BAB II

TINJAUAN PUSTAKA

II.1. Sistem Pakar

Sistem pakar merupakan cabang dari *Artifical Intelligence* (AI) yang cukup tua karena sistem ini mulai dikembangkan ada pertengahan 1960. Sistem pakar yang muncul pertama kali adalah *General-purpose problem solver* (GPS) yang dikembangkan oleh Newel dan Simon. Sampai saat ini sudah banyak sistem pakar yang dibuat, seperti MYCIN untuk diagnosa penyakit, DENDRAL untuk mengidentifikasi struktur molekul campuran yang tak dikenal, XCON & XSEL untuk membantu konfigurasi sistem komputer besar, SOPHIE untuk analisis sirkuit elektronik, Prospector digunakan di bidang geologi untuk membantu mencari dan menemukan deposit, FOLIO digunakan untuk membantu memberikan keputusan bagi seorang manager dalam stok dan investasi, DELTA dipakai untuk pemeliharaan lokomotif listrik diesel, dan sebagainya.

Istilah sistem pakar berasal dari istilah *knowledge-based expert system*. Istilah ini muncul karena untuk memecahkan masalah, sistem pakar menggunakan pengetahuan seorang pakar yang dimasukkan ke dalam komputer. Seseorang yang bukan pakar menggunakan sistem pakar untuk meningkatkan kemampuan pemecahan masalah, sedangkan seorang pakar menggunakan sistem pakar untuk *Knowledge assistant* (T. Sutojo, dkk; 2011: 165).

Dengan kata lain, sistem pakar adalah sistem komputer yang ditujukan untuk menirukan semua aspek (*emulates*) kemampuan pengambilan keputusan (*decision making*) seorang pakar. Sistem pakar memanfaatkan secara maksimal pengetahuan khusus selayaknnya seorang pakar untuk memecahkan masalah (Rika Rosnelly; 2012: 2).

II.1.1. Manfaat Sistem Pakar

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

1. Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat daripada manusia.
2. Membuat seorang yang awam bekerja 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. Meningkatkan kapabilitas sistem komputer. Integrasi sistem pakar dengan sistem komputer lainmembuat sistem lebih efektif dan mencakup lebih banyak aplikasi.

9. Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti. Berbeda dengan sistem komputer konvensional, sistem pakar dapat bekerja dengan informasi yang lengkap. Pengguna dapat merespon dengan: “tidak tahu” atau “tidak yakin” pada satu atau lebih pertanyaan selama konsultasi dan sistem pakar tetap akan memberikan jawabannya.
10. Bisa digunakan sebagai media pelengkap dalam pelatihan. Pengguna pemula yang bekerja sebagai sistem pakar akan menjadi lebih berpengalaman karena adanya fasilitas penjelas yang berfungsi sebagai guru.
11. Meningkatkan kemampuan untuk menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar.

II.1.2. Kekurangan Sistem Pakar

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

1. Biaya yang sangat mahal.
2. Sulit dikembangkan karena keterbatasan keahlian dan ketersedian pakar.
3. Sistem pakar tidak 100% bernilai benar.

II.1.3. Ciri – cirri Sistem Pakar

Ciri – ciri dari 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 menjelaskan alasan – alasan dengan cara yang dapat dipahami.
4. Bekerja dengan kaidah/rule tertentu.
5. Mudah dimodifikasi.
6. Basis pengetahuan dan mekanisme inferensi terpisah.
7. Keluarannya bersifat anjuran.
8. Sistem dapat mengaktifkan kaidah secara searah yang sesuai, dituntun oleh dialog dengan pengguna.

II.1.4. Area Permasalahan Aplikasi Sistem Pakar

Biasanya aplikasi sistem pakar menyentuh beberapa area permasalahan berikut :

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: merencanakan tindakan – tindakan yang akan dilakukan.

6. Monitoring: membandingkan hasil pengamatan dengan proses perencanaan.
7. Debugging: menentukan penyelesaian dari suatu kesalahan sistem.
8. Reparasi: melaksanakan rencana perbaikan.
9. *Instruction*: melakuakan instruksi untuk diagnosis, debugging, dan perbaikkan kinerja.
10. Kontrol: melakuakn kontrol terhadap hasil interprestasi, diagnosis, debugging, monitoring, dan perbaikan tingkah laku sistem.

II.1.5. Konsep Dasar Sistem Pakar

Konsep dasar sistem pakar meliputi enam hal berikut ini :

II.1.5.1. Kepakaran (*Expertise*)

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

1. Fakta – fakta tentang bidang permasalahan tertentu.
2. Teori – teori tentang bidang permasalahan tertentu.
3. Aturan – aturan dan prosedur – prosedur menurut bidang permasalahn umumnya.
4. Aturan heuristic yang harus dikerjakan dalam suatu situasi tertentu.
5. Strategi global untuk memecahkan permasalahan.

