BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Sistem adalah sekelompok unsur yang erat hubungannya satu dengan yang lain yang berfungsi bersama-sama untuk mencapai tujuan tertentu. Setiap sistem terdiri dari unsur-unsur, unsur-unsur tersebut merupakan bagian terpadu dari sistem yag bersangkutan. Unsur-unsur sistem berhubungan erat satu dengan yang lain dan sifat serta kerja sama antar unsur tersebut mempunyai bentuk tertentu (Tata Sutabri; 2005:8).

II.1.1. Karakteristik Sistem

Adapun karakteristik sistem yang dimaksud adalah :

1. Komponen Sistem *(Components)*

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama dan membentuk saru kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar, yang disebut supra sistem.

2. Batasan Sistem *(Boundary)*

   Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisahkan.

3. Lingkungan Luar Sistem *(Environtment)*

   Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut lingkungan luar sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi bagi sistem tersebut. Dengan demikian lingkungan luar tersebut harus tetap dijaga dan dipelihara. Lingkungan luar yang merugikan harus dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup sistem tersebut.

4. Penghubung Sistem *(Interface)*

   Media yang menghubungkan sistem dengan subsistem lain disebut penghubung sistem atau *interface.* Penghubung ini memungkinkan sumber-sumbr daya mengalir dari satu subsistem ke subsistem lain. Bentuk keluaran dari satu subsistem akan menjadi masukan untuk subsistem lain melalui penghubung tersebut. Dengan demikian dapat terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. Masukan Sistem *(Input)*

   Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yag dapat berupa pemeliharaan *(maintenance input)* dan sinyal *(signal input)*. Contoh, di dalam suatu unit sistem komputer, ”program” adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya dan ”data” adalah signal input untuk diolah menjadi informasi.

6. Keluaran Sistem *(Output)*

   Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain. Contoh, sistem informasi. Keluaran yang dihasilkan adalah informasi. Informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang menjadi input bagi subsistem lain.

7. Pengolah Sistem *(Proses)*

   Suatu sistem dapat mempuyai suatu proses yang akan mengubah masukan menjadi keluaran.

8. Sasaran Sistem *(Objective)*

   Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan (Tata Sutabri; 2005:11).

II.1.2. Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang, di antaranya :

1. Sistem Abstrak dan Sistem Fisik

   Sistem Abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, sedangkan sistem fisik merupakan sistem yang ada secara fisik.

2. Sistem Alamiah dan Sistem Buatan Manusia

   Sistem Alamiah adalah sistem yang terjadi melalu proses alam, tidak dibuat oleh manusia, sedangkan sistem buatan manusia merupakan sistem yang melibatkan interaksi manusia dengan mesin yang disebut *human machine system.*

3. Sistem Deterministik dan Sistem Probabilistik

   Sistem yang beroperasi dengan tingkah laku yang dapat diprediksi disebut sistem deterministik. Sedangkan sistem yang bersifat probabilistik adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilistik.

4. Sistem Terbuka dan Sistem Tertutup

   Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luarnya. Sistem ini bekerja seara otomatis tanpa campur tangan pihak luar. Sedangkan sistem terbuka adalah sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk subsistem lainnya (Tata Sutabri, 2005:13).

II.1.2. Daur Hidup Sistem

Menurut Tata Sutabri (2005:14), Siklus hidup sistem *(System Life Cycle)* adalah proses evolusioner yang diikuti dalam menerapkan sistem atau subsistem informasi berbasis komputer. Siklus hidup sistem terdiri dari serangkaian tugas yang erat mengikuti langkah-langkah pendekatan sistem karena tugas-tugas tersebut mengikuti pola yang teratur dan dilakukan secara *top down*. Siklus hidup sistem sering disebut sebagai pendekatan air terjun *(waterfall approach)* bagi pembangunan dan pengembangan sistem. Ada beberapa fase/tahapan dari daur hidup suatu sistem :

1. Mengenali Adanya Kebutuhan

   Kebutuha dapat terjadi sebagai hasil perkembangan dari organisasi dan volume yang meningkat melebihi kapasitas dari sistem yan ada. Semua kebutuhan harus didefenisikan dengan jelas tanpa kejelasan dari kebutuhan yang ada pembangunan sistem akan kehilangan arah dan efektifitasnya.

2. Pembangunan Sistem

   Suatu proses atau seperangkat prosedur yang harus diikuti untuk menganalisis kebutuhan yang timbul dan membangun suatu sistem untuk dapat memenuhi kebutuhan tersebut.

