BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Secara leksikal, sistem berarti : susunan yang teratur dari pandangan, teori, asas dan sebagainya. Dengan kata lain, sistem adalah suatu kesatuan usaha yang terdiri dari bagian – bagian yang berkaitan satu sama lain yang berusaha mencapai suatu tujuan dalam suatu lingkungan kompleks. (Prof.Dr.Ir.Marimin, M.Sc ; 2008: 1).

II.1.1. Pakar

Pakar adalah seseorang yang memilki pengetahuan khusus, pemahaman, pengalaman, dan metode - metode yang digunakan untuk memecahkan persoalan dalam bidang tertentu (Rika Rosnelly ; 2011 : 10)

II.1.2. Sistem Pakar

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 ; 2011 : 10)

II.1.3. Manfaat Sistem Pakar

Sistem pakar menjadi sangat populer karena sangat banyak kemampuan dan manfaat yang diberikannya, di antaranya :

1. Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat daripada manusia.
2. Membuat seorang yang awam bekerja seperti layaknya seorang pakar.
3. Meningkatkan kualitas, dengan memberi nasehat yang konsisten dan mengurangi kesalahan.
4. Mampu menangkap pengetahuan dan kepakaran seseorang.
5. Dapat beroperasi di lingkungan yang berbahaya.
6. Memudahkan akses pengetahuan seorang pakar.
7. Andal, sistem pakar tidak pernah menjadi bosan dan kelelahan atau sakit.
8. Bisa digunakan sebagai meddia pelengkap dalam pelatihan. Pengguna pemula yang bekerja dengan sistem pakar akan menjadi lebih berpengalaman karena adanya fasilitas penjelas yang berfungsi sebagai guru.
9. Meningkatkan kemampuan untuk menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar.

   (T.Sutojo, Dkk ; 2011 : 160)

II.1.4. Kekurangan Sistem Pakar

Selain manfaat, ada juga beberapa kekurangan yang ada pada sistem pakar, diantaranya :

1. Biaya yang sangat mahal untuk membuat dan memeliharanya.
2. Sulit dikembangkan karena keterbatasan keahlian dan ketersediaan pakar.
3. Sistem pakar tidak 100% bernilai benar. (T.Sutojo, Dkk ;2011:16)

II.1.5. Ciri – Ciri Sistem Pakar

Ciri – ciri sistem pakar adalah sebagai berikut :

1. Terbatas pada domain keahlian tertentu.
2. Dapat memberikan penalaran untuk data – data yang tidak lengkap atau tidak pasti.
3. Dapat menjelaswkan alas an – alas an dengan cara yang dapat dipahami.
4. Bekerja berdasarkan kaidah / *rule* tertentu.
5. Mudah dimodifikasi.
6. Basis pengetahuan dan mekanisme inferensi terpisah.
7. Keluarannya berupa anjuran.
8. Sistem dapat mengaktifkan kaidah secara searah yang sesuai, dituntun oleh dialog dengan pengguna. (T.Sutojo, DKK ; 2011 : 162)

II.1.6. Area Permasalahan Aplikasi Sistem Pakar

Biasanya Aplikasi Sistem Pakar menyentuh beberapa area permasalahan beikut :

1. Interprestasi : Menghasilkan deskripsi situasi berdasarkan data – data masukan.
2. Prediksi : Memperkirakan akibat yang mungkin terjadi dari situasi yang ada.
3. Diagnosis : Menyimpulkan suatu keadaan berdasarkan gejala – gejala yang diberikan (*symptoms).*
4. Desain : Melakukan perancangan berdasarkan kendala – kendala yang diberikan.

5. Planning :Merencakana tindakan – tindakan yang akan dilakukan.
6. Monitoring : Membandingkan hasil pengamatan dengan proses perencanaan.
7. Debugging : Menentukan penyelesaian dari suatu kesalhan sistem.
9. Reparasi : Melaksanakan rencana perbaikan.
10. *Instruction* : Melakukan intruksi untuk diagnosi, debugging dan perbaikan kinerja.
11. Kontrol : Melakukan control terhadap hasil interprestasi, diagnosis, debugging, monitoring dan perebaikan tingkah laku sistem. (T.Sutojo, DKK ; 2011 : 162 - 163)

II.1.7. Konsep Dasar Sistem Pakar

Konsep dasar sistem pakar meliputi enam hal berikut ini :

1. Kepakaran (*Expertise)*

   Kepakaran merupakan suatu pengetahuan yang diperoleh dari pelatihan, membaca, dan pengalaman. Kepakran inilah yang memungkinkan para ahli dapat mengambil keputusan lebih cepta dan lebih baik daripada seseorang yang bukan pakar. Kepakaran itu sendiri meliputi pengetahuan tentang :

   a. Fakta – fakta tentang bidang permasalahan tertentu.
   b. Teori – teori tentang bidang permaslahan tertentu.
   c. Aturan – aturan dan prosedur – prosedur – prosedur menurut bidang permaslahan umumnya.

d. Aturan *heuristic* yang harus dikerjakan dalam suatu situasi tertentu.
e. Strategi global untuk memecahkan permasalahan.