6. Pengetahuan tentang pengetahuan (*meta knowledge*)

II.1.5.2. Pakar (*Expert*)

Pakar adalah seorang yang mempunyai pengetahuan, pengalaman, dan metode khusus, serta mampu menerapkannya untuk memecahkan masalah dan 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 seseorang pakar harus mampu melakukan kegiatan – kegiatan berikut :

1. Mengenali dan memformulasikan permasalahan.
2. Memecahkan permasalahan secara cepat dan tepat.
3. Menerangkan pemecahannya.
4. Belajar dari pengalaman.
5. Merestrukturasi pengetahuan.
6. Memecahkan aturan – aturan.
7. Menentukan relevansi.

II.1.5.3. Pemindahan Kepakaran (*Transferring Expertise*)

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

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

II.1.5.4. Inferensi (*Inferencing*)

Inferensi adalah sebuah prosedur (program) yang mempunyai kemampuan dalam 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. Tugas mesin inferensi adalah mengambil kesimpulan berdasarkan basis pengetahuan yang dimilikinya.

II.1.5.5. Aturan – aturan (*Rule*)

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

II.1.5.6. 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 memungkinkan sistem untuk memeriksa penalaran yang dibuatnya sendiri dan menjelaskan operasi – opersinya.

Karakteristik ddan kemampuan yang dimiliki sistem pakar berbeda dengan sistem konvensional. Perbedaan ini ditunjukkan pada table II.1.

Tabel II.1 Perbandingan Antara Sistem Konvensional Dengan Sistem Pakar

Sistem Konvensional	Sistem Pakar
Informasi dan pemrosesannya biasanya digabungkan dalam suatu program.	Basis pengetahuan dipisahkan secara jelas dengan mekanisme inferensi.
Program tidak membuat kesalahan (yang membuat kesalahan: pemrogram atau pengguna).	Program dapat berbuat kesalahan.
Biasanya tidak menjelaskan mengapa data masukan diperlukan atau bagaimana output dihasilkan.	Penjelasan merupakan bagian terpenting dari semua sistem pakar.
Peruubahan program sangat menyulitkan.	Perubahan dalam aturan – aturan mudah untuk dilakukan.
Sistem hanya bias beroperasi setelah lengkap atau selesai.	Sistem dapat beroperasi hana dengan aturan – aturan yang sedikit (sebagai prototipe awal)
Eksekusi dilakukan langkah demi	Eksekusi dilakukan dengan

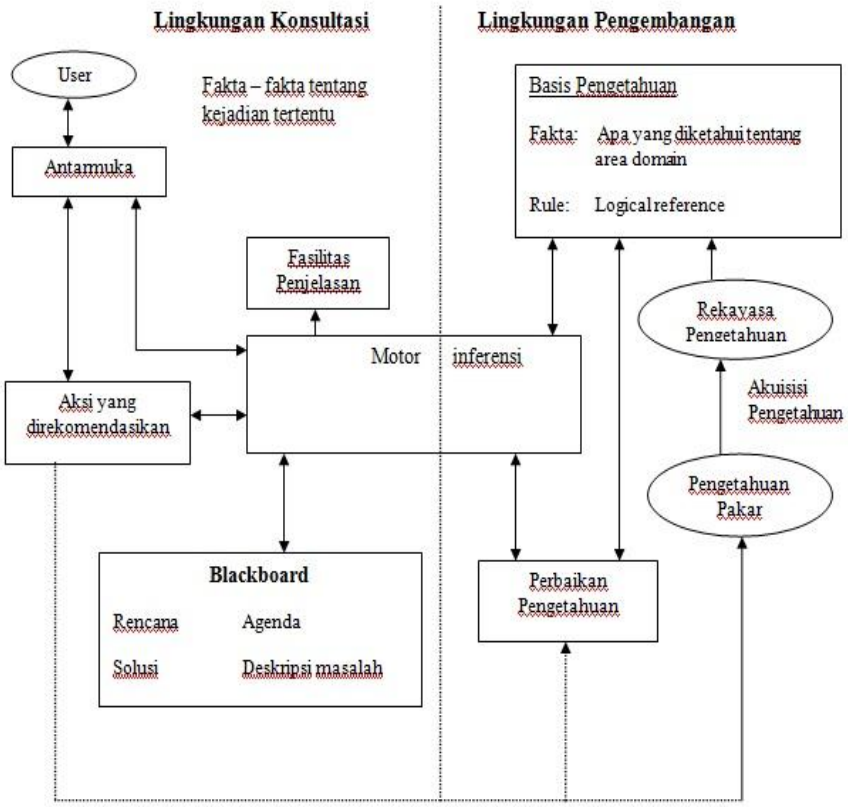
langkah (algoritmik).	menggunakan heuristik dan logika pada seluruh basis pengetahuan.
Sistem Konvensional	**Sistem Pakar**
Perlu informasi lengkap agar bisa beroperasi.	Dapat beroperasi dengan informasi yang tidak lengkap atau mengandung ketidakpastian.
Manipulasi efektif dari basis data yang besar.	Manipulasi efektif dari basis pengetahuan yang besar.
Menggunakan data.	Menggunakan pengetahuan.
Tujuan utama: efisiensi	Tujuan utama: efektivitas.
Mudah berurusan dengan data kuantitatif.	Mudah berurusan dengan data kualitatif.
Menangkap, menambah, dan mendistribusikan akses ke data numerik atau informasi.	Menangkap, menambah, dan mendistribusikan akses ke pertimbangan dan pengetahuan.