3. Pemasangan Sistem

   Setelah taham pembangunan selesai. Sistem kemudian akan dioperasikan. Pemasangan sistem merupakan tahap yang penting pula dalam daur hidup sistem. Peralihan dari tahap pembangunan menuju tahap

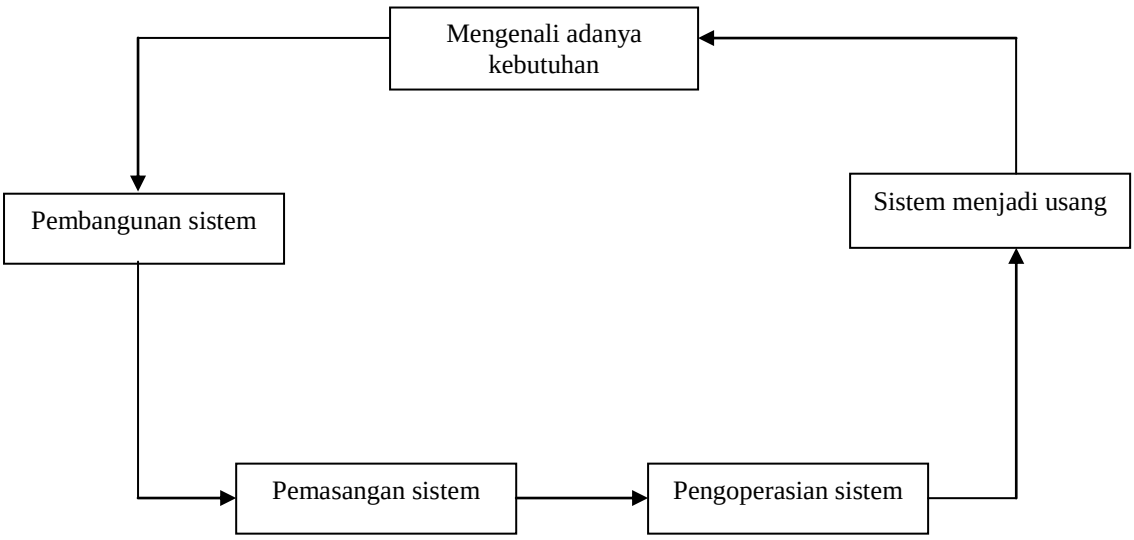
operasional terjadi pemasangan sistem yang sebenarnya, yang merupakan langkah akhir dari pembangunan sistem.

4. Pengoperasian Sistem

   Program-program komputer dan prosedur-prosedur pengoperasian yang membentuk suatu sistem informasi semuanya bersifat statis.

5. Sistem Menjadi Usang

   Kadang perubahan yang terjadi begitu drastis sehingga tidak dapat diatasi hanya dengan melakukan perbaikan-perbaikan pada sistem yang berjalan. Tibalah saatnya secara ekonomis dan teknis sistem yang ada sudah tidak layak lagi untuk dioperasikan dan sistem yang baru perlu dibangun untuk menggantikannya (Tata Sutabri; 2005:14).



Gambar II.1. Daur Hidup Sistem

(Sumber : Tata Sutabri; 2005:15)

II.2. Kepakaran *(Expertise)*

Menurut T.Sutojo dkk (2011:163), Kepakaran merupakan suatu pengetahuan yang diperoleh dari pelatihan, membaca dan pengalaman. Kepakaran inilah yang memungkinkan para ahli dapat mengambil keputusan lebih cepat dan lebih baik daripada seseorang yang bukan pakar.

Pakar *(Expert)* adalah seseorang yang mempunyai pengetahuan, pengalaman, dan metode khusus serta mampu menerapkannya untuk memecahkan masalah atau memberi nasihat. Seorang pakar harus mampu menjelaskan dan mempelajari hal-hal baru yang berkaitan denga topik permasalahan, jika perlu harus menyusun kembali pengetahuan-pengetahuan yang didapatkan, dan dapat memecahkan aturan-aturan serta menentukan relevansi kepakarannya (T.Sutojo; 2011:163).

II.3. Sistem Pakar

Menurut T.Sutojo (2011:160), Sistem pakar berasal dari istilah *knowledge-based expert system,* istilah ini muncul 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 menggunakana sistem pakar untuk *knowledge assistant.*

Tujuan dari sistem pakar adalah memindahkan kepakaran *(Transferring*

Expertise) seorang pakar ke dalam komputer, kemudian ditransfer kepada orang lain yang bukan pakar (T.Sutojo; 2011:160).

II.3.1. Manfaat Sistem Pakar

Sistem pakar sangat populer karena banyak kemampuan dan manfaat yang diberikannya antara lain:

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 (T.Sutojo; 2011:160).

