedia Inc.* yang kemudian sampai dengan saat ini pengembangannya diteruskan oleh *Adobe Systems Inc.* Setelah diambil alih oleh *Adobe Systems Inc.*, *Dreamweaver* dikembangkan dan dirilis dengan kode nama *Creative Suite (CS)*.

Ruang Kerja atau *Workspace* adalah bagian keseluruhan tampilan dari *Adobe Dreamweaver*. Ruang kerja *Dreamweaver* terdiri dari *Welcome Screen*, *Menu*, *Insert bar*, *Document Window*, *CSS Panel*, *Application Panel*, *Tag Inspector*, *Property Inspector*, *Result Panel* dan *Files Panel*. Masing-masing dari komponen tersebut memiliki fungsi dan aturan. (Sibero, 2013 : 384).

## **II.8. Unified Modeling Language (UML)**

Menurut Windu Gata (2013 : 4) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language (UML)*. UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

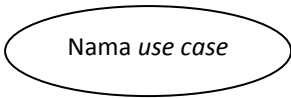
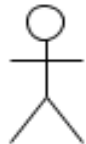

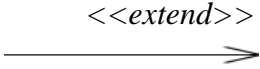
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

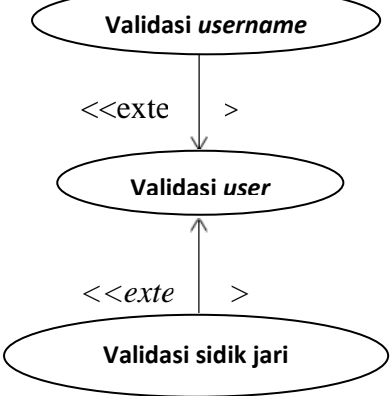

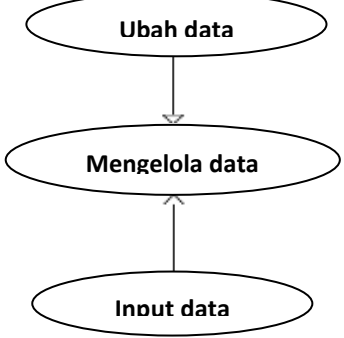

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

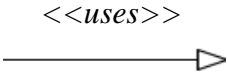
### 1. *Use case* Diagram

*Use case* diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut ini adalah simbol-simbol yang digunakan dalam *use case* diagram, yaitu :

**Tabel II.3. Simbol – Simbol *Use Case* Diagram**

<b>Simbol</b>	<b>Deskripsi</b>
<p><i>Use case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal diawal frase nama <i>use case</i></p>
<p>Aktor / <i>actor</i></p>  <p><b>nama aktor</b></p>	<p>orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari <i>actor</i> adalah gambar orang, tapi <i>aktor</i> belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal nama <i>aktor</i>.</p>
<p>Asosiasi/ <i>association</i></p> 	<p>Komunikasi antara <i>actor</i> dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor</p>
<p>Ekstensi / <i>extend</i></p> 	<p>Relasi <i>use case</i> tambahan kesebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu: mirip dengan prinsip <i>inheritance</i> pada pemograman berorientasi objek; biasanya <i>use</i></p>

	<p><i>case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal</p>  <pre> graph TD     A([Validasi username]) -- &lt;&lt;exte&gt;&gt; B([Validasi user])     C([Validasi sidik jari]) -- &lt;&lt;exte&gt;&gt; B   </pre> <p>Arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>
<p>Generalisasi/<i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:</p>  <pre> graph TD     A([Ubah data])     B([Mengelola data]) --&gt; A     C([Inout data]) --&gt; B   </pre> <p>Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).</p>
<p>Menggunakan / <i>include</i> / <i>uses</i></p> <p>&lt;&lt;<i>include</i>&gt;&gt;</p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat <i>use case</i> ini.</p> <p>Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i> :</p> <p><i>include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan</p>



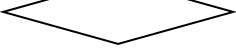

	<p>dijalankan.</p> <p><i>uses</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan.</p>
---	--


Sumber : (Rosa A.S,dkk ; 2013 : 155-158)

## 2. Diagram Aktivitas (*Activity Diagram*)

*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Berikut ini adalah simbol-simbol yang digunakan dalam *activity diagram*, yaitu :

**Tabel II.4. Simbol - Simbol *Activity Diagram***

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

<p>Status akhir</p> 	<p>Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.</p>
<p>Swimlane</p> <div style="border: 1px solid black; padding: 2px; width: fit-content;"> <p>nama swimlane</p> </div>	<p>Memisahkan organisasi bisnis yang bertanggungjawab terhadap aktivitas yang terjadi.</p>

Sumber : (Rosa A.S, dkk ; 2013 : 161-163)

### 3. Diagram Kelas (*Class Diagram*)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

*Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

Tabel II.5. *Multiplicity Class Diagram*

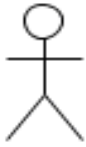

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4





(Sumber : Windu Gata ; 2013 : 9)

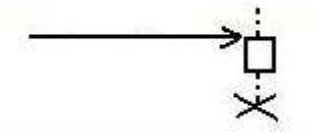
4. Diagram Urutan (*Sequence Diagram*)

*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

Tabel II.6. Simbol *Sequence Diagram*

Simbol	Deskripsi
<p>Aktor</p>  <p>nama aktor</p>	<p>orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari <i>actor</i> adalah gambar orang, tapi <i>actor</i> belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal nama <i>actor</i>.</p>
<p>Garis hidup / <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek.</p>