*(**Sumber :** T. Sutojo, dkk; 2011: 165 – 166)*

II.1.6. 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 kedalam *knowledge base* (basis pengetahuan). Lingkungan konsultasi digunakan oleh pengguna untuk berkonsultasi sehingga pengguna mendapatkan pengetahuan dan nasihat sistem pakar layaknya berkonsultasi

dengan seorang pakar. Gambar II.1 menunjukkan komponen – komponen yang penting dalam sebuah sistem pakar.



Gambar II.1 Komponen – komponen dalam sistem pakar
*(**Sumber :** T. Sutojo, dkk; 2011: 167)*

Keterangan :

1. Akuisisi Pengetahuan

   subsistem ini digunakan untuk memasukkan pengetahuan dari seorang pakar dengan cara merekayasa pengetahuan agar bisa diproses oleh komputer dan menaruhnya kedalam basis pengetahuan dengan format tertentu (dalam bentuk representasi pengetahuan).

2. Basis Pengetahuan (*Knowledge Base*)

   Basis pengetahuan mengandung pengetahuan yang diperlukan untuk memahami, memformulasikan, dan menyelesaikna masalah. Basis pengetahuan terdiri dari dua elemen dasar, yaitu fakta dan rule.

3. Mesin Inferensi (*Inference Engine*)

   Mesin inferensi adalah sebuah program yang berfungsi untuk memandu proses penalaran terhadap suatu kondisi berdasarkan pada basis pengetahuan yang ada, memanipulasi dan mengarahkan kaidah, model, dan fakta yang disimpan dalam basis pengetahuan untuk mencapai solusi atau kesimpulan.

4. Daerah Kerja (*Blackboard*)

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

5. Antarmuka Pengguna (*User Interface*)

   Digunakan sebagai media komunikasi antara pengguna dan sistem pakar. Komunikasi ini paling bagus bila disajikan dalam bahasa alami (*natural language*) dan dilengkapi dengan grafik, menu, dan formulir elektronik.

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

Berfungsi memberi penjelasan kepada pengguna, bagaimana suatu kesimpulan dapat diambil.

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 pengetahuannya sehingga dapat dipakai pada masa mendatang.

8. Pengguna (*User*)

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

II.1.7. Tim Pengembangan Sistem Pakar

1. *Domain Expert* adalah pengetahuan dan kemampuan seorang pakar untuk menyelesaikan masalah terbatas pada keahliannya saja. Misalnya seorang pakar penyakit jantung, ia hanya mampu menangani masalah – masalah yang berkaitan dengan penyakit jantung saja. Ia tidak bisa menyelesaikan masalah – masalah ekonomi, politik, hokum, dan lain – lain. Keahlian inilah yang dimasukan dalam sistem pakar.

2. *Knowledge Engineer* (Perekayasa Pengetahuan) adalah orang yang mampu mendesain, membangun, dan menguji sebuah sistem pakar.
3. *Programmer* adalah orang yang membuat program sistem pakar, mengode domain pengetahuan agar dapat dimengerti oleh komputer.
4. *Project manager* adalah pemimpin dalam tim pengembangan sistem pakar.
5. *End-User* (biasanya disebut *user* saja) adalah orang yang menggunakan sistem pakar.

II.2. Penyakit Autoimun Pada Kulit

Tubuh kita beraksi dalam bermacam-macam bentuk ketika terjadi kekacauan autoimun/penyakit autoimun yang menyebabkan sistem autoimun menyerang jaringannya sendiri. Suatu kekacauan autoimun /penyakit autoimun dapat menimbulkan kerusakan pada organ-organ tubuh,sendi dan otot, dan jaringan tubuh lainnya, salah satu jaringan yang umum diserang oleh kekacauan autoimun adalah kulit. Ada banyak tipe tipe penyakit yang menyerang kulit, diantaranya scleroderma, psoriasis, dermatomyositis, epidermolysis bullosa, dan bullous pemphigoid.

1. Scleroderma

Kulit hanyalan salah satu bagian tubuh yang diserang scleroderma, yang kondisinya bisa meluas merusak jaringan tubuh yang terkait. Jika penyakit ini

meluas ke seluruh tubuh, maka pasien akan mengalami tidak saja perubahan pada kulitnya, juga pada pembukluh darah, otot, dan organ-organ lainnya. Bentuk terbatas dari scleroderma adalah berupa bercak dari penebalan kulit, sedangkan scleroderma yang sifatnya sistematik bisa memengaruhi kehidupan si penderita.