2. Pengetahuan tentang (meta *knowledge)* Pakar (*Expert)*

   Pakar adalah seseorang yang mempunyai pengetahuan, pengalaman, dan metode khusus, serta mampu menerapkannya untuk memecahkan masalah atau memberi nasehat. Seorang pakar harus mampu menjelaskan dan mempelajari hal – hal baru yang berkaitan dengan topik permasalahan, jika perlu harus mampu menyusun kembali pengetahuan – pengetahuan yang didapatkan, dan dapat memecahkan aturan – aturan serta menentukan relevansi kepakarannya. Jadi seorang pakar harus mampu melakukan kegiatan – kegiatan berikut :

   a. Mengenali dan memformulasikan permasalahan.
   b. Memecahkan permasalahan secara cepat dan tepat.
   c. Menerangkan pemecahannya.
   d. Belajar dari pengalaman.
   e. Merestrukturisasi pengetahuan.
   f. Memecahkan aturan – aturan.
   g. Menentukan relevansi.

3. Pemindahan kepakran (Transfering Expertise)

   Tujuan dari Sistem Pakar adalah memudahkan kepakaran dari seorang pakar ke dalam komputer, kemudian di transfer kepada orang lain yang bukan pakar. Proses kegiatan ini melibatkan empat kegiatan, yaitu :

a. Akuisisi pengetahuan (dari pakar atau sumber lain)
b. Representasi pengetahuan (pada computer)
c. Inferensi pengetahuan.
d. Pemindahan pengetahuan ke pengguna.

4. Inferensi (*Inferencing)*

   Inferensi adalah sebuah prosedur (program) yang mempunyai kemampuan dalan melakukan penalaran . Inferensi ditampilkan pada suatu komponen yang disebut mesin inferensi yang mencakup prosedur – prosedur mengenai pemecahan masalah. Semua pengetahuan yang dimiliki oleh seorang pakar disimpan pada basis pengetahuan oleh sistem pakar . Tugsa mesin inferensi adalah mengambil kesimpulan berdasarkan basi pengetahuan yang dimilikinya.

5. Aturan – aturan (*Rule)*

   Kebanyakan *software* sistem pakar komersial adalah sistem yang berbasis *rule (rule – based system)* , yaitu pengetahuan disimpan terutama dalam bentuk *rule*, sebagai prosedur – prosedur pemecahan masalah.

6. Kemampuan menjelaskan (*Explanation Capability)*

   Fasilitas lain dari Sistem Pakar adalah kemampuannya untuk menjelaskan saran atau rekomendasi yang diberikannya. Penjelasan dilakukan dalam subsistem yang disebut subsistem penjelasan (*explanation).* Bagian dari sistem ini memungkinakan sistem untuk memeriksa penalaran yang dibuatnya sendiri dan menjelaskan operasi – operasinya.

Karakteristik dan kemampuan yang dimiliki oleh sistem pakar berbeda dengan sistem konvensional. Perbedaannya di tunjukan pada berikut :

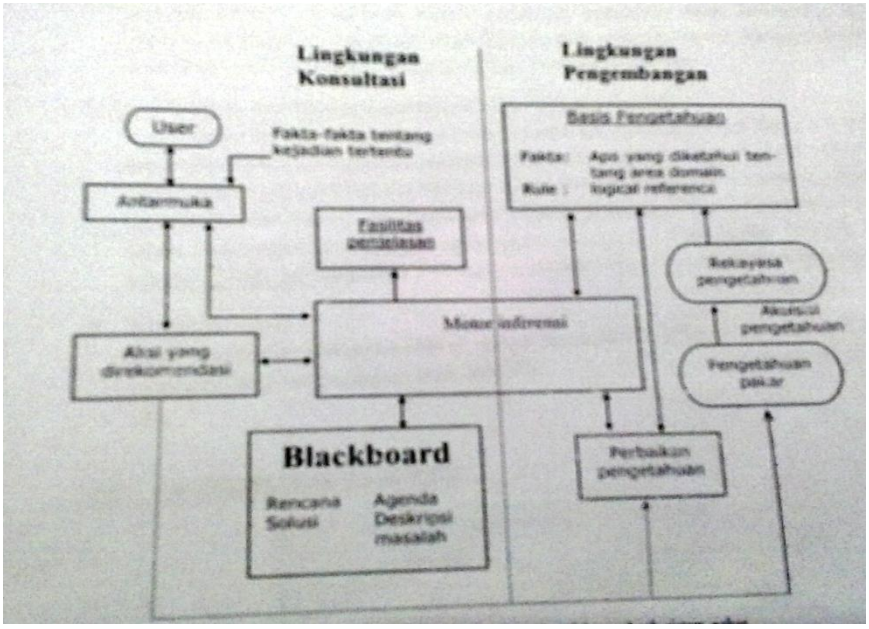
Sistem Konvensional	Sistem Pakar
Informasi dan pemrosesanya biasanya digabungkan dalam suatu program.	Basis pengetahuan dipisahkan secara jelas dengan mekanisme inferensi.
Program tidak membuat kesalahan (yang membuat kesalahan : program atau pengguna).	Program dapat melakukan kesalahan.
Biasanya tidak menjelaskan mengapa data masukan diperlukan atau bagaimana *output* dihasilkan.	Penjelasan merupakan bagian terpenting dari semua sistem pakar.
Perubahan program sangant menyulitkan.	Perubahan dalam aturan – aturan mudah untuk dilakukan.
Sistem hanya bisa beroperasi setelah lengkap atau selesai	Sistem dapat beroperasi hanya dengan aturan – aturan yang sedkit (sebagai prototype awal).
Eksekusi dilakukan langkah demi langkah (algoritmik).	Eksekusi dilakukan dengan menggunakan hueristik da logika apada seluruh basis pengetahuan.
Perlu ada informasi lengkap agar bisa beroperasu.	Dapat beroperasi dengan ingormasi yang tidak lengkap atau mengandung ketidakpastian.
Manipulasi efektif dari basis data yang besar	Manipulasi efektif dari basis pengetahuan yang besar.
Menggunakan data.	Menggunakan pengetahuan.
Tujuan utama : efesisensi	Tujuan utama : efektivitas
Mudah berurusan dengan data kuantitatif	Mudah berurusan dengan data kualitatif
Menangkap, menambah dan mendistribusikan akses data numeric atau informasi.	Menangkap, menambah dab mendistribusikan akses ke pertimbangan dan pengetahuan.