<p>Objek</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <u>nama objek : nama kelas</u> </div>	<p>Menyatakan objek yang berinteraksi pesan.</p>
<p>Waktu aktif</p> <div style="border: 1px solid black; width: 20px; height: 40px; margin: 10px auto;"></div>	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah tahapan yang dilakukan didalamnya.</p>
<p>Pesan tipe create</p> <div style="text-align: center; margin: 10px auto;"> <p>&lt;&lt;create&gt;&gt;</p>  </div>	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>
<p>Pesan tipe call</p> <div style="text-align: center; margin: 10px auto;"> <p>1 : nama_metode{ }</p>  </div>	<p>Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri.</p>
<p>Pesan tipe send</p> <div style="text-align: center; margin: 10px auto;"> <p>1 : masukan</p>  </div>	<p>Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>
<p>Pesan tipe return</p> <div style="text-align: center; margin: 10px auto;"> <p>1 : keluaran</p>  </div>	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>

<p>Pesan tipe destroy</p> <p><code>&lt;&lt;destroy&gt;&gt;</code></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada <i>destroy</i>.</p>
---	--

Sumber : (Rosa A.S, dkk ; 2013 : 165-167)

## **BAB II**

### **LANDASAN TEORI**

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

*Artificial intelligence* (AI) memiliki beberapa domain masalah atau area. Bidang sistem pakar merupakan penyelesaian pendekatan yang sangat berhasil dan bagus untuk permasalahan AI klasik dari pemrograman *intelligent* (cerdas).

Sistem pakar (*expert system*) merupakan solusi AI bagi masalah pemrograman pintar (*intelligent*). Profesor Edward Feigenbaum dari Stanford University yang merupakan pionir dalam teknologi sistem pakar mendefinisikan sistem pakar sebagai sebuah program komputer 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.

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 (Rosnelly, 2013 : 2).

### II.1.1 Kelebihan Sistem Pakar

Sistem pakar memiliki beberapa fitur menarik yang merupakan kelebihannya (Rosnelly, 2013 : 5-6), seperti :

13. Meningkatkan ketersediaan (*increased availability*). Kepakaran atau keahlian menjadi tersedia dalam sistem komputer. Dapat dikatakan bahwa sistem pakar merupakan produksi kepakaran secara masal (*massproduction*).
14. Mengurangi biaya (*reduce cost*). Biaya yang diperlukan untuk menyediakan keahlian per satu orang *user* menjadi berkurang.
15. Mengurangi bahaya (*reduce danger*). Sistem pakar dapat digunakan di lingkungan yang mungkin berbahaya bagi manusia.
16. Permanen (*permanence*). Sistem pakar dan pengetahuan yang terdapat di dalamnya bersifat lebih permanen dibandingkan manusia yang dapat merasa lelah, bosan dan pengetahuannya hilang saat pakar meninggal dunia.
17. Keahlian multiple (*multiple expertise*). Pengetahuan dari beberapa pakar dapat dimuat ke dalam sistem dan bekerja secara simultan dan kontinyu menyelesaikan suatu masalah setiap saat. Tingkat keahlian atau pengetahuan yang digabungkan dari beberapa pakar dapat melebihi pengetahuan satu orang pakar.
18. Meningkatkan kehandalan (*increased reliability*). Sistem pakar meningkatkan kepercayaan dengan memberikan hasil yang benar sebagai alternatif pendapat dari seorang pakar atau sebagai penengah
19. jika terjadi konflik antara beberapa pakar. Namun hal tersebut tidak berlaku jika sistem dibuat oleh salah seorang pakar, sehingga akan selalu sama

dengan pendapat pakar tersebut kecuali jika sang pakar melakukan kesalahan yang mungkin terjadi pada saat tertekan.

20. Penjelasan (*explanation*). Sistem pakar dapat menjelaskan detail proses penalaran (*reasoning*) yang dilakukan hingga mencapai suatu kesimpulan. Seorang pakar mungkin saja terlalu lelah, tidak bersedia atau tidak mampu melakukannya setiap waktu. Hal ini akan meningkatkan tingkat percayaan bahwa kesimpulan yang dihasilkan adalah benar.

21. Respon yang cepat (*fast response*). Respon yang cepat atau *real time* diperlukan pada beberapa aplikasi. Meskipun bergantung pada *hardware* dan *software* yang digunakan, namun sistem pakar relatif memberikan respon yang lebih cepat dibandingkan seorang pakar.

22. Stabil, tidak emosional dan memberikan respon yang lengkap setiap saat (*steady, unemotional and complete response at all times*). Karakteristik ini diperlukan pada situasi *real time* dan keadaan darurat (*emergency*) ketika seorang pakar mungkin tidak berada pada kondisi puncak disebabkan oleh stres atau kelelahan.

23. Pembimbing pintar (*intelligent tutor*). Sistem pakar dapat berperan sebagai *intelligent tutor* dengan memberikan kesempatan pada *user* untuk menjalankan contoh program dan menjelaskan proses *reasoning* yang dilakukan.

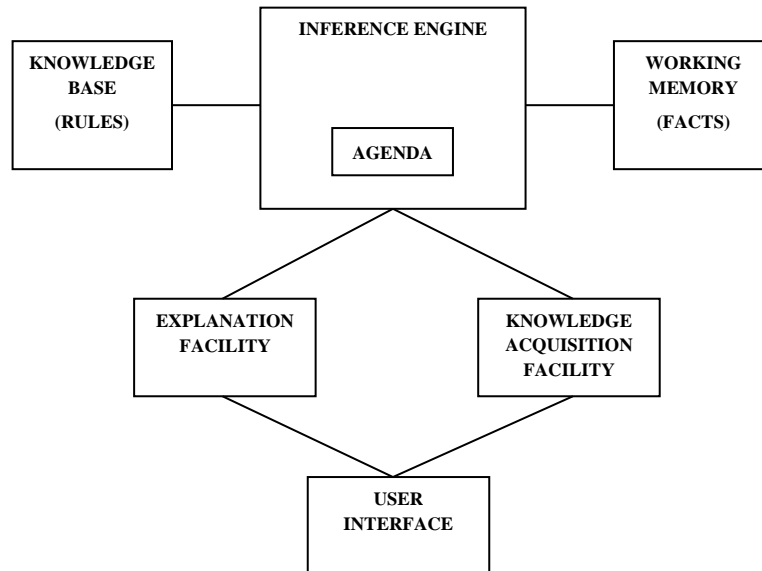
24. Basis data cerdas (*intelligent database*). Sistem pakar dapat digunakan untuk mengakses basis data secara cerdas.