Ada dua bentuk scleroderma sistematik, yaitu *progressive systemic sclerosis (PSS)* yang kadang hanya disebut *systemic sclerosis (SS)*, dan CREST *sysdrome*. Pasien dengan SS mengalami gejala yang mempengaruhi esophagus (saluran yang menghubungkan tekak dan lambung), intestine (saluran pencernaan antara lambung dan dubur), paru, jantung, dan ginjal. Sedangkan CREST syndrome dinamai karena gejalanya yaitu kalsinosis (akumulasi kalsium di bawah kulit), Raynaud phenomenon (warna kemerahan atau kebiruan pada jari-jari tangan dan kaki), esophageal dysfunction, sclerodactyly (jari jari menebal dan tegang pada kulit jari tangan dan kaki), dan telangiectasia (jerawat/bisul kemerahan yang disebabkan pembuluh darah yang melebar). Kemungkinan ada gejala tambahan berupa sakit sendi, sesak napas, napas berbunyi, konstipasi atau diare, pembengkakan, penurunan berat badan, rasa panas perut, atau rasa gatal dan panas di matai.

Pria maupun wanita beresiko mendapat scleroderma, namun kasus terbanyak pada wanita pada usia 30-an dan 40-an. Lingkungan kerja dengan banyak debu silica atau polivinil merupakan faktor resiko untuk penyakit autoimun jenis ini.

Menurut Yayasan Scleroderma di Amerika Serikat, sekitar 300.000 penduduknya menderita scleroderma; dan 33%-nya dengan tipe sistematik.

2. Psoriaris

Ini penyakit autoimun yang kronik dengan gejalanya berupa kemerahan pada kulit dan iritasi kulit. Ada lima tipe psoriasis yang berbeda, yaitu guttate, plaque, invese, erythrodemic, dan pustular. Yang paling umum adalah tipe plaque psoriaris. Gejala khasnya adalah bercak-bercak kemerahan pada kulit dilapisi oleh bercak-bercak warrna perak-keputihan dari kulit mati (dikenal sebagai scales atau sisik). Penelitian mutakhir mengindikasikan bahwa psoriaris agaknya dapat diturunkan-biasanya pasien penderita psoriarisis mempunyai keluarga/kerabat yang juga berpenyakit yang sama atau penyakit autoimun lainnya.

Gejala psoriasis dapat datang dan pergi selama hidup pasien. Datangnya gejala bisa disebabkan oleh serangan infeksi, luka pada kulit, terekspos matahari, medikasi, alcohol, bahkan stress. Mereka yang sistem imun-nya sudah “cacat” disebabkan misalnya HIV atau mengalami pengobatan kemoterapi, beresiko lebih besar mendapat serangan psoriaris.

The National Institute of Health di Amerika Serikat melaporkan sekitar 7,5 juta warga Amerika hidup bersama psoriaris. Umumnya munculnya gejala pada usia antara 15 – 35 tahun, meski orang pada semua usia bisa saja diserang penyakit autoimun ini. Sekitar 30% penderita psoriaris juga menderita arthritis – kondisi ini dikenal sebagai psoriatic arthritis.

3. Dermatomyositis

Penyakit autoimun ini aslinya menyerang otot, namun karena dermatomyositis juga menyerang kulit, maka sering kali dikategorikan sebaga penyakit autoimun yang menyerang kulit. Dermatomyositis "bergandengan tangan" dengan polymyositis (suatu penyakit autoimun yang menyebabkan otot lemah, sakit, dan kaku). Pasien dengan penyakit ini juga mengalami sulit menelan dan sesak napas. Kedua penyakit ini sama-sama mempunyai gejala serupa. Bedanya dermatomyositis mempunyai gejala tambahan berupa ruam kulit di bagian atas tubuh, juga penebalan dan ketegangan kulit pada beberapa bagian tubuh. Pasien dermatomyositis juga mempunyai kelopak mata berwana keunguan.

Dermatomyositis pada anak berbeda dengan pada orang dewasa dalam bentuknya, dengan gejala demam, fatique, ruam kulit, dan rasa lemah. Resiko terbesar pada anak muncul pada usia antara 5 – 15 tahun, dari pada orang dewasa pada usia 40 – 60 tahun. Lebih banyak menyerang wanita disbanding pria.

4. Epidermolysis Bullosa

Ada banyak bentuk dari epidermolysis bullosa, tapi hanya satu yakni epidermolysis bullosa acquisita yang aslinya adalah penyakit autoimun. Semua bentuk epidermolysis bullosa menyebabkan kulit melepuh berisi cairan yang bisa menjadi luka. Gosokan ringan pada kulit atau meningkatnya suhu ruangan bisa menimbulkan kulit melepu.

Mendiagnosis denga tepat bentuk epidermolysis bullosa tidaklah mudah. Namun ada suatu karakteristik dari epidermolysis bullosa acquisita yang secara normal tidak akan berkembang sampai saat kelah dalam hidup pasien – setelah usia 50 tahun sedangkan bentuk non-autoimun epidermolysis bullosa muncul saat kelahiran atau segera setelah itu. Meskipun begitu, diagnose tetap sulit, karena sulit membedakannya dari mucous membrane pemphigoid(penyakit autoimun yang gejalanya juga serupa, yaitu kulit melepuh).