(T.Sutojo,Dkk ; 2011 : 163 - 166)

II.1.8. Struktur Sistem Pakar

Ada dua bagian penting dari sistem pakar yaitu lingkungan pengembangan (*development environment)* dan lingkungan konsultasi (*consultation environment).* Lingkungan pengembangan digunakan oleh pembuat sistem pakar untuk membangun komponen – komponennya dan memperkenalkan

pengetahuan ke dalam *knowledge base* (basis pengetahuan). Lingkungan konsultasi digunakan oleh pengguna untuk berkonsultasi sehingga pengguna mendapatkan pengetahuan dan nasihat dari sistem pakar layaknya berkonsultasi dengan seorang pakar. Gambar 2.1 menentukan komponen – komponen yang penting dalam sebuah sistem pakar.



Gambar II.1 : Komponen – komponen yang penting dalam sebuah sistem pakar

(Sumber : T.Sutojo,Dkk ; 2011 : 167)

Keterangan :

1. Akuisisi Pengetahuan

   Subsistem ini digunakan untuk memasukkan pengetahuan dari seorang pakara dengan cara merekayasa pengetahuan agar bisa diproses oleh komputer dan menaruhnya ke dalam basis pengetahuan dengan format tertentu (dalam bentuk representasi pengetahuan) . Sumber – sumber

pengethauan bisa diperoleh dari pakar, buku, dokumen multimedia, basis data, laporan riset khusus dan informasi yang terdapat di web.

2. Basis Pengetahuan (*Knowledge Base)*

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

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

3. Mesin Inferensi (*Inference Engine).*

   Mesin Inferensi adalah sebuah progeam yang berfungsi untuk memandu proses penalaran terhadap suatu kondisi berdasarkan pada basis pengetahuan yang ada, manipulasi dan mengarahkan kaidah, model dan fakta yang disimpan dalam basis pengetahuan untuk mencapai solusi atau kesimpulan. Dalam prosesnya, mesin inferensi menggunakan strategi pengendalian, yaitu strategi yang berfungsi sebagai panduan arah dalam melakukan proses penalran . Ada tiga pengendalian yang digunakan yaitu *forward chaining, backward chaining* dan gabungan dari kedua teknik tersebut.

4. Daerah Kerja (*Blackboard)*

   Untuk merekam hasil sementara yang akan dijadikan sebagai keputusan dan untuk menjelaskan sebuah masalah yang sedang terjadi . Sistem pakar membutuhkan *Blackboadr* yaitu area pada memori

yang berfungsi sebagai basis data. Tiga tipe keputusan yang dapat direkam pada *blackboard* , yaitu :

a. Rencana : Bagaimana menghadapi masalah
b. Agenda : Aksi – aksi potensial yang sedang menunggu untuk di eksekusi.
c. Solusi : Calon aksi yang dibangkitkan.

5. Antarmuka Pengguna (*User Interface)*

   Digunakan sebagai media komunikasi anatar pengguna dan sistem pakar . Komunikas ini paling bagus bila disajikan dalam bahasa alami (*natural language)* dan dilengkapi dengan grafik, menu, dan formulir eletktronik. Pada bagian ini akan terjadi dialog anatara sistem pakar dan pengguna.

6. Subsistem Penjelasan (*Explanation Subsystem / Justifier)*

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

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

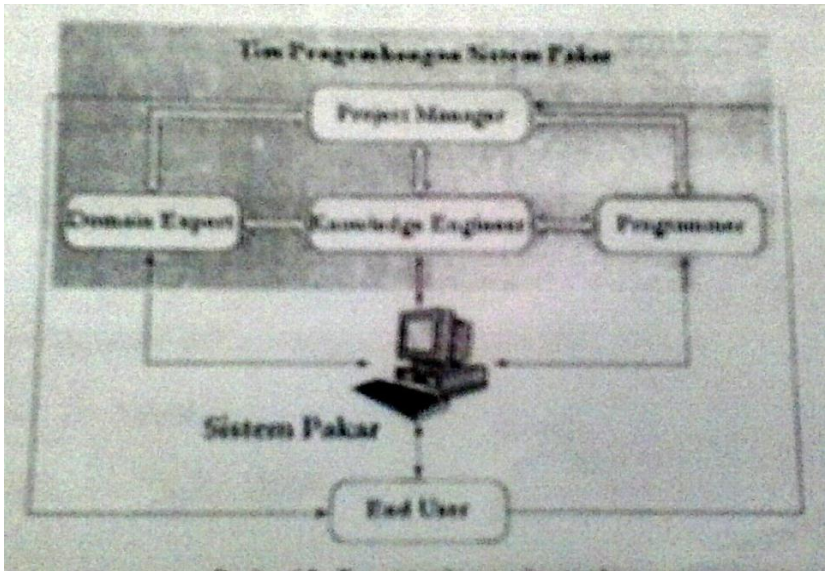
   Kemampuan memperbaiki pengetahuan (*Knowledge Refining System)* dari seorang pakar diperlukan untuk menganalisis pengetahuan, belajar dari kesalahan masa lalu, kemudian memperbaiki pengetahuanmya sehingga dapat dipakai pada masa mendatang. Kemampuan evaluasi diri seperti itu diperlukan oleh program agar dapat menganalisi alas an

– alas an kesuksean dan kegagalannya dalam mengambil kesimpullan. Dengan cara ini basis pengethauan yang lebih baik dan penalaran yang lebih efektif akan dihasilkan.