### II.1.2. Konsep Umum Sistem Pakar

Sistem pakar umumnya dirancang dengan cara yang berbeda dengan sistem konvensional lain, terutama karena masalah yang dihadapi umumnya tidak memiliki solusi algoritmik dan bergantung pada inferensi untuk mendapatkan solusi yang terbaik yang paling mungkin. Oleh karena itu sistem pakar harus mampu menjelaskan inferensi yang dilakukannya sehingga hasil yang diperoleh dapat diperiksa. *Explanation facility* merupakan bagian terintegrasi dari sebuah sistem pakar. Sebuah *explanation facility* yang detail dirancang untuk memungkinkan *user* mengeksplorasi *rules* melalui tipe pertanyaan “*what if*” yang disebut *hypothetical reasoning* dan bahkan menerjemahkan *natural language* ke dalam *rules* dengan cara induksi dari tabel data. (Rosnelly, 2013 : 8).

### II.1.3. Struktur Sistem Pakar

Adapun struktur sistem pakar dapat dilihat pada gambar II.1. berikut ini :



**Gambar II.1. Struktur Sistem Pakar**  
(Sumber : Rika Rosnelly, 2013 : 13)

Komponen yang terdapat dalam struktur sistem pakar ini adalah *knowledge base (rules)*, *inference engine*, *working memory*, *explanation facility*, *knowledge acquisition facility*, *user interface*. (Rosnelly, 2013 : 13-15).

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

Basis pengetahuan mengandung pengetahuan untuk pemahaman, formulasi dan penyelesaian masalah. Komponen sistem pakar disusun atas dua elemen dasar, yaitu fakta dan aturan. Fakta merupakan informasi tentang objek dalam area permasalahan tertentu, sedangkan aturan merupakan informasi tentang cara bagaimana memperoleh fakta baru dari fakta yang telah diketahui.

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

Mesin inferensi merupakan otak dari sebuah sistem pakar dan dikenal juga dengan sebutan *control structure* (struktur kontrol) atau *rule interpreter* (dalam sistem pakar berbasis kaidah). Komponen ini mengandung mekanisme pola pikir dan penalaran yang digunakan oleh pakar dalam menyelesaikan suatu masalah. Mesin inferensi disini adalah *processor* pada sistem pakar yang mencocokkan bagian kondisi dari *rule* yang tersimpan di dalam *knowledge base* dengan fakta yang tersimpan di *working memory*.

#### 9. *Working Memory*

Berguna untuk menyimpan fakta yang dihasilkan oleh *inference engine* dengan penambahan parameter berupa derajat kepercayaan atau dapat juga dikatakan sebagai global database dari fakta yang digunakan oleh rule-rule yang ada.

#### 10. *Explanation Facility*

Menyediakan kebenaran dari solusi yang dihasilkan kepada user (*reasoning chain*).

#### 11. *Knowledge Acquisition Facility*

Meliputi proses pengumpulan, pemindahan dan perubahan dari kemampuan pemecahan masalah seorang pakar atau sumber pengetahuan terdokumentasi pemecahan ke program komputer, yang bertujuan untuk memperbaiki atau mengembanglan basis pengetahuan.

## 12. *User Interface*

Mekanisme untuk memberi kesempatan kepada user dan sistem pakar untuk berkomunikasi. Antar muka menerima informasi dari pemakai dan mengubahnya ke dalam bentuk yang dapat diterima oleh sistem. Selain itu, antar muka menerima informasi dari sistem dan menyajikannya ke dalam bentuk yang dapat dimengerti oleh pemakai.

### II.1.4. Karakteristik Sistem Pakar

Menurut Rosnelly (2013 : 20-21), sistem pakar umumnya dirancang untuk memenuhi beberapa karakteristik umum berikut ini :

- f. **Kinerja sangat baik** (*high performance*). Sistem harus mampu memberikan respon berupa saran (*advice*) dengan tingkat kualitas yang sama dengan seorang pakar atau melebihinya.
- g. **Waktu respon yang baik** (*adequate respon 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*).
- h. **Dapat diandalkan** (*good reliabilty*). Sistem harus dapat diandalkan dan tidak mudah rusak/*crash*.
- i. **Dapat dipahami** (*understandable*). Sistem harus mampu menjelaskan langkah-langkah penalaran yang dilakukannya seperti seorang pakar. Hal ini penting untuk beberapa alasan, yaitu :

- 3) Dimungkinkan bahwa sistem pakar berkaitan dengan nyawa manusia atau properti lainnya sehingga harus dapat menjelaskan mengapa dihasilkan suatu kesimpulan tertentu.
  - 4) Untuk mengkonfirmasi bahwa pengetahuan pakar telah dikumpulkan dengan benar dan digunakan oleh sistem dengan benar pula. Hal ini penting dalam proses debugging pengetahuan yang mungkin salah karena pengetikan atau pemahaman yang salah dari *knowledge engineer*.
- j. **Fleksibel** (*flexibility*). Sistem harus menyediakan mekanisme untuk menambah dan menghapus pengetahuan.

## II.2. Tanaman Anggrek *Phalaenopsis* (*Phalaenopsis amabilis*)

Anggrek sebagai tanaman hias pot sangat diminati masyarakat karena keindahan dan keunikan bunganya. Keindahan dan keunikan bunga anggrek dapat dilihat dari variasi warna, ukuran dan bentuk bunga yang dihasilkan. Hal lain juga, beberapa jenis anggrek mengeluarkan aroma wangi. Keunikan semakin bernilai tinggi dijumpai pada jenis-jenis anggrek spesies. Anggrek bulan (*Phalaenopsis amabilis*) merupakan jenis anggrek (*Orchidaceae*) yang mempunyai ciri khas kelopak bunga yang lebar dan berwarna putih. Meskipun saat ini sudah banyak anggrek bulan hasil persilangan ( anggrek bulan hibrida ) yang memiliki corak dan warna beragam jenis ( Yulia Andiani; 2008:21 ).