5. Bullos Pemphigoid

Penyakit autoimun kronik ini mempunyai gejala kulit melepuh)bervariasi ringan sampai para). Dalam beberapa kasus pasien hanya menderita kemerahan ringan pada kulit atau iritasi kulit. Sementara pasienl lain lebih parah dengan gejala rentetan lepuhan (multiple blisters) yang bisa pecah dan membentuk borok. Pasien bullous pemphigoid normalnya mempunyai gejala lepuhan pada kulit lengan, kaki, atau pinggang dan sepertiga dari kasus lepuhan terbentuk di mulut. Beberapa pasien (tidak semua) dengan kondisi ini juga merasakan gatal dan mengalami perdarahan pada gusi.

Kasus bullous pemphigoid telah dilaporkan terjadi pada semua usia, namun umumnya pada usia lanjut. Pria dan wanita sama-sama beresiko. Sulit menetapkan penyakit ini karena gejalanya datang dan pergi, dan banyak pasien yang kondisinya berangsur membaik tanpa gejala setelah enam tahun.

II.3. Teorema Bayes

Metode yang diterapkan dalam pembuatan sistem pakar ini adalah metode pendekatan teorema bayes dengan cara kerja sebagai berikut :

Bentuk teorema bayes untuk evidence tunggal E dan hipotesis H adalah :

$$p(H|E,e) = \frac{p(E|H) \times p(H)}{p(E)}$$

Dengan :

p(H|E) = probabilitas hipotesis H terjadi jika *evidence* E terjadi

p(E|H) = probabilitas munculnya *evidence* E, jika hipotesis H terjadi

p(H) = probabilitas hipotesis H tanpa memandang *evidence* apapun.

P(E) = probabilitas *evidence* E tanpa memandang apapun.

H_i, H_2,…H_n adalah ;

$$p(H_i|E_1E_2 \dots E_3) = \frac{p(E_1|H_1) \times p(E_2|H_1) \times \dots \times p(E_m|H_i) \times p(H_i)}{\sum_{k=1}^{n} p(H_1|H_k) \times \dots \times p(E_m|H_k) \times p(H_k)}$$

(Sumber : Sistem Pakar Konsep dan Teori; Rika Rosnelly; 2012)

II.4. Pengertian Database

Menurut Yuhefizard; 2010:2. Banyak sekali definisi tentang database yang diberikan oleh para pakar dibidang ini. Database terdiri dari dua penggalan kata yaitu data dan base, yang artinya berbasiskan pada data. Tetapi secara konseptual, database diartikan sebuah koleksi atau kumpulan data yang saling berhubungan (relation), disusun menurut aturan tertentu secara logis, sehingga

menghasilkan informasi. Sebuah informasi yang berdiri sendiri tidaklah dikatakan database.

Contoh : Nomor telepon seorang pelanggan, disimpan dalam banyak tempat apakah itu di file pelanggan, di file alamat dan di lokasi yang lain. Antara file yang satu dengan file yang lainnya tidak saling berhubungan, sehingga apabila salah seorang pelanggan berganti nomor telepon dan anda hanya mengganti di file pelanggan saja, akibatnya akan terjadi ketidakcocokan data, karena di lokasi yang lain masih tersimpan data telepon yang lama.

Dalam sistem database hal ini tidak boleh dan tidak bisa terjadi, karena antara file yang satu dengan file yang lain saling berhubungan. Jika suatu data yang sama anda ubah, data tersebut di file yang lain akan otomatis berubah juga. Sehingga mampu menjadi informasi yang diinginkan dan dapat dilakukan proses pengambilan, penghapusan, pengeditan, terhadap data secara mudah dan cepat (Efektif, Efisien dan Akurat).

Data adalah fakta, baik berupa sebuah objek, orang dan lain – lain yang dapat dinyatakan dengan suatu nilai tertentu (angka, simbol, karakter tertentu, dan lain – lain). Sedangkan informasi adalah data yang telah diolah sehingga bernilai guna dan dapat dijadikan bahan dalam pengambilan keputusan.

Hubungan data dan informasi dapat digambarkan sebagai berikut :

Gambar II.2. Data dan Informasi.

(Sumber : Yuhefizard ; 2010 : 2)

II.5. Visual Basic

Menurut Edy Winarto; 2010:1. Visual Basic dibuat oleh microsoft, merupakan salah satu bahasa pemrograman berorientasi objek yang mudah dipelajari. Selain menawarkan kemudahan, Visual Basic juga cukup andal untuk digunakan dalam pembuatan berbagai aplikasi, terutama aplikasi database. Visual basic merupakan bahasa pemrograman event drive, dimana program akan menunggu sampai ada respons dari user/pemakai program aplikasi yang dapat berupa kejadian atau event, misalnya ketika user mengklik tombol atau menekan enter. Jika kita membuat aplikasi dengan visual basic maka kita akan mendapatkan file yang menyusun aplikasi tersebut, yaitu :

1. File Project (*.vbp)

   File ini merupakan kumpulan dari aplikasi yang kita buat. File project bisa berupa file *.frm, *.dsr atau file lainnya.