8. Pengguna (*User*)

   Pada umunya pengguna sistem pakar bukanlah seorang pakar (*non-expert)* yang membutuhkan solusi, saran, atau pelatihan (*training)* dari berbagai permasalahan yang ada. (T.Sutojo,Dkk ; 2011 : 166 - 169)

II.1.9. Tim Pengembangan Sistem Pakar



Gambar II.2 : Tim pengembang sistem pakar

(Sumber : T.Sutojo,Dkk ; 2011 : 169)

Domain *expert* adalah pengetahuan dan kemampuan seorang pakar untuk menyelesaikan masalah terbatas pada keahliannya saja. Misalnya seorang pakar penyakit jantung, ia hanya mampu mengangani masalah – masalah yang berkaitan dengan penyakit jantung saja. Ia tidak bisa menyelesaikan masalah – masalah

ekonomi, poltik , hukum dan lain – lain. Keahlian inilah yang dimasukkan dalam sistem pakar.

Knowledge Enginer (Perekayasa Pengetahuan) adalah orang yang mampu mendesain, membangun, dan menguji sebuah sistem pakar.

Programmer adalah orang yang mebuat program sistem pakar, mengode domain pengetahuan agar dapat dimengerti oleh computer.

Project Manager adalah pemimpin dalam tim pengembangan sistem pakar.

End – User (Biasanya sisebut *user* saja) adalah orang yang menggunakan sistem pakar. (T.Sutojo, DKK ; 2011 : 169 - 170)

II.1.10. *Rule* Sebagai Teknik Representasi Pengetahuan

Setiap *rule* terdiri dari dua bagian , yaitu bagian **IF** disebut evidence (fakta – fakta) dan bagian **THEN** disebut Hipotesis atau kesimpulan.

Syntax *Rule* adalah : IF E THEN H

E : *Evidence* (fakta – fakta) yang ada

H : Hipotesis atau Kesimpulan yang dihasilkan

Secara umum, *rule* mempunyai *evidence* lebih dari satu yang dihubungkan oleh kta penghubung AND atau OR, atau kombinasi keduanya. Tetapu sebaiknya biasakan menghindari penggunaan AND dan OR secara sekligus dalam satu *rule.*

IF (E1 AND E2 AND E3 …. AND EN) THEN H

IF (E1 OR E2 OR E3…. OR EN) THEN H

Satu *evidence* bisa juga mempunyai hipotesis lebih dari satu

IF E THEN (H1 AND H2 AND H3 …. AND HN)

(T.Sutojo,Dkk ; 2011 : 170 - 171)

II.2. Teorema Bayes

Teorema Bayes merupakan satu dari cabang teori statistik matematik yang memungkinkan kita untuk membuat suatu model ketidakpastian dari suatu kejadian yang terjasi dengan menggabungkan pengetahuan umum dengan fakta dari hasil pengamatan. Probabilitas bayes merupakan salah satu cara untuk mengatasi ketidakpastian data dengan menggunakan formula bayes yang di nyatakan :

$$P(H \mid E) = \frac{P(E \mid H) \cdot P(H)}{P(E)}$$

Dimana **:**

P(H |E) : probabilitas hipotesis H jika diberi Evidence E

P(E |H) : probabilitas munculnya evidence E jika diketahui hipotesis H

P(H) : probabilitas hipotesis H tanpa memandang evidence apapun

P(E) : probabilitas evidence E (Wisnu, DKK , 2011)

Contoh Kasus :

Berikut ini contoh perhitungan manual penyakit Perikondritis jika diketahui 2 gejala sebagai berikut :

1. Kerusakan pada Kartaligo : 0,7
2. Cedera pada telinga : 0,6

Jika probabilitas gejala-gejala dengan memperhatikan penyakit yang terjadi adalah :

1. Kerusakan pada Kartaligo : 0,5
2. Cedera pada telinga : 0,5

Perhitungan nilai bayes

P (Perikondritis | Kerusakan pada kartilago)

$$= \frac{P(\text{Kerusakan pada kartilago} \mid \text{Perikondritis}) * P(\text{Perikondritis})}{P(\text{Keruskana pada kartilago} \mid \text{Perinkondritis}) + P(\text{Cedera pada telinga} \mid \text{Perikondritis})}$$

P (Perikondritis | Kerusakan pada kartilago)

$$= \frac{(0,5) * (0,7)}{(0,5) + (0,5)}$$

$$= \frac{0,35}{1} = 0,35$$

P (Perikondritis | Cedera pada Telinga)

$$= \frac{P(\text{Cedera pada telinga} \mid \text{Perikondritis}) * P(\text{Perikondritis})}{P(\text{Keruskana pada kartilago} \mid \text{Perinkondritis}) + P(\text{Cedera pada telinga} \mid \text{Perikondritis})}$$