Anggrek *Phalaenopsis* termasuk tanaman monopodial yaitu pertumbuhan tanaman vertikal pada satu titik tumbuh dan hanya satu batang utama. Ia hidup sebagai tanaman epifit yaitu hidup menumpang pada tanaman lain tetapi tidak

mengambil makanan dari tanaman yang ditumpangi atau sebagai tanaman litofit yang tumbuh diatas batuan bukit karang yang sudah lapuk dan kadang berlumut.

Tanaman yang memiliki bunga indah ini berbatang pendek dan tidak menghasilkan umbi palsu seperti tanaman anggrek lainnya. Akar *Phalaenopsis* adalah yang paling menarik diantara anggrek lainnya, karena dalam kehidupan liarnya akan bergelayut erat pada tanaman yang di tumpangi. Akarnya seakan berdaging dan tumbuh liar.

### II.2.1. Klasifikasi dan Morfologi Tanaman Anggrek *Phalaenopsis*

#### 3. Klasifikasi Tanaman Anggrek *Phalaenopsis*

Kedudukan tanaman anggrek bulan dalam sistematika (*taksonomi*) tumbuhan dapat diklasifikasikan sebagai berikut :

Kerajaan	: <i>Plantae</i>
Divisi	: <i>Spermatophyta</i>
Sub divisi	: <i>Angiospermae</i>
Kelas	: <i>Monocotyledonae</i>
Bangsa	: <i>Orchidales</i>
Suku	: <i>Orchidaceae</i>
Marga	: <i>Phalaenopsis</i>
Jenis	: <i>Phalaenopsis</i> sp.

#### 4. Morfologi Tanaman Anggrek *Phalaenopsis*

Susunan tubuh tanaman anggrek bulan terdiri dari bunga, buah, biji, daun, batang, dan akar.

##### e. Bunga Anggrek Bulan

Bunga anggrek bulan tersusun dalam tandan dan kadang-kadang bercabang dengan panjang karangan bunga mencapai 50 cm yang tumbuh menjuntai. Setiap tangkai mendukung 10-12 kuntum bunga dengan daun penumpu 5 mm berbentuk segitiga, bunganya cukup harum dan waktu mekarnya lama. Perhiasan bunga tersusun membulat dengan diameter 6-10 cm atau lebih dan mahkotanya bertumpang tindih dengan kelopak tersusun membundar (Puspitaningtyas, 2010).

Bentuk bunga anggrek *Phalaenopsis* ada dua, yaitu bulat (*round shape*) dan bintang (*star*). Bunga anggrek terdiri dari kelopak (*sepal*), mahkota (*petal*), dan lidah (*labelum*). Sepal yang dimiliki anggrek terdiri atas tiga helai dan tiga helai petal yang salah satu petal berubah menjadi bibir bunga atau labelum. Selain itu, terdapat bagian lain yang disebut tugu, yaitu perpanjangan gagang bunga (bakal buah), dibentuk oleh penyatuan putik dan benang sari (Kencana, 2007).

Warna bunga putih bersih dengan sedikit variasi kuning dan bintik kemerahan di bibir bunga. Bunga anggrek *Phalaenopsis* juga memiliki motif yang beragam diantaranya motif titik-titik, garis-garis, blok dan sembur (*splash*). Susunan bunganya sangat artistik, tersusun rapi, menjuntai ke bawah, dan berselang-seling (Setiawan, 2005). Bibir kedua cuping samping

tegak melebar dan bagian tepi depannya berwarna kuning dengan garis kemerahan. Buah berbentuk bulat lonjong, berukuran 7,5 x 1,3 cm (Puspitaningtyas, 2010).

f. Buah dan Biji Anggrek Bulan (*Phalaenopsis*)

Bentuk buah anggrek merupakan lentera atau *capsular* yang memiliki 6 rusuk. Tiga diantaranya merupakan rusuk sejati dan tiga rusuk yang lain merupakan tempat melekatnya dua tepi daun buah yang berlainan. Buah anggrek yang mencapai besarnya jari kelingking memiliki ratusan ribu bahkan jutaan biji anggrek yang sangat lembut dalam ukuran yang sangat kecil di dalamnya. Biji-biji anggrek tidak memiliki endosperm sebagai cadangan makanan, sehingga untuk perkecambahannya dibutuhkan nutrisi yang berfungsi membantu pertumbuhan biji.

g. Daun dan Batang Anggrek Bulan (*Phalaenopsis*)

Anggrek bulan termasuk anggrek epifit monopodial yang tumbuh menjuntai. Batangnya sangat pendek dan terbungkus oleh seludang daun. Daunnya berjumlah kurang dari lima helai, berwarna hijau, tebal, berdaging, berbentuk lonjong bulat telur sungsang atau jorong, melebar di bagian ujungnya, berujung tumpul, atau sedikit meruncing, dengan panjang 20-30 cm dan lebar 5-8 cm.

h. Akar Anggrek Bulan (*Phalaenopsis*)

Akar anggrek bulan berbentuk bulat memanjang serta berdaging, bercabang, berwarna putih dan hijau di bagian ujungnya (Puspitaningtyas, 2010). Menurut Rukmana (2008), akar tanaman anggrek bulan terdiri dari dua

macam yaitu akar lekat dan akar udara (*aerial*). Akar lekat berfungsi untuk melekat dan menahan keseluruhan tanaman agar tetap berada pada posisinya. Bagian ujung akar meruncing dan sedikit lengket, dalam keadaan kering, akar tampak berwarna putih keperak-perakan dan hanya bagian ujung akar saja berwarna hijau atau tampak keunguan. Akar yang sudah tua akan berwarna coklat tua dan kering. Akar udara atau akar aerial merupakan akar yang keluar dari batang atas. Akar udara atau akar aerial yang tidak melekat pada batang pohon tidak ditumbuhi rambut akar. Akar aerial yang masih aktif ujungnya berwarna hijau, hijau keputihan atau kuning kecoklatan, licin dan mengkilat. Akar aerial ini mempunyai lapisan sel atau jaringan yang disebut *velamen* yang bersifat *spongy* (berongga). Jaringan tersebut berfungsi untuk memudahkan akar menyerap air hujan yang jatuh pada kulit pohon inang dan membasahi akar udara. Akar udara berperan dalam proses pertumbuhan dan perkembangan tanaman karena berkemampuan menyerap unsur hara dan sebagai alat pernafasan anggrek (Utami *dkk*, 2007).