2. File Form (*.frm)

   File ini merupakan file yang berfungsi untuk menyimpan informasi tentang bentuk form maupun interface yang kita buat.

II.6. SQL Server

Menurut Cybertron Solution; 2010:101. SQL *Server* 2008 adalah sebuah RDBMS (*Relational Database Management System*) yang di *develop* oleh

Microsoft, yang digunakan untuk menyimpan dan mengolah data. Pada SQL *Server* 2008, kita bisa melakukan pengambilan dan modifikasi data yang ada dengan cepat dan efisien. Pada SQL *Server* 2008, kita bisa membuat *object – object* yang sering digunakan pada aplikasi bisnis, seperti membuat *database*, *table*, *fuction*, *stored procedure*, *trigger* dan *view*. Selain *object*, kita juga menjalankan perintah SQL (*Structured Query Language*) untuk mengambil data.

II.7. *Unified Modeling Language* (UML)

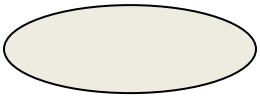
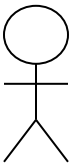
UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industry perangkat lunak dan pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut :

1. *Use Case* Diagram

Use case diagram merupakan pemodelan untuk kelakukan (*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

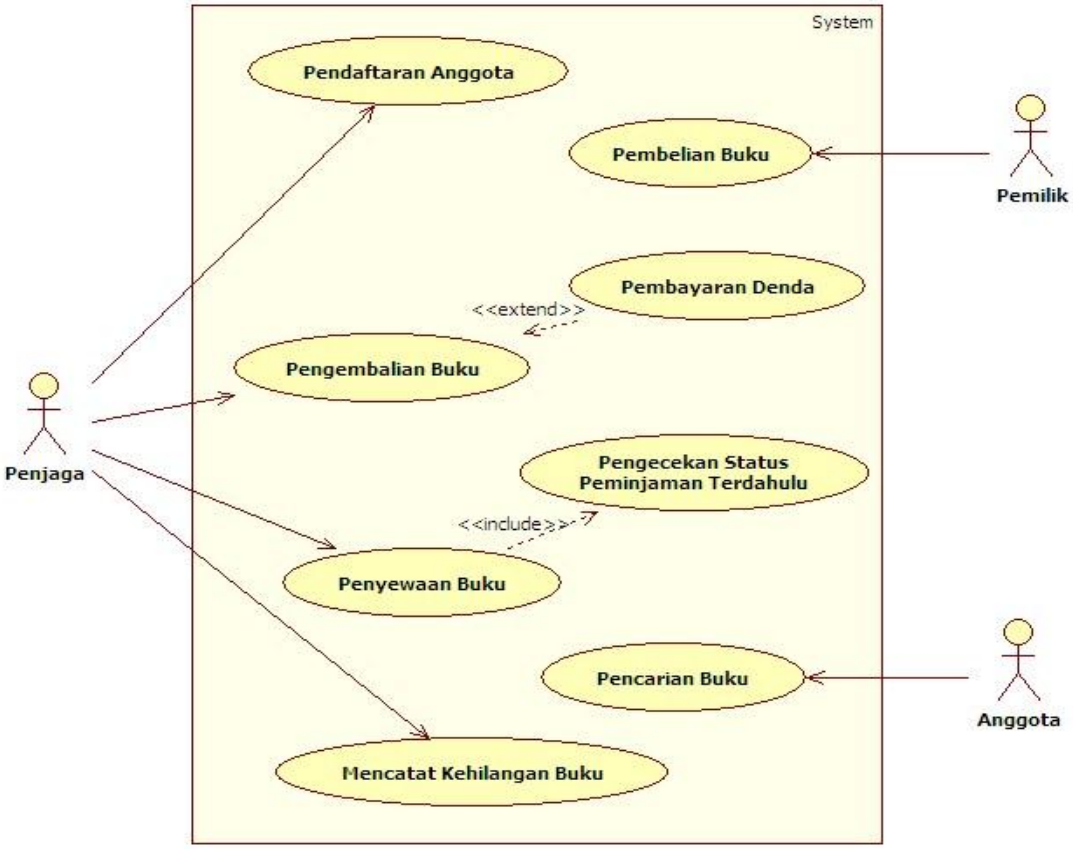
dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi – fungsi tersebut. (Windu Gata ; 2013 : 4).

Tabel II.2. Simbol – symbol yang digunakan dalam *Use Case* diagram

Gambar	Keterangan
	*Use case* menggambarkan 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 yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan aktor, 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 aktor beriteraksi dengan *use case*, tetapi tidak memiliki kontrol terhadap *use case*.
	Asosiasi antara aktor dan *use case*, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan *use case* yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
<<*include*>>	*Include*, merupakan di dalam *use case* lain (*required*) atau pemanggilan *use case* oleh *use case lain*,

	contohnya adalah pemanggilan sebuah fungsi program.
<<*extend*>>	*Extend*, merupakan perluasan dari *use case* lain jika kondisi atau syarat terpenuhi.