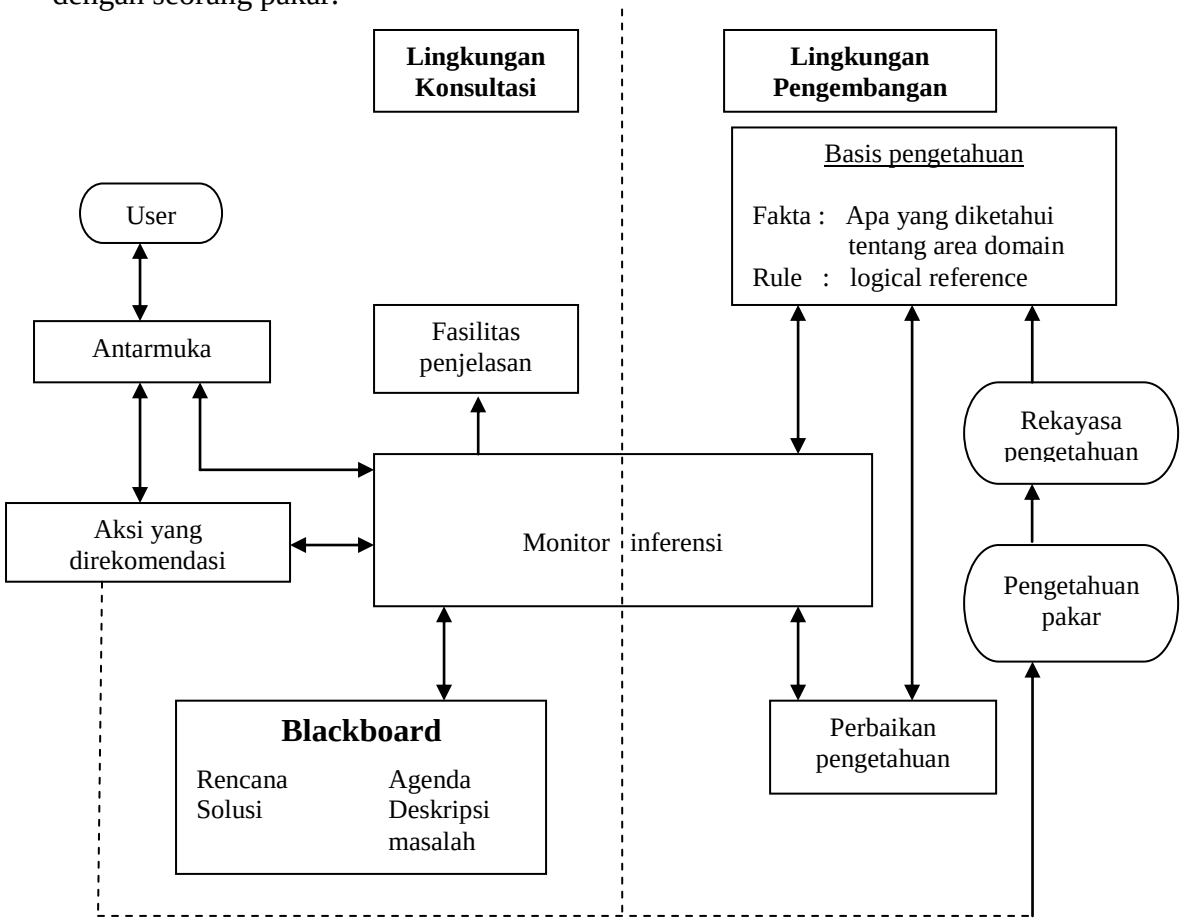
II.3.2. Kekurangan Sistem Pakar

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

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; 2011:161).

II.3.3. Struktur Sistem Pakar

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



Gambar II.2. Komponen-komponen yang penting dalam sebuah sistem pakar

(Sumber : T.Sutojo; 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 ke dalam basis pengetahuan dengan format tertentu (dalam bentuk representasi pengetahuan). Sumber-sumber pengetahuan bisa diperoleh dari pakar, buku, dokumen multimedia, basis data, laporan riset khusus, dan informasi yang terdapat di web.

2. Basis Pengetahuan *(Knowledge Base)*

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

   a. Fakta, misalnya situasi, kondisi dan permasalahan yang ada.
   b. *Rule* (Aturan), untuk mengarahkan pengguna pengetahuan dalam memecahkan masalah.

3. Mesin Inferensi *(Inference Engine)*

   Mesin Inferensi adalah sebuah program yang berfungsi untuk memandu proses penalaran terhadap suatu kondisi berdasarkan pada basis pengetahuan yang ada, 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. Tiga keputusan yang dapat durekam *Blackboard* yaitu :

   a. Rencana : bagaimana menghadapai masalah.
   b. Agenda : aksi-aksi potensial yang sedang menunggu dieksekusi.
   c. Solusi : calon aksi yang akan dibangkitkan.

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. Pada bagian ini akan terjadi dialog antara sistem pakar dan pengguna.

6. Subsistem penjelasan *(Explanation Subsytem / Justifier)*

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

7. Sistem perbaikan pengetahuan *(Knowledge Refining System)*

   Kemampuan memperbaiki pengetahuan dari seorang pakar diperlukan untuk menganalisis pengetahuan, belajar dari kesalahan masa lalu,

kemudian memperbaiki pengetahuannya sehingga dapat dipakai pada masa mendatang.

8. Pengguna *(User)*
9. Pada umumnya pengguna sistem pakar bukanlah seorang pakar *(non-expert)* yang membutuhkan solusi, saran atau pelatihan *(training)* dari berbagai permasalahn yan ada (T.Sutojo; 2011: 167-168).