Anggrek bulan memiliki karakter tumbuh monopodial, sehingga tidak menghasilkan anakan ke samping. Dalam hal ini, perbanyakan *Phalaenopsis* akan lebih efektif dilakukan secara generatif daripada vegetatif. Proses perkecambahan biji dilakukan di laboratorium, yaitu dalam medium agar buatan yang dilakukan secara steril (Puspitaningtyas, 2010).

## II.2.2. Penyakit pada Tanaman Anggrek *Phalaenopsis*

Tabel II.1. Penyakit pada Tanaman Anggrek *Phalaenopsis*

No	Nama Penyakit	Penyebab	Gejala
1	Busuk Pucuk (Crown Rot)	<i>Cendawan Phytophthora Palmivora</i>	<ul style="list-style-type: none"> <li>- Perubahan warna pangkal daun.</li> <li>- Akar mudah membusuk.</li> <li>- Pangkal daun mudah merambat kebawah.</li> <li>- Pangkal daun mudah mati.</li> <li>- Daun mudah menguning.</li> <li>- Umbi batang mudah berkeriput.</li> <li>- Daun mudah rontok dan mati.</li> </ul>
2	Busuk Lunak	<i>Bakteri Erwinia Caratovora</i>	<ul style="list-style-type: none"> <li>- Muncul bintik kecil berwarna pada daun.</li> <li>- daun menjadi lunak.</li> <li>- daun berair</li> <li>- turgornya hilang</li> <li>- munculnya bintik berwarna kuning dan mudah mati</li> </ul>
3	Busuk Hitam	<i>Cendawan Phytophthora Omnivora</i>	<ul style="list-style-type: none"> <li>- Muncul warna kehitaman pada daun.</li> <li>- Muncul bercak kuning pada daun.</li> <li>- Pangkal daun mudah membusuk dan lunak.</li> <li>- Muncul cendawan berupa air pada pucuk daun.</li> <li>- Akar pada daun mudah keriput dan putus.</li> </ul>
4	Petal Blight	<i>Cendawan Botrytis</i>	<ul style="list-style-type: none"> <li>- Muncul bundaran-bundaran kecil berwarna kecoklatan pada petal bunga.</li> <li>- titik-titik Hitam muncul di sepal bunga.</li> </ul>
5	Bercak Coklat	<i>Pseudomonas Cattleya</i>	<ul style="list-style-type: none"> <li>- Muncul bercak kecil bening pada pucuk pucuk.</li> <li>- Mudah berkecambah menjadi lunak dan mati.</li> <li>- Ada nya cairan pada pucuk daun.</li> </ul>

			- Mudahnya pucuk daun mengalami penularan pada akar daun.
--	--	--	---

( Sumber Haryani,dkk ; 1991 )

### II.3. Metode Dempster Shafer

Teori *Dempster-Shafer* adalah teori yang mampu menangani berbagai kemungkinan yang mengkombinasikan satu kemungkinan dengan fakta yang ada. Dalam *Dempster-Shafer Theory* (DST) ada berbagai konflik yang dipersatukan untuk mengkombinasikan dari berbagai informasi yang ada. Kumpulan informasi yang bersifat berbeda dan menyeluruh dalam teori ini dikenal dengan *frame discernment* yang dinotasikan dengan  $q$  ( $\theta$ ) (Hartati, 2008).

Teori *dempster-Shafer* ini melakukan pembuktian berdasarkan *belief function and plausible reasoning* (fungsi kepercayaan dan pemikiran yang masuk akal) yang digunakan untuk mengkombinasikan potongan informasi yang terpisah untuk mengkalkulasikan kemungkinan dari suatu peristiwa. Teori ini dikembangkan oleh Arthur P. Demster dan Glenn Shafer. Secara umum teori *Dempster-Shafer* ditulis dalam suatu interval :

#### [*Belief, Plausibility*]

*Belief* (bel) adalah ukuran kekuatan *evidence* (bukti) dalam mendukung suatu himpunan proposisi. Jika bernilai 0 maka mengidentifikasi bahwa tidak ada *evidence* dan jika bernilai 1 menunjukkan adanya kepastian.

*Plausibility* (PI) dinotasikan sebagai :

$$PI(s) = 1 - Bel (\sim s) \dots\dots\dots [2.1]$$

*Plausibility* juga bernilai 0 sampai 1. Jika kita yakin  $\sim s$ , maka dapat dikatakan bahwa

$$\text{Bel}(s) = 1 \text{ dan } \text{PI}(\sim s) = 0.$$

$$\text{Misalkan } q = \{A, F, D, B\}$$

Dengan :

A = Alergi

F = Flue

D = Demam

F = Bronkitis

Tujuannya adalah untuk mengkaitkan ukuran kepercayaan elemen-elemen dari  $q$ . Tidak semua evidence secara langsung mendukung tiap-tiap elemen. Untuk itu perlu adanya probabilitas fungsi densitas ( $m$ ). Nilai  $m$  tidak hanya mendefinisikan elemen-elemen  $q$  saja, tetapi juga semua himpunan bagianya (sub-set). Sehingga jika  $q$  berisi  $n$  elemen, maka sub-set dari  $q$  berjumlah  $2^n$ .