P (Perikondritis | Cedera pada telinga)

$$= \frac{(0,5) * (0,6)}{(0,5) + (0,5)}$$

$$= \frac{0,3}{1} = 0,3$$

Perhitungan total bayes

Total Bayes = Bayes1 + Bayes2

= 0,35 + 0,3

= 0,65

Jadi, Total Bayes penyakit Perikondritis adalah 0,65

(Jurnal Informatika Vol.2. No.2, Juli 2008)

II.3. Penyakit Itik Ratu

Itik termasuk ternak yang mempunyai daya tahan tubuh yang tinggi terhadap penyakit namun secara umum, penyakit itik timbul sebagai reaksi skunder akibat adanya faktor utama yang tidak berfungsi dengan baik. Faktor tersebut adalah sebagai berikut :

1. Sanitasi yang tidak baik

   Sanitasi meliputi semua aspek yang berhubungan langsung dangan itik. Sanitasi yang harus diperhatikan antara lain sanitassi tempat pakan dan minum, lantai kandang, serta sanitasi lingkungan kandang sekitar.

2. Biosekuriti yang kurang ketat

   Biosekuriti dalam peternakan merupakan faktor yang harus dilaksanakan. Biosekuriti dalam beternak itik dia antaranya sebagai berikut :

   a. Hindari masuknya hewan lain seperti tikus atau kucing kedalam kandang karena hewan tersebut dapat membawa bibit penyakit dari luar dan dapat menyebabkan itik stress.
   b. Petugas kandang memakai alas kaki yang mrncegah kontaj kaki dengan tanah.
   c. Petugas kandang memakai pakaian khusus saat berada di dalam kandang. Pakaian dari luar tidak boleh di pakai kedalam kandang.

3. Manajemen yang salah

   Manajemen yang dimaksud disini adalah semua pengaturan yang berhubungan dengan pemelihara itik, diantaranta sebagai berikut :

a. Manajemen pemberian pakan yang salah seperti kadar proteinnya rendah, jumlah pakan untuk perekornya kurang, komposisi pakan yang tidak sesuai dengan kebutuhan itik, kualita pakan yang jelek seperti menggumpal , bau apek dan berjamur. Hal – hal tersebut menyebabkan pertumbuhan itik menjadi terhambat dan itik menjadi lemah sehingga bibit penyakit mudah masuk.
b. Manajemen lantai kandang. Kandang yang lembap ditambah dengan kotoran yang menumpuk akan menyebabkan gas amoniak tinggi. Hal ini dapat menyebabkan itik kekurangan pksigen dab gangguan pernapasan sehingga efek sekundernya adalah bibit penyakit akan mudah masuk . Untuk itu, perlu dilakukan program pembersihan kotoran itik atau pengapuran lantai kandang secara teratur.
c. Manajemen Ventilasi. Petukaran udara dalam kandang harus baik sehingga bau amoniak dapat diminimalkan. Semakin besar itik, oksigen yang di butuhkan semakin besar.
d. Pengaruh faktor lingkungan, terutama cuaca dan suhu . Pengaruh suhu yang berubah – ubah secara ekstrim akan menyebabkan itik menjadi stress sehimgga akan mudah terserang penyakit. Selain itu faktor lingkungan seperti pergantian tenaga kandang, perubahan warna seragam kandang di tengah periode produksi dan banyaknya orang yang masuk kedalam kandang menyebabkan itik stress.

   (Ir. Supriyadi MM ; 2010 : 69 – 71)

II.3.1. Jenis Jenis Penyakit Itik

Penyakit yang menyerang itik dapat disebabkan oleh bakteri, virus, jamur dan parasit, serta penyakit akibat lainya.

1. *Salmonellosis*

   Penyebabnya adalah bakteri *Salmonella typhimurium* dan *Salmonella enteriditis*.

   Gejala :

   a. Kotoran itik encer dan bewarna hijaun keputihan.
   b. Itik Sulit bernapas
   c. Itik tampak kehausan.
   d. Warna bulu terlihat kusam.
   e. Sayap terkulai.

2. *Botulismus*

   Penyebabnya adalah bakteri *Clostridium botulnium*. Bakteri tersebut dapat berasal dari pakan sisa yang membusuk, bankai itik, atau bangkai tikus.

   Gejala :

   a. Itik tidak banyak bergerak,diam dan murung
   b. Itik mengalami kelumpuham pada leher, sayap dan kaki.
   c. Pupil matanya melebar.
   d. Kadang-kadang mengalami rontok bulu

e. Kotoran itik encer dan bewarna hijau keputihan.

3. *Avian Cholera*

Penyebabnya adalah *Pasteurela Avicida*. Faktor lainnya adalah itik mengalami kedinginan berlebih dan sanitasi kandang jelek.

Gejala

a. Itik mengalami sessak napas dan memperlihatkan gerakan yang tidak terkontrol.
b. Pialnya membengkak dan berubah warna dari merah cerah menjadi merah padam.
c. Pada mata keluar air mata, dari lubang hidung keluar eksudat (semacam lendir) seperti gelatin.
d. Badannya lemas dan panas.

4. *Coccidiosis*

Penyebabnya adalah bakteri *Coccidia sp*.

Gejala

a. Kotoran cair dan bewarna merah karena bercampur darah
b. Nafsu makan turun
c. Kondsi tubuh lemah

5. *Corryza*

   Penyebabnya adalah bakteri *Haemophilus Gallinarum*. Dapat dipicu juga dengan adanya perubahan cuaca dari musim kemarau ke musim hujan atau sebaliknya, dari musim hujan ke kemarau.