II.4. *Internet*

Internet merupakan sekumpulan jaringan yang terhubung satu dengan lainnya, di mana jaringan menyediakan sambungan menuju global informasi. Pada umumnya, untuk membangun sebuah jaringan *internet* membutuhkan peralatan jaringan seperti *Repeater* (penguat sinyal), *Bridge* (penghubung antar jaringan), *Router* (pengatur lalu lintas dalam jaringan), dan *Gateway*.

Komputer yang terkoneksi ke *internet* merupakan bagian jaringan. Komputer terhubung ke *internet* dengan menggunakan modem yang terkoneksi ke sebuah *internet Service Provider* (ISP). Kemudian, ISP akan terkoneksi ke dalam sebuah jaringan yang lebih besar, demikian seterusnya. Jadi, *internet* merupakan jaringan yang berisi jaringan.

Jika pengguna ingin terkoneksi ke *internet* menggunakan ISP, maka ISP harus menyediakan tempat atau terminal yang digunakan untuk mengakses jaringan dan disebut *Point of Presence* (POP). (Budi Sutedjo Dharma Oetomo dkk; 2007:117).

II.4.1 Jenis – Jenis Koneksi Internet

1. *Dial-Up*

   Merupakan koneksi komputer ke dalam jaringan melalui modem. Modem digunakan untuk mengubah *sinyal digital* komputer menjadi *sinyal analog* telepon atau sebaliknya.

2. *Digital Subscriber Line (DSL)*

   Teknologi DSL memberikan kecepatan akses *internet* untuk perorangan atau perusahaan.

3. *Internet Cable*

   Dengan menggunakan peralatan yang disebut "*cable modem"* yang menghungkan *PC* dengan TV kabel. Sebuah *cable modem* memberikan layanan TV dan akses *internet*.

4. Satelit

   Akses *internet* dengan menyewa layanan satelit untuk koneksinya.

5. *Wireless*

   Dikenal sebagai *Wireless Fidelity* (Wi-Fi), yaitu hubungan *internet* yang menggunakan frekuensi radio dengan menggunakan peralatan yang disebut *access point.*

6. *Leased Line*

Sama seperti *dial-up, leased line* menggunakan pula koneksi telepon. *Leased line* digunakan untuk menghubungkan perusahaan yang terpisah secara geografis.

7. Kartu Ponsel

   Koneksi *internet* dengan menggunakan kartu ponsel memanfaatkan teknologi *General Packet Radio Service* (GPRS). (Budi Sutedjo Dharma Oetomo dkk; 2007:126-127).

II.5. *Web*

Menurut (Budi Sutedjo Dharma Oetomo dkk; 2007:145) *Web* secara fisik adalah kumpulan komputer pribadi, *web browser*, koneksi ke *ISP*, komputer *server*, *router*, dan *switch* yang digunakan untuk mengalirkan informasi dan menjadi wahana pertama berbagai pihak terkait. *Web* dibagi menjadi beberapa jenis, yaitu :

1. *Web Search Engine*

   Adalah *web* yang memiliki kemampuan untuk melakukan pencarian dokumen berdasarkan kata kunci tertentu.

2. *Web* Portal

   Adalah *web* yang berisi kumpulan *link, search engine,* dan informasi.

3. *Web* Pribadi

Adalah *web* yang memberikan profil pemilik *web*.

II.6. Metode *Certainty Factor*

Menurut (T.Sutojo dkk; 2011:194) Metode *Certainty Factor* dibuat untuk mengakomadasi ketidakpastian pemikiran seorang pakar. Dikarenakan seorang pakar sering kali menganalisis informasi yang ada dengan ungkapan ”mungkin”, ”kemungkinan besar”, ”hampir, maka untuk mengakomodasi hal tersebut maka metode ini guna menggambarkan tingkat keyakinan pakar terhadap masalah yang sedang dihadapi.

Ada dua cara dalam mendapatkan tingkat keyakinan (CF) dari sebuah *rule*, yaitu :

1. Metode *'Net Belief'* yang diusulkan oleh E. H. Shortliffe dan B. G. Buchanan.

   $$CF(Rule) = MB(H,E)$$

   $$MB(H,E) = \begin{cases} 1 & P(H) = 1 \\ \dfrac{\max[P(H \mid E),P(H)] - P(H)}{\max[1,0] - P(H)} & \text{lainnya} \end{cases}$$

   $$MD(H,E) = \begin{cases} 1 & P(H) = 0 \\ \dfrac{\min[P(H \mid E),P(H)] - P(H)}{\max[1,0] - P(H)} & \text{lainnya} \end{cases}$$

Dimana :

CF (Rule) = faktor kepastian

MB(H,E) = *measure of belief* (ukuran kepercayaan) terhadap hipotesis H, jika diberikan *evidence* E (antara 0 dan 1).

MD(H,E) = *measure of disbelief* (ukuran ketidakpercayaan) terhadap *evidenc*e H, jika diberikan *evidence* E (antara 0 dan 1).