Selanjutnya harus ditunjukkan bahwa jumlah semua densitas ( $m$ ) dalam sub-set  $q$  sama dengan 1. Andaikan tidak ada informasi apapun untuk memilih keempat hipotesis tersebut, maka nilai dari :

$$m\{q\} = 1, 0$$

Jika kemudian diketahui bahwa panas merupakan gejala dari Flue, Demam dan Bronkitis dengan  $m = 0,8$  maka :

$$m\{F, D, B\} = 0,8$$

$$m\{q\} = 1 - 0,8 = 0,2$$

Andaikan diketahui  $X$  adalah sub-set dari  $q$  dengan  $m_1$  sebagai fungsi densitasnya, dan  $Y$  juga merupakan sub-set dari  $q$  dengan  $m_2$  sebagai fungsi densitasnya, maka dapat dibentuk suatu fungsi kombinasi  $m_1$  dan  $m_2$  sebagai  $m_3$ , yaitu :

Dimana :

$$m_3(Z) = \frac{\sum_{x_1 y=z} m_1(X).m_2(Y)}{1-K} \quad \dots\dots\dots[2.2]$$

Keterangan :

4.  $m_1, m_2, m_3$  = densitas gejala
5.  $X, Y, Z$  = himpunan penyakit
6.  $K$  = jumlah *conflict evidence*

Contoh :

Vany mengalami gejala panas badan. Dari diagnosa dokter kemungkinan Vany menderita Flue, Demam atau Bronkitis. Tunjukkan kaitan ukuran kepercayaan dari

elemen-elemen yang ada !

- c. Gejala 1: panas

Apabila diketahui nilai kepercayaan setelah dilakukan observasi panas sebagai gejala Flue, Demam dan Bronkitis adalah :

$$m_1\{F,D,B\} = 0,8$$

$$m_1\{q\} = 1 - 0,8 = 0,2.$$

Sehari kemudian Vany datang ke dokter lagi dengan gejala hidung buntu.

- d. Gejala 2: hidung buntu

Setelah observasi diketahui bahwa nilai kepercayaan hidung buntu sebagai gejala Alergi, Flue dan Deman adalah :

$$m_2\{A, F, D\} = 0,9$$

$$m_2\{q\} = 1 - 0,9 = 0,1$$

Munculnya gejala baru maka harus dihitung densitas baru untuk beberapa kombinasi ( $m_3$ ). Untuk memudahkan perhitungan maka himpunan-himpunan bagian dibawa ke bentuk table berikut ini :

**Tabel II.2. Aturan Kombinasi untuk  $m_3$**

	{A,F,D} (0,9)	q (0,1)
{F,D,B} (0,8)	{F,D} (0,72)	{F,D,B} (0,08)
q (0,2)	{A,F,D} (0,18)	q (0,02)

Keterangan :

- Kolom pertama berisikan himpunan bagian pada gejala pertama (panas) dengan  $m_1$  sebagai fungsi densitasnya.
- Baris pertama berisikan semua himpunan bagian gejala kedua (hidung buntu), dengan  $m_2$  sebagai fungsi densitasnya
- Baris kedua dan ketiga pada kolom kedua merupakan irisan dari kedua himpunan.

Selanjutnya dihitung densitas baru untuk beberapa kombinasi ( $m_3$ ) dengan persamaan *Dempster-Shafer*, sebagai berikut :

$$m_3 \{ F, D \} = \frac{0,72}{1-0} = 0,72$$

$$m_3 \{ A, F, D \} = \frac{0,18}{1-0} = 0,18$$

$$m_3 \{ F,D,B \} = \frac{0,08}{1-0} = 0,08$$

$$m_3 \{ q \} = \frac{0,02}{1-0} = 0,02$$

Keterangan :

- Terlihat bahwa pada mulanya dengan hanya gejala panas,  $m\{F,D,B\} = 0,8$ . Namun setelah ada gejala baru (hidung buntu), maka nilai  $m\{F,D,B\} = 0,08$ .
- Demikian pula pada mulanya hanya dengan gejala hidung buntu,  $m\{A,F,D\} = 0,9$ . Namun setelah ada gejala baru (panas) maka  $m\{A,F,D\} = 0,18$ .
- Dengan adanya 2 gejala tersebut, maka nilai densitas yang paling kuat adalah  $m\{F,D\} = 0,72$ .

Dari hasil perhitungan di atas dapat disimpulkan bahwa dengan dua gejala yang dialami vany kemungkinan terbesar vany terkena Flue dan Demam.