   Gejala

   a. Muka dan Mata itik bengkak
   b. Keluar lendir kental dari lubang hidung itik.
   c. Itik kelihatan mengantuk.
   d. Nafsu makan menurun.

6. Ngorok

   Penyebabnya adalah *Mycoplasma Gallisepticum* atau *Mycoplasma Anatis*. Amoniak yang tinggi di atas 25 ppm dan debu kandang yang pekat dapat menghambat sekresi lendir permukaan saluran pernapasan dan mengundang infeksi mikoplasma. Penularan terjadi melalui kontak langsung dengan itik yang terserang, peralatan kandang, telur tetas yang terinfeksi atau melalui petugas kandang.

   Gejala

   a. Napasnya berbunyi atau ngorok
   b. Batuk batuk, terutama pada malam hari
   c. Sering menggeleng-gelengkan kepalanya
   d. Keluar cairan terus dari hidung

7. *Avian Influenza*

   Penyebabnya adalah *virus H5N1*.

   Gejala :

   a. Pada periode *starter* : nafsu makan turun dalam 1 hari, tubuh lemas dan mati mendadak dalam jumlah yang banyak.
   b. Pada periode *grower* : nafsu makan turun dan pertumbuhan terhambat, belum menimbulkan kematian.
   c. Pada periode *layer* : nafsu makan turun dan produksi telur turun, belum menimbulkan kematian.

8. *Fowl Pox*

   Penyebabnya dalah *virus pox*. Penularan penyakit dapat terjadi melaui udara atau gigitan nyamuk.

   Gejala

   a. Timbul bintil-bintil pada kaki, sekitar mata dan paruh
   b. Sulit bernafass
   c. Mata bengkak dan berisi pus yang telah mengkeju.

(Ir. Supriyadi MM ; 2010 : 71 – 78)

II.4. UML *(Unified Modelling Language)*

Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk ***Unified Modelling Language*** **(UML)** . UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun

perangkat lunak. UML merupakan metedologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saaat ini sangat banyak dipergunakan dalam dunia industry yang merupakan standar baqhsa pemodelan umum dalam industry perangkat lunak dan pengembangan sistem.

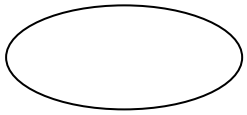
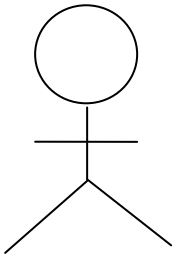
Alat bantu yang dapat digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut :

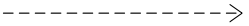
a. *Use Case Diagram*

Use Case Diagram merupakan pemodelan untuk kelakuan (*behavio*r) sistem informasi yang akan dibuat. *Use Case* mendiskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Dapat dikatakan *Use Case* di gunakan unntuk mengetahui fungsi apa saja yang ada digunakan fungsi – fungsi tersebut .

Simbol – simbol yang digunakn dalam *Use Case Diagram* yaitu :

Tabel II.1 Simbol – simbol dalam Use Case Diagram

Gambar	Keterangan
	*Use case* Mengambarkan fungsionalitas yang disediakan sistem sebagai unit – unit yang bertukar pesan antar unit dengan actor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama use case
	*Actor* atau Aktor adalah abstraction dari orang atau sistem yang lain untuk mengaktifkan fungsi dari target sistem . Untuk mengindentifikasikan *actor*, harus ditentukan pembagian tenaga kerja dan tugas – tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa *actor* beriteraksi dengan *use case* , tetapu tidak memiliki kontrol terhadap *use case*.
	Asosiasi antara *actor* dan *usecase* , digambarkan

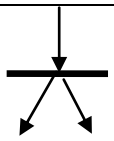
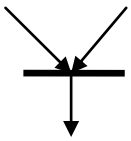
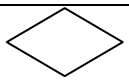
	dengan garis tanpa tanda panah yang mengindikasikan siapa atau apa yanh meminta interaksi secara langsung.
	Asosiasi antara *actor* dan *use case* yang menggunakan panah terbuka untuk mengindikasikan bila actor berinteraksi secara pasif dengan sistem.
 <<include>>	*Include* merupakan didalam *use case* lain (*required*) atau pemanggilan use case oleh use case lain, contohnya adalah pemanggilan sebuah funsi program.
<<extend>>	*Extend* merupakan perluasan dari use case lain jika kondisi atau syarat terpenuhi

(Sumber : Windu Gata, Grace Gata ; 2010 ; 4 – 6)

b. *Activity Diagram*

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis. Simbol – simbol yang digunakan dalam *Activity Diagram,* yaitu :

Tabel II.2. A*ctivity Diagram*

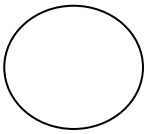
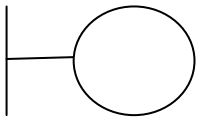
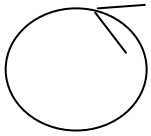
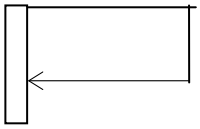
Gambar	Keterangan
	*Start Point*, diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	*End Point*, akhir aktifitass
	*Activities*, menggambarkan suatu proses / kegiatan bisnis.
	*Fork* ; digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan parallel menjadi satu
	*Join* (Penggabungan) digunakan untuk menunjukkan adanya deomposisi
	*Decision Points*, menggambarkan pilihan untuk pengambilan keputusan , tru atau false