(***Sumber :*** Windu Gata ; 2013 : 4*)*

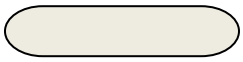
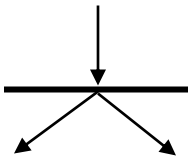
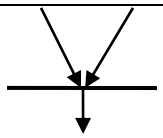
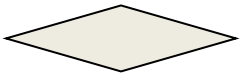
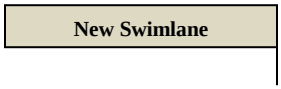


Gambar II.3 Contoh *Use Case Diagram*

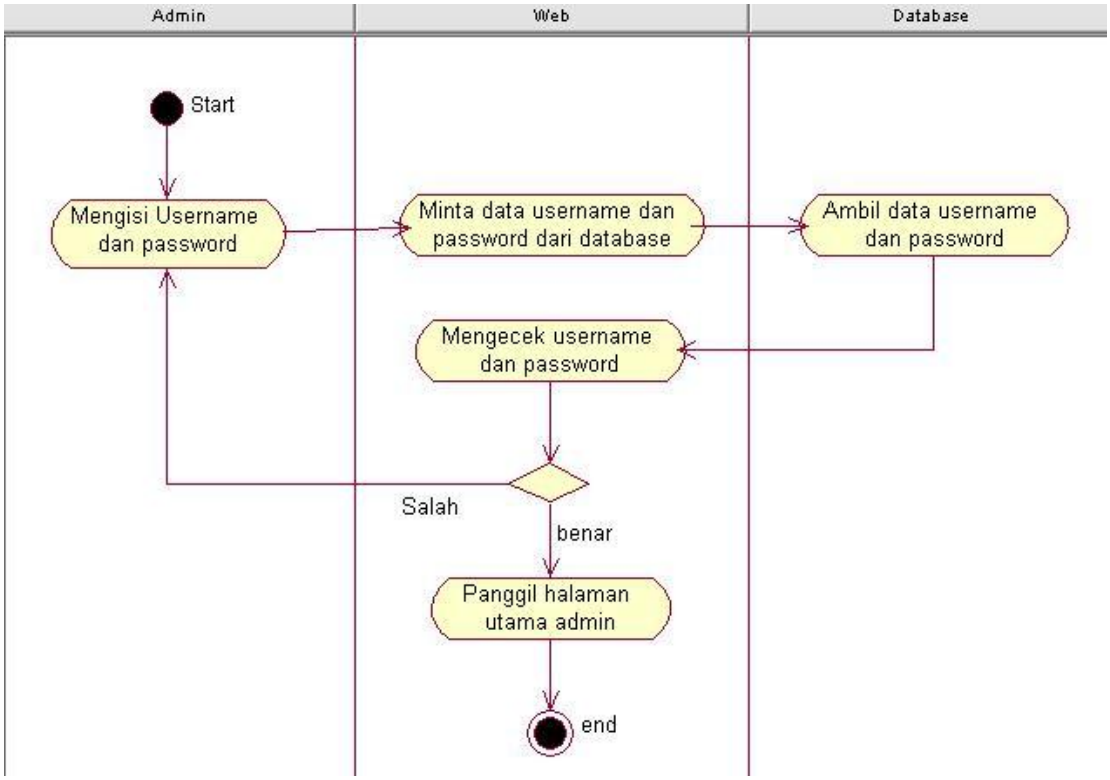
2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. (Windu Gata ; 2013 : 6).

Tabel II.3. Simbol – simbol yang digunakan dalam *Activity* diagramyaitu:

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

(***Sumber :*** Windu Gata ; 2013 : 6*)*

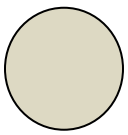


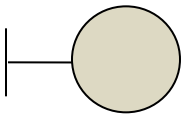
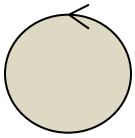
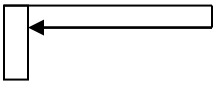
Gambar II.4 Contoh *Activity Diagram*