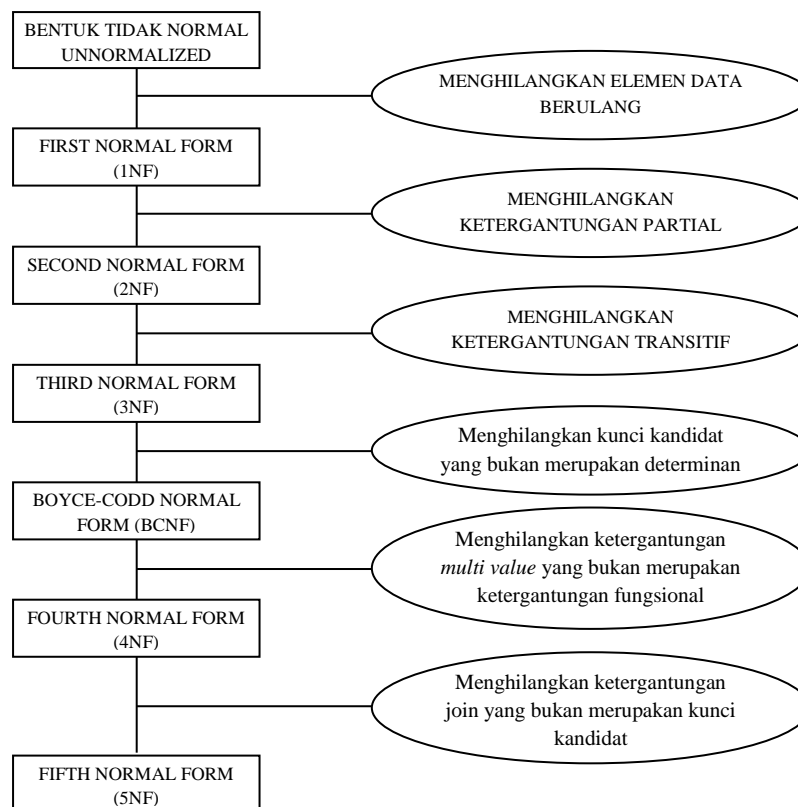
#### II.4. Normalisasi

Istilah normalisasi berasal dari E. F. Codd, salah seorang perintis teknologi basis data. Selain dipakai sebagai metodologi tersendiri untuk menciptakan struktur tabel (relasi) dalam basis data (dengan tujuan untuk mengurangi kemubaziran data), normalisasi terkadang hanya dipakai sebagai perangkat verifikasi terhadap tabel-tabel yang dihasilkan oleh metodologi lain. Normalisasi memberikan panduan yang sangat membantu bagi pengembang untuk mencegah

penciptaan struktur tabel yang kurang fleksibel atau mengurangi ketidakefisienan (Kadir, 2003 : 65).

## 2. Langkah-langkah pembentukan normalisasi

Langkah-langkah pembentukan normalisasi dapat dilihat pada gambar II.2. berikut ini beserta penjelasannya.



**Gambar II.2. Langkah-langkah pembentukan normalisasi**

(Sumber : Linda Marlinda, 2004 : 121)

Keterangan gambar di atas dapat dijelaskan sebagai berikut (Marlinda, 2003 : 122-125).

**h. Bentuk tidak normal (*Unnormalized Form*)**

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti suatu format tertentu. Dapat saja data tidak lengkap atau terduplikasi. Data dikumpulkan apa adanya sesuai dengan saat menginput.

**i. Bentuk normal kesatu (1 NF/*First Normal Form*)**

Suatu relasi 1 NF jika dan hanya jika sifat dan setiap relasi atributnya bersifat atomik. Atom adalah zat terkecil yang masih memiliki sifat induknya, bila dipecah lagi maka ia tidak memiliki sifat induknya.

**j. Bentuk normal kedua (2 NF/*Second Normal Form*)**

Bentuk normal kedua mempunyai syarat yaitu bentuk data telah memenuhi kriteria bentuk normal kesatu. Atribut bukan kunci haruslah bergantung secara fungsi pada *primary key*. Jadi, untuk membentuk normal kedua haruslah sudah ditemukan kunci-kunci field. Kunci field haruslah unik dan dapat mewakili atribut lain yang menjadi anggotanya.

**k. Bentuk normal ketiga (3 NF/*Third Normal Form*)**

Untuk menjadi bentuk normal ketiga maka relasi haruslah dalam bentuk normal kedua dan semua atribut bukan primer tidak punya hubungan yang transitif. Dengan kata lain, setiap atribut bukan kunci haruslah bergantung hanya pada *primary key* dan pada *primary key* secara menyeluruh.

**l. *Boyce-Codd Normal Form* (BCNF)**

BCNF mempunyai paksaan lebih kuat dan bentuk normal ketiga. Untuk menjadi BCNF, relasi harus dalam bentuk normal kesatu dan setiap atribut harus bergantung fungsi pada atribut *superkey*.

**m. Bentuk normal keempat (4 NF/*Fourth Normal Form*)**

Relasi R adalah bentuk 4 NF jika dan hanya jika relasi tersebut juga termasuk BCNF dan semua ketergantungan *multi value* adalah juga ketergantungan fungsional.

**n. Bentuk normal kelima (5 NF/*Fifth Normal Form*)**

Disebut juga PJNF (*Projection Join Normal Form*) dan 4 NF dilakukan dengan menghilangkan join yang bukan merupakan kunci kandidat.

## **II.5. PHP Hypertext Preprocessor (PHP)**

Menurut Sibero (2013 : 49), PHP adalah pemrograman *interpreter* yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan. PHP disebut sebagai pemrograman *Server Side Programming*, hal ini dikarenakan seluruh prosesnya dijalankan pada *server*. PHP adalah suatu bahasa dengan hak cipta terbuka atau yang juga dikenal dengan istilah *Open Source*, yaitu pengguna dapat mengembangkan kode-kode fungsi PHP sesuai dengan kebutuhannya.

## **II.6. Database MySQL**

MySQL atau dibaca "*My Sekuel*" dengan adalah suatu RDBMS (*Relational Database Management System*) yaitu aplikasi sistem yang menjalankan fungsi pengolahan data. MySQL pertama kali dikembangkan oleh MySQL AB yang kemudian diakuisisi Sun Microsystem dan terakhir dikelola oleh Oracle Corporation. (Sibero, 2013 : 97).

## **II.7. Adobe Dreamweaver**

*Adobe Dreamweaver* adalah suatu produk *Web Developer* yang dikembangkan oleh *Adobe Systems Inc.*, sebelumnya produk *Dreamweaver* dikembangkan oleh *Macromedia Inc.* yang kemudian sampai dengan saat ini pengembangannya diteruskan oleh *Adobe Systems Inc.* Setelah diambil alih oleh *Adobe Systems Inc.*, *Dreamweaver* dikembangkan dan dirilis dengan kode nama *Creative Suite (CS)*.

Ruang Kerja atau *Workspace* adalah bagian keseluruhan tampilan dari *Adobe Dreamweaver*. Ruang kerja *Dreamweaver* terdiri dari *Welcome Screen*, *Menu*, *Insert bar*, *Document Window*, *CSS Panel*, *Application Panel*, *Tag Inspector*, *Property Inspector*, *Result Panel* dan *Files Panel*. Masing-masing dari komponen tersebut memiliki fungsi dan aturan. (Sibero, 2013 : 384).