P(H) = probabilitas kebenaran hipotesis H

P(H|E) = probabilitas bahwa H benar karena fakta E (T.Sutojo, 2011:194)

2. Dengan mewawancarai seorang pakar

Nilai CF(Rule) didapat dari interprestasi *"term"* dari pakar, yang diubah menjadi nilai CF tertentu.

II.6.1. Perhitungan *Certainty Factor* Gabungan

1. Rule dengan *evidence* E tunggal dan hipotesis H tunggal

   IF E **THEN** H (CF rule)

   CF(H,E) = CF(E) x CF (rule)

   Secara praktik, nilai CF rule ditentukan oleh pakar, sedangkan nilai CF(E) ditentukan oleh pengguna saat berkonsultasi dengan sistem pakar.

   Contoh :

   IF hari ini terang (CF = 0,4) THEN besok hujan (CF = 0,6)

   CF (besok hujan, hari ini terang) = 0,4 x 0,6 = 0,24

Artinya, jika hari ini terang, tingkat kepastian besok hujan adalah 0,24 (T.Sutojo; 2011:197)

2. Rule dengan *evidence* E ganda dan Hipotesis H tunggal

IF E_1 AND E_2.....AND E_n THEN H (CF rule)

$CF(H,E) = \min[CF(E_1), CF(E_2),..., CF(E_n)] \times CF(rule)$

IF E_1 OR E_2.....OR E_n THEN H (CF rule)

$CF(H,E) = \max[CF(E_1), CF(E_2),..., CF(E_n)] \times CF(rule)$

Contoh :

IF demam (CF = 0,4) AND batuk (CF = 0,2) AND muntah (CF = 0,7) THEN penyakit TBC (CF = 0,3)

CF (TBC, demam ∩ batuk ∩ muntah) = min [0,4;0,2;0,7] x 0,3 = 0,2 x 0,3 = 0,06

Artinya, jika gejala deman, batuk dan muntah maka tingkat kepastian terkena penyakit TBC adalah 0,06 (T.Sutojo dkk; 2011:197)

3. Kombinasi dua buah *rule* dengan *evidence* brebeda (E_1 dan E_2), tetapi hipotesa sama.

IF E_1 THEN H Rule 1 $CF(H, E_1) = CF_1 = C(E_1) \times CF(Rule1)$

IF E_2 THEN H Rule 2 $CF(H, E_2) = CF_2 = C(E_2) \times CF(Rule2)$

$$CF(CF_1, CF_2) = \begin{cases} CF_1 + CF_2 \times (1 - CF_1) & \text{jika } CF_1 > 0 \text{ dan } CF_2 > 0 \\ \dfrac{CF_1 + CF_2}{1 - \min[|CF_1|, |CF_2|]} & \text{jika } CF_1 < 0 \text{ atau } CF_2 < 0 \\ CF_1 + CF_2 \times (1 + CF_1) & \text{jika } CF_1 < 0 \text{ dan } CF_2 < 0 \end{cases}$$

(T.Sutojo dkk; 2011:198)

II.6.2. Kelebihan dan Kekurangan Metode *Certainty Factors*

II.6.2.1.Kelebihan metode *Certainty Factors* adalah:

1. Metode ini cocok dipakai dalam sistem pakar untuk mengukur sesuatu apakah pasti atau tidak pasti dalam mendiagnosis penyakit sebagai salah satu contohnya.
2. Perhitungan dengan menggunakan metode ini dalam sekali hitung hanya dapat mengolah dua data saja sehingga keakuratan data dapat terjaga.

II.6.2.2 Kekurangan metode *Certainty Factors* adalah:

1. Pemodelan ketidakpastian yang menggunakan perhitungan metode *certainty factors* biasanya masih diperdebatkan.
2. Untuk data lebih dari 2 buah, harus dilakukan beberapa kali pengolahan data. (T.Sutojo dkk; 2011:204)

II.7. *PHP*

PHP merupakan singkatan dari *PHP Hypertext Preprocessore,* yang merupakan bahasa berbentuk skrip yang ditempatkan dalam server dan diproses diserver, hasilnya yang dikirim ke *client.* Pada dasarnya *PHP* mempunyai fungsi yang sama dengan skrip-skrip seperti ASP *(Active server page),* Cold Fusion

ataupun Perl, namun perlu diketahui bahawa *PHP* sebenarnya bisa dipakai secara *command line.* Artinya skrip *PHP* dapat dijalankan tanpa melibatkan *web server* maupun *browser*.

Konsep kerja *PHP* yaitu dengan model kerja *HTML* diawali dengan permintaan satu halaman web oleh *browser.* Berdasarkan *URL (Uniform Resource Locator)* atau dikenal dengan sebutan alamat internet, *browser* mendapatkan alamat dari *web server* mengidentifikasi halaman yang dikehendaki dan menyampaikan segala informasi yang dibutuhkan oleh *web server*. Selanjutnya *web server* akan mencari *file* yang diminta dan memberikan isinya ke *web browser, browser* yang mendapat isinya segera melakukan proses penerjemahan kode HTML dan menampilkannya ke layar pemakai (Abdul Kadir; 2008:2).