(Sumber : Windu Gata, Grace Gata ; 2010 : 6 -7)

c. *Sequence Diagram*

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendskripsikan waktu hidup objek dan pesan yang telah dikirmkan dan diterima antar objek. Simbol – simbol yang digunakan dalam *Squence Diagram* , Yaitu :

Tabel II.3. Simbol – simbol dalam *Sequence Diagram*

Gambar	Keterangan
	*Entity Class*, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas – entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun bassis data.
	*Boundary Class*, berisi kumpulan kelas yang menjadi interface atau interaksi antara satu atau lebih actor dengan sistem, seperti tampilan formentary dan form cetak
	*Control Class* , suatu objek yang berisi logika aplikasi yang tidak memilki tanggung jawab kepada entitias, Contohnya adalah kalkulasi dan aturan bibsnis yang melibatkan berbagai objek.
	*Message,* simbol mengirim pesan antar class
	*Recursive*, menggambarkan pengiriman pesan yang dikirimkan untuk dirinya sendiri
	*Activation* mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan aktivaai sebuah program
	*Lifeline* , garis titik – titik yang terhubung dengan objek, sepanjang lifeline terdapat activation.

(Sumber : Windu Gata, Grace Gata ; 2010 7 - 8)

d. *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 jawan entitas yang menetukan 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, *Assocations*, *Generaliation* dan *Aggregration*, Atribut (*Attributes*), Operasi (*Operations/Method*), dan *Visibility*, tingkat akses objek eksternal kepada operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau kardinaliti .

Tabel II.4. *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, Grace Gata ; 2010 : 8 – 9)

II.5. PHP

PHP (*PHP Hypertext Prepocessor*) adalah bahasa *scripting server-side* bagi pemograman web.Secara sederhana, PHP merupakan *tool* bagi

pengembangan web dinamis. PHP sangat popular karena memiliki fungsi *built-in* lengkap, cepat, mudah dipelajari dan bersifat gratis (Angga Wibowo ; 2007 : 2)

II.6. MySql

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. Tidak seperti Apache yang merupakan software yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia yaitu MySQL AB. MySQL AB memegang penuh hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

Beberapa kelebihan MySQL antara lain :

a. Free (bebas didownload)
b. Stabil dan tangguh
c. Fleksibel dengan berbagai pemrograman
d. Security yang baik
e. Kemudahan *management database.*
f. Perkembangan software yang cukup cepat. (Achmad Solichin ; 2010 : 85)

II.7. Kamus Data

Kamus data (*Data Dictionary)* mencakup definisi – definisi dari data yang disimpan di dalam basis data dan dikendalikan oleh sistem manajemen basis data.Struktur basis data memuat dalam kamus data adalah kumpulan dari seluruh definisi filed, definisi table, relasi table dan hal – hal lainya.(Raymcloed Mcloed, Jr, DKK ; 2008 ; 171)

Kamus data juga mempunyai suatu bentuk untuk memperasingkat arti/makna dari simbol yang dijelaskan, yang disebut notasi. Notasi ini lebih mudah dimengerti daripada narasi. Notasi atau simbol yang digunakan dibagi menjadi dua macam, yaitu sebagai berikut :

1. Notasi Tipe Data

   Notasi ini digunakan untuk membuat spesifikasi format input maupun output suatu data. Notasi yang umum digunakan antara lain adalah :

Notasi	Keterangan
X	Setiap karakter
A	Karakter alphabet
Z	Angka dari nol ditampilakan sebagai spasi kosong
.	Titik, sebagai pemisah ribuan
,	Koma, sebagai pemisah pecahan
-	Hypen, sebagai tanda penghubung (contoh : 021-7500282)
/	Slash sebagai tanda pembagi (contoh : 24/10/1969)

2. Notasi Struktur Data

   Notasi ini digunakan untuk membuat spesifikasi elemen data di mana notasi yang umum digunakan adalah sebagai berikut :

Notasi	Keterangan
=	Terdiri dari
+	And (dan)
0	Pilihan (boleh ya atau tidak)

{}	Iterasi / pengulangan proses
[]	Pilih salah satu pilihan
I	Pemisah pilihan didalam tanda []
*	Keterangan atau catatan
@	Petunjuk (key field)

(Yusmaida Satia ; 2012 : 23 – 24)

II.8. Normalisasi

Normalisasi merupakan cara pendekatan dalam membangun desain logika basis data relasional yang tidak secara nlangsung berkaitan dengan model data., tetapi dengan menerapkan sejumlah aturan dan kriteria standar untuk menghasilkan struktur tabel yang normal. Pada dasarnya desain logika basis data relasionak dapatvmenggunakan prinsip normalisasi maupun transformasi dari model E-R ke betuk fisik. (Kusrini ; 2007 : 40)

II.8.1 Bentuk – Bentuk Normalisasi

a. Bentuk tidak normal

   Bentuk ini merupakan umpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan apa adanya sesuai keadaannya

b. Bentuk Normal tahap pertama

   Definisi :

   Sebuah tabel disebut 1NF jika

   1. Tidak ada baris yang duplikasi dalam tabel tersebut
   2. Masing – masing cell bernilai tunggal

Catatan : Permintaan yang menyatakan tidak ada baris yang duplikasi dalm sebuah tabel berarti tabel tersebut memiliki sebuah kunci, meskipun kunci tersebut dibuat dari kombinasi lebih dari satu kolom atau bahkan kunci tersebut merupakan kombinasi dari semua kolom.