3. Diagram Urutan (*Sequence Diagram*)

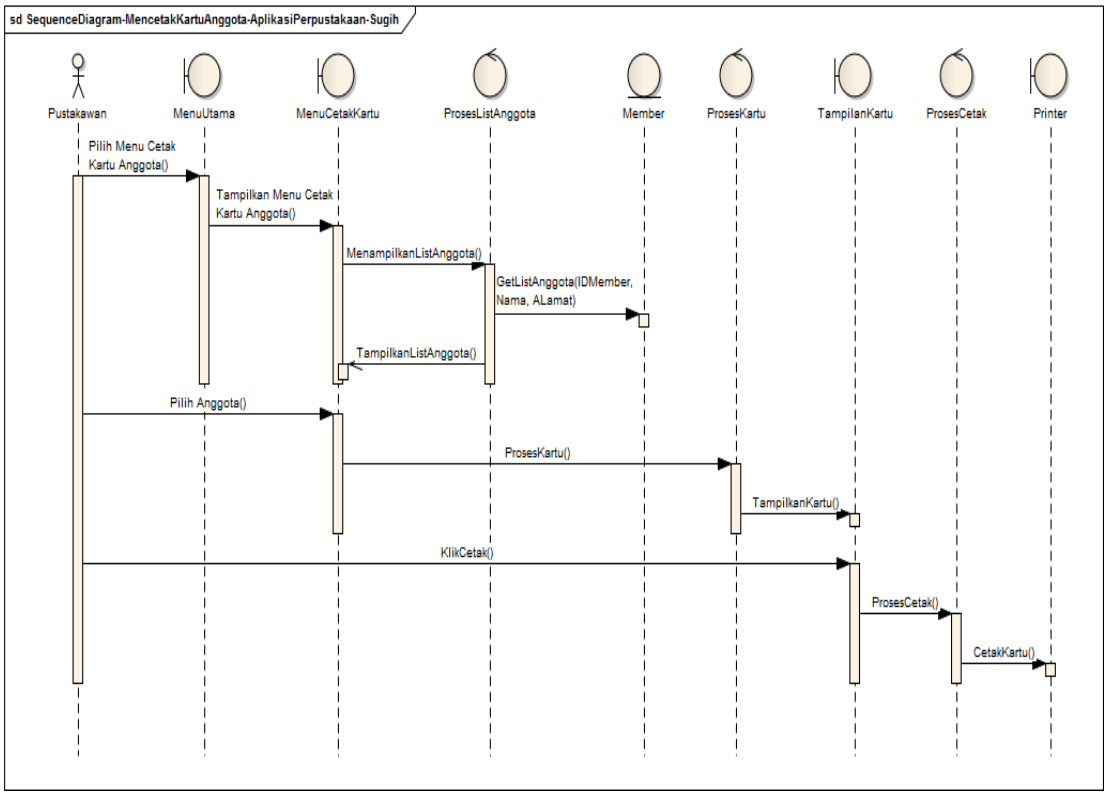
Sequence diagram menggambarkan kelakuan objek pada usecase dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. (Windu Gata ; 2013 : 7).

Tabel II.4. Simbol – simbol yang digunakan dalam *sequence* diagram

Gambar	Keterangan
	*Entity Class*, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas – entitas yang membentuk gambaran awal sistem yang menjadi landasan untuk

	menyusun basis data.
	*Boundary Class*, berisi kumpulan kelas yang menjadi interface atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan form cetak.
	*Control Class*, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek, *control objek* mengkoordinir pesan antara *boundary* dengan entitas
	*Message*, simbol mengirim pesan antara *class.*
	*Recursive*, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	*Activation*, *activation* mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.
	Lifeline, garis titik – titik yang terhubung dengan objek, sepanjang *lifeline* terdapat *activation*.

*(**Sumber :*** Windu Gata ; 2013 : 7*)*



Gambar II.5 Contoh *Sequence Diagram*

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggungjawab 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*,

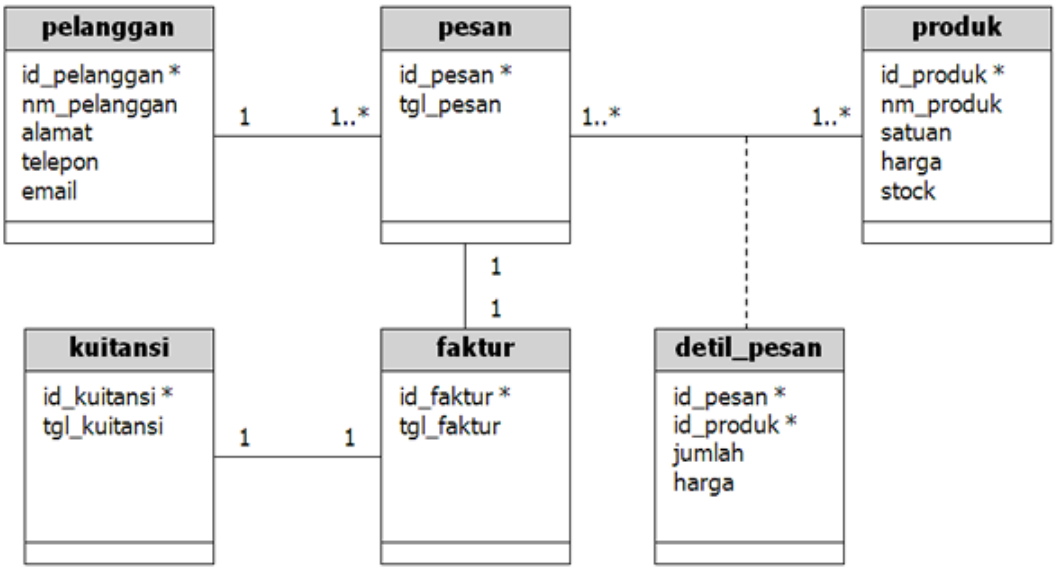
tingkat akses objek eksternal kepada suatu operasi atau atribut. (Windu Gata ; 2013 : 8).

Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau kardinaliti.

Tabel II.4. Hubungan antar kelas

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*)*



Gambar II.6 Contoh *Class Diagram*