## **II.8. Unified Modeling Language (UML)**

Menurut Windu Gata (2013 : 4) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language (UML)*. UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

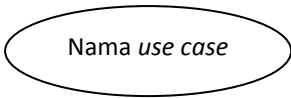
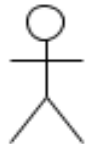

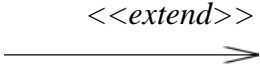
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

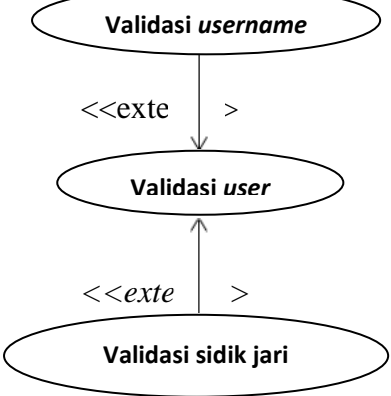

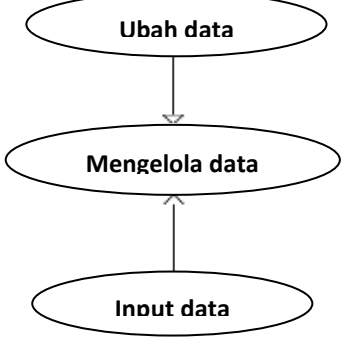

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

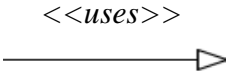
### 5. *Use case* Diagram

*Use case* diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut ini adalah simbol-simbol yang digunakan dalam *use case* diagram, yaitu :

**Tabel II.3. Simbol – Simbol *Use Case* Diagram**

<b>Simbol</b>	<b>Deskripsi</b>
<p><i>Use case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal diawal frase nama <i>use case</i></p>
<p>Aktor / <i>actor</i></p>  <p><b>nama aktor</b></p>	<p>orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari <i>actor</i> adalah gambar orang, tapi <i>aktor</i> belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal nama <i>aktor</i>.</p>
<p>Asosiasi/ <i>association</i></p> 	<p>Komunikasi antara <i>actor</i> dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor</p>
<p>Ekstensi / <i>extend</i></p> 	<p>Relasi <i>use case</i> tambahan kesebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu: mirip dengan prinsip <i>inheritance</i> pada pemograman berorientasi objek; biasanya <i>use</i></p>

	<p><i>case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal</p>  <pre> graph TD     A([Validasi username]) -- &lt;&lt;exte&gt;&gt; B([Validasi user])     C([Validasi sidik jari]) -- &lt;&lt;exte&gt;&gt; B   </pre> <p>Arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>
<p>Generalisasi/<i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:</p>  <pre> graph TD     A([Ubah data])     B([Mengelola data]) --&gt; A     C([Input data]) --&gt; B   </pre> <p>Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).</p>
<p>Menggunakan / <i>include</i> / <i>uses</i></p> <p>&lt;&lt;<i>include</i>&gt;&gt;</p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat <i>use case</i> ini.</p> <p>Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i> :</p> <p><i>include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan</p>



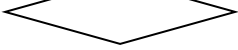

	<p>dijalankan.</p> <p><i>uses</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan.</p>
---	--


Sumber : (Rosa A.S,dkk ; 2013 : 155-158)

#### 6. Diagram Aktivitas (*Activity Diagram*)

*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Berikut ini adalah simbol-simbol yang digunakan dalam *activity diagram*, yaitu :

**Tabel II.4. Simbol - Simbol *Activity Diagram***

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

<p>Status akhir</p> 	<p>Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.</p>
<p>Swimlane</p> <div style="border: 1px solid black; padding: 2px; width: fit-content;"> <p>nama swimlane</p> </div>	<p>Memisahkan organisasi bisnis yang bertanggungjawab terhadap aktivitas yang terjadi.</p>

Sumber : (Rosa A.S, dkk ; 2013 : 161-163)

## 7. Diagram Kelas (*Class Diagram*)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

*Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

Tabel II.5. *Multiplicity Class Diagram*

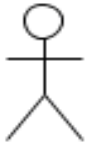

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4




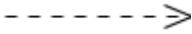
(Sumber : Windu Gata ; 2013 : 9)

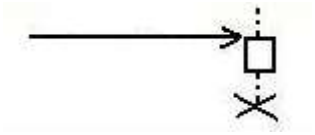
8. Diagram Urutan (*Sequence Diagram*)

*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

Tabel II.6. Simbol *Sequence Diagram*

Simbol	Deskripsi
<p>Aktor</p>  <p>nama aktor</p>	<p>orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari <i>actor</i> adalah gambar orang, tapi <i>actor</i> belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal nama <i>actor</i>.</p>
<p>Garis hidup / <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek.</p>

<p>Objek</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <u>nama objek : nama kelas</u> </div>	<p>Menyatakan objek yang berinteraksi pesan.</p>
<p>Waktu aktif</p> <div style="border: 1px solid black; width: 20px; height: 40px; margin: 10px auto;"></div>	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah tahapan yang dilakukan didalamnya.</p>
<p>Pesan tipe create</p> <div style="text-align: center; margin: 10px auto;"> <p>&lt;&lt;create&gt;&gt;</p>  </div>	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>
<p>Pesan tipe call</p> <div style="text-align: center; margin: 10px auto;"> <p>1 : nama_metode{ }</p>  </div>	<p>Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri.</p>
<p>Pesan tipe send</p> <div style="text-align: center; margin: 10px auto;"> <p>1 : masukan</p>  </div>	<p>Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>
<p>Pesan tipe return</p> <div style="text-align: center; margin: 10px auto;"> <p>1 : keluaran</p>  </div>	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>

<p>Pesan tipe destroy</p> <p><code>&lt;&lt;destroy&gt;&gt;</code></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada <i>destroy</i>.</p>
---	--

Sumber : (Rosa A.S, dkk ; 2013 : 165-167)