Berikut in akan dicontohkan normalisasi dari tabel Kuliah yang memilik atribut : Kode_kul, Nama_kil, sks, semester, waktu, tempat, dan nama_dosen.

Tabel Kuliah tersebut tidak memenuhi normalisasi pertama karena terdapat atribut waktu yang tergolong ke dalam atribut bernilai banyak. Agar tabel tersebut dapat memenuhi 1NF, maka solusisnya adalah dengan mendekomposisi tabwl Kuliah menjadi :

a. Tabel Kuliah (kode_kul, nama_kul, sks, semester, nama_dos)
b. Tabel Jadwal (kode_kul, waktu, ruang)

c. Bentuk Normal tahap kedua

Bentuk Normal Kedua (2NF) terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam primary key memiliki ketergantungan fungsional pada *primary key* secara utuh.

Sebuah tabel dikatakan tidak memenuhi 2NF, jika ketergantungannya hanya bersifat persial (hanya tergantung pada sebagian dari *primary key).*

Bentuk normal kedua akan dicontohkan berikut :

Misal tabel Nilai terdiri dari atribut kode_kul, nim dan nilai. Jika pada tabel Nilai, misalnya kita tambahkan sebuah atribut yang bernilai redundan , yaitu : nama_mhs, maka tabel Nilai ini di anggap menlnggar 2NF.

Primary Key pada tabel Nilai adalah (kode_kul, nim)

Penambahan atribut baru (nama_mhs) akan menyebabkan adanya ketergantungan fungsional yang baru yaitu nim->nama_mhs. Karena atribut nama_mhs ini hanya memliki ketergantungan parsial pada *Primary Key* secara utuh (hanya tergantung pada nim, padahal nim hanya bagian dari *primary key)* . Bentuk kedua ini dianggap belum memadai karena meninjau sifat ketergantungan atribut terhadapa *primary key* saja.

d. Bentuk normal tahap ketiga

Sebuah tabel dikatakan memenuhi bentuk normal ketiga (3NF), jika untuk setiap ketergantungan fungsional dengan notasi X -> A, dimana A mewakili semua atribut tunggal di dalam tabel di dalam tabel yang tidak ada i dalam X, maka :

1. X harus lah superkey pada tabel tersebut
2. Atau A merupakan bagian dari *primary key* pada tabel tersebut.

Misalkan pada tabel Mahasiswa , atribut alamat_mhs dipecah kedalam alamat_jalan, alamat_kota dan kode_pos. Bentuk ini tidak memenuhi 3NF, karena terdapat ketergantungan fungsional baru yang muncul pada tabel yaitu:

alamat_jalan nama_jelas -> kode_ pos

Dalam hal ini (alamat_jalan, nama_kota) bukan superkey sementara kode_pos juga bukan bagian dari primary key pada mahasiswa.

Jika tabel Mahasiswa didekomposisi menjadi tabel Mahasiswa dan tabel Alamat, maka telah memenuhi 3NF. Hal itu dapat dibuktukan dengan

memeriksa dua ketergantungan fungsional pada tabel alamat tersebut, yaitu :

alamat_jalan, nama_kota -> kode_pos

kode_pos -> nama_kota

Ketergantungan fungsional yang pertama tidak melanggar 3NF, karena (alamat_jalan, nama_kota) merupakan superkey (sekaligus sebagai primary key) dari tabel Alamat tersebut. Demikian juga dengan ketergantungan fungsional yang kedua meskipun (kode_pos) bukan merupakan superkey, tetapi nama_kota merupakan bagian dari primary key dari tabel Alamat. Karena telah memenehi 3NF, maja tabel tersebut tidak perlu di-dekomposisi lagi.

e. Bentuk Normal tahap keempat daan kelima

Penerapan aturan normalisasi sampai bentuk normal ketiga sudah memeadai untuk menghasilkan tabel berkualitas baik. Namun demikian, terdapat pula bentuk normal keempat (4NF) dan kelima (5NF). Bentuk normal keempat berkaitan dengan sifat ketergantungan banyak nilai (Multivalued Depency) pada suatu tabel yang merupakan pengembangan dari ketergantungan fungsional. Adapun bentuk normal tahap kelima merupakan nama lain dari *Project Join Normal Form* (PJNF)

f. *Boyce Code Normal Form* (BCNF)

- Memenuhi 1NF
- Relasi harus bergantung fungsi pada atribut *superkey.*

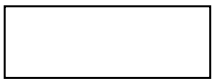
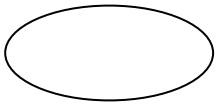
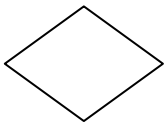
(Kusrini ;2007 : 41-43)

II.9. ERD (*Entity Relationship Diagrams)*

ERD merupakan salah satu alat (*tool)* berbentuk grafis yang populer untuk desain database. *Tool* ini relatif lebih mudah dibandingkan dengan normalisasi. Kebanyakan sistem analisi memakai alat ini, tetapi yang jadi masalah, kalau kita cermati secra seksama , tool ini mencapai 2NF. ERD sendiri dibagi dua bagian , antara lain desain awal (*Prelimiary Design)* dan desain akhir (*Final Design).*

II.9.1. Bentuk Simbol Grafis ERD

Bentuk grafis dari ERD sebagai berikut :

1. Entitas 
2. Attribut/ *Field* 
3. *Link* (Hubungan)
4. Himpunan Relasi / *Interface* 

(Yuniar Supardi ; 2010: 448)