II.8. *MySQL*

MySQL merupakan *software RDBMS* (*server database*) yang dapat mengelola *database* dengan sangat cepat dan dapat menampung data dalam jumlah yang sangat besar, dapt diakses oleh banyak *user (multi-user)* dan dapat melakukan suatu proses secara sinkron atau berbarengan *(multi-threaded). MySQL* banyak digunakan di berbagai kalangan untuk melakukan penyimpanan dan pengelolahan data, mulai dari kalangan akademis sampai industri. Lisensi *MySQL* terbagi menjadi dua, yaitu *MySQL* sebagai produk *open source* di bawah *GNU General Public License* atau dapat membeli lisensi dari versi komersialnya (Budi Raharjo; 2011:21).

Perangkat lunak yang ditujukan untuk mengelola database biasa dinamakan *DBMS (Database Management System). Access, MS SQL Server*, dan *MySQL* merupakan contoh produk pengelola database. Beberapa diantaranya berkelas *database server*, yaitu jenis yang secara aktif memantau permintaan akses terhadap data, dalam hal ini *database server* akan segera menanggapi permintaan data. Pengaksesan data dalam database dapat dilakukan dengan melalui *SQL (Structured Query Language)* (Abdul Kadir; 2009:15).

II.9. *UML*

UML singkatan dari *Unifed Modeling Language* yang berarti bahasa pemodelan standar. (Chonoles, 2003:bab 1) mengatakan sebagai bahasa berarti *UML* memiliki *sintaks* dan semantik. *UML* bukan hanya sekedar diagram tetapi juga menceritakan konteksnya.

UML diaplikasikan untuk maksud tertentu, yaitu :

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya (Prabowo Pudjo Widodo dan Herlawati; 2011:6).

Blok pembangunan *UML* adalah diagram, beberapa diagram ada yang rinci *(timing diagram)* dan lainnya ada yang bersifat umum (misalnya diagram

kelas). Intinya, *UML* merupakan alat komunikasi yang konsisten dalam mensuport para pengembang sistem saat ini.

Ada sembilan jenis diagram, yang dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis, antara lain :

1. Diagram Kelas

   bersifat statis. Diagram ini memperlihatkan himpuan kelas-kelas, antamuka-antarmuka, kolaborasi-kolaborasi, serta relasi,relasi. Diagram ini umum dijumpai pada pemodelan system berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas –kelas aktif. Atribut dari suatu kelas digambarkan dalam dua notasi yang berbeda yaitu atribut *Inline,* atribut relasi, atribut Turunan (*Derived Atributes), Atribut Multiplicity*, *Property Atribut*, Hambatan (*constraints),* dan *Atribut statis*.

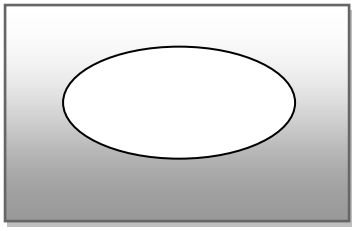
2. Diagram Paket (*Package Diagram*)

   Bersifat statis. Diagram ini memperlihatkan kumpulan kelas – kelas, merupakan bagian dari diagram komponen.

3. *Diagram Use Case*.

   Menurut Pilone (2005:bab 7.1) *use case* menggambarkan fungsi tertentu dalam suatu sistem berupa komponen, kejadian atau kelas. Sedangkan Whitten (2004: 258) mengartikan *use case* urutan langkah-langkah yang secara tindakan saling terkait (skenario), baik terotomatisasi maupun

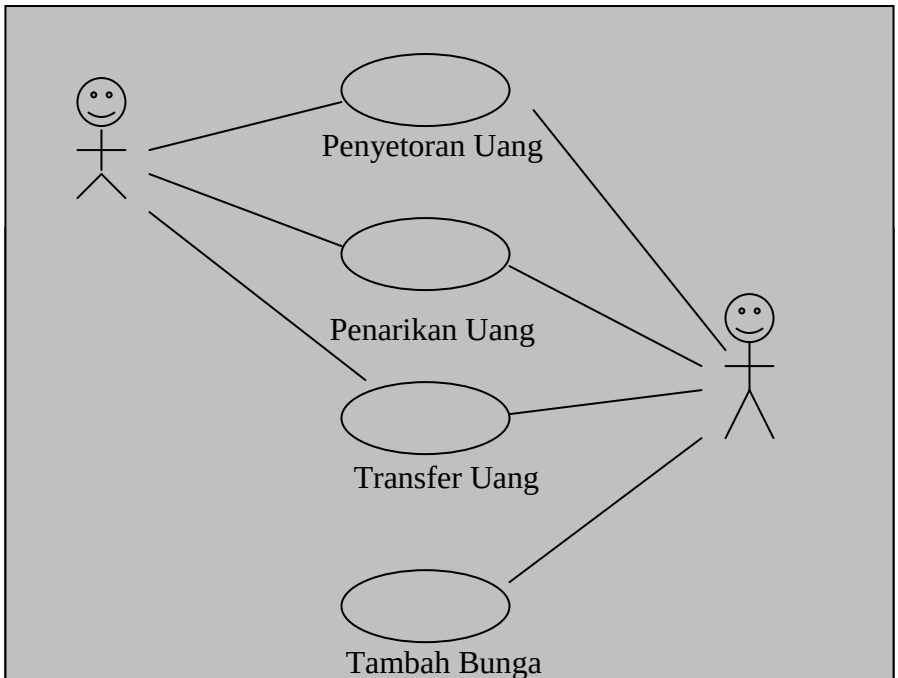
secara manual, untuk tujuan melengkapi satu tugas bisnis tunggal. *Use case* digambarkan dalam bentuk ellips/oval.



Gambar II.3. Simbol *Use Case*

(Sumber : Prabowo Pudjo Widodo dan Herlawati; 2011:22)

Gambar di bawah ini merupakan salah satu contoh bentuk diagram *use case*.



Gambar II.4. Gambar *Use Case*

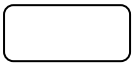
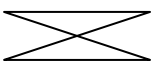
(Sumber : Prabowo Pudjo Widodo dan Herlawati; 2011:17)

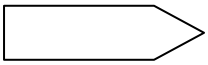
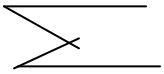
4. Diagram interaksi dan *Sequence* (urutan). Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.
5. Diagram Komunikasi (*Communication Diagram*). Bersifat dinamis. Diagram sebagai pengganti diagram kolaborasi UML 1.4 yang menekankan organisasi structural dari objek – objek yang menerima serta mengirim pesan.
6. Diagram *Statechart* (*Statechart Diagram).* Bersifat dinamis. Diagram status memperlihatkan keadaan – keadaan pada sistem, memuat status

(*state*), transisi, kejadian serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka (*interface*), kelas, kolaborasi dan terutama penting pada pemodelan sistem – sistem yang reaktif.

7. *Actifity Diagram* adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi – fungsi suatu sistem dan member tekanan pada aliran kendali antar objek.

Tabel II.1. : Simbol – Simbol Pada *Actifity Diagram*

Simbol	Keterangan
	Titik Awal
	Titik Akhir
	Actifity
	Pilihan untuk pengambilan keputusan
	Fork; digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	Rake; menunjukkan adanya dekomposisi
	Tanda Waktu

	Tanda pengiriman
	Tanda Penerimaan
	Aliran Akhir (Flow Final)

(Sumber : Munawar ; 2005 : 109-110)

8. Diagram Komponen (*Component Diagram*). Bersifat statis. Diagram ini memperlihatkan organisasi serta kebergantungan sistem/ perangkat lunak pada komponen – komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan ke dalam satu atau lebih kelas – kelas, antarmuka – antarmuka serta kolaborasi – kolaborassi.
9. Diagram *Deployment* (*Deployment Diagram*) Bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time*). Memuat simpul – simpul beserta komponen – komponen yang ada didalamnya.

II. 10. NORMALISASI

Normalisasi merupakan cara pendekatan dalam membangun desain logika basis data relasional yang tidak secara langsung berkaitan dengan model data, tetapi dengan menerapka sejumlah aturan dan kriteria standar untuk menghasilkan struktur tabel yang normal. Pada dasarnya desain logika basis data relasional dapat menggunakan prinsip normalisasi maupun transformasi dari model E-R ke bentuk fisik.

Dalam perspektif normalisasi sebuah database dikatakan baik jika setiap tabel yang membentuk basis data sudah dalam keadaan normal. Suatu tabel dikatakan normal, jika:

1. Jika ada dekomposisi/penguraian tabel, maka dekomposisinya di jamin aman (*lossless-join decomposition*)
2. Terpeliharanya ketergantungan funtional pada saat perubahan data (*dependency preservation*)
3. Tidak melanggar *Boyce Code normal Form* (BCNF), jika tidak bisa minimal tidak melanggar bentuk normalisasi ketiga.

   (Kusrini,M.kom : 2007 : 40).

II.10.1. Bentuk – Bentuk Normalisasi

1. Bentuk tidak normal

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

2. Bentuk normal tahap pertama(1 st Normal Form)

   Defenisi:

Sebuah table disebut 1nf jika:

a. Tidak ada data yang duplikat dalam tabel tersebut
b. Masing-masing cell bernilai tunggal

Catatan: Permintaan yang menyatakan tidak ada data yang duplikat dalam sebuah tabel beraarti 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 ini akan dicontohkan normalisasi dari tabel kuliah yang memiliki atribut:kode_kul,nama-kul,sks,semester,waktu,tempat dan nama_dos.

Tabel kuliah tersebut tidak memenuhi normalisasi pertama,karena terdapat atributi waktu yang tergolong ke dalam atribut bernilai banyak. Agar tabel tersebut dapat memenuhi 1NF,maka solusinya adalah dengan mendokomposisi tabel kuliah menjadi :

a. Tabel kuliah (kode_kull,nama_kul,sks,semester,nama_dos).
b. Tabel jadwal(kode_kul,waktu,ruang).

3. Bentuk normal tahap kedua(2st normal Form)

Bentuk normal kedua 2NF terpenuhi jika pada sebuah tabel semua atribut yang tidak termaksud dalam primary key memiliki ketergantungan fungsional pada primary key secara utuh.

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

Bentuk normal kedua akan di contohkan berikut.

Misal tabel Nilai terdiri dari atribut kode_kul, nim dan nilai. Jika pada tabel Nilai, misalnya kita tambahkan sebuah atribut yang bersifat rerundan. Yaitu nama_mhs,maka tabel Nilai ini dianggap melanggar 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 memiliki 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 terhdap primary key saja.

4. Bentuk normal tahap ketiga (3^{rd} normal Form)

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

   a. X haruslah superkey pada tabel tersebut

   b. Atau A merupakan bagian dari primary key pada tabel tersebut

   Misalnya pada tabel Mahasiswa ,atribut alamat_mhs dipecak ke dalam alamat_jalan.alamat_kota dan kode_pos.Bentuk ini tidak memenuhi

3NF,karena terdapat ketergantungan fungsional baru yang muncul pada tabel tersebut,yaitu:

alamat_jalan nama_kota ->kode_pos

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

Jika tabel Mahasiswa didekomposisi menjadi tabel Mahasiswa dan tabel alamat,maka telah memenuhi 3NF.Hal ini dapat dibuktikan 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 sebagian primary key) dari tabel alamat tersebut.Demikian juga dengan ketergantungan fungsional yang kedua meskipun(kode_pos)bukan merupakan superkey,tetapi nama_kota merupakan dari primary key dari tabel Alamat.Karena telah memenuki 3NF.maka tabel tersebut tidak perlu di-dekomposisis lagi.

5. Bentuk Normal Tahap Keempat dan Kelima

Penerapan aturan normalisasi sampai bentuk normal ketiga sudah memadai 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 depenency)pada sebuah tabel yang merupakan sebuah pengembangan dari ketergantungan fungsional .Adapun bentuk normal tahap kelima merupakan nama lain dari *Project Join Normal Form*(PJNF)

6. *Boyce Code Normal Form* (BNCF)
   a. Memenuhi 1^{st} NF
   b. Relasi harus bergantung fungsi pada atribut *superkey*

(Kusrini,M.kom : 2007 : 40).

II.11. KAMUS DATA

Kamus data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan kamus data sistem analis dapat mendefinisikan data yang mengalir pada sistem yang lengkap.

Kamus data dibuat dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem. Pada tahap analisis kamus data digunakan sebagai alat komunikasi antara sistem analis dengan user tentang data yang mengalir pada sistem tersebut serta informasi yang dibutuhkan oleh pemakai sistem. Untuk keperluan ini maka kamus data harus memuat hal-hal sebagai barikut:

1. Arus Data

Arus data menunjukkan dari mana data mengalirdan kemana data akan menuju.Keterangan arus data ini perlu dicatat di kamus data untuk memudahkan mencari arus data di dalam data flow diagram (DFD).

2. Nama Arus Data

Karena kamus data dibuat berdasarkan arus data yang mengalir di data flow diagram. Maka nama dari arus data juga harus dicatat dikamus data, sehingga mereka yang membaca DFD dan memerlukan penjelasan lebih lanjut tentang suatu arus data tertentu di data flow diangram dapat langsung mencarinya dengan mudah di kamus data.

3. Tipe Data

Telah diketahui bahwa arus data dapat mengalir dari hasil suatu proses ke proses lainnya. Data yang mengalir ini biasanya dalam bentuk laporan serta dokumen hasil cetakan komputer.

4. Struktur Data

Struktur data menunjukkan arus data yang dicatat pada kamus data yang terdiri dari item-item atau elemen-elemen data.

5. Alias

Alias atau nama lain dari data juga harus dituliskan. Alias perlu ditulis karena data yang sama mempunyai nama yang berbeda untuk orang atau departemen lainnya.

6. Volume

Volume yang perlu dicatat di dalam kamus data adalah volume rata-rata dan volume puncak dari arus data. Volume rata-rata menunjukkan

banyaknya arus data yang mengalir dalam satu periode tertentu sementara volume puncak menunjukkan volume yang terbanyak.

7. Periode

Periode ini menunjukkan kapan terjadinya arus data. Periode perlu dicatat di kamus data karena dapat digunakan untuk mengidentifikasikan kapan input data harus dimasukkan kedalam sistem, kapan proses program harus dilakukan dan kapan laporan-laporan harus dihasilkan.

8. Penjelasan

Untuk lebih memperjelas makna dari arus data yang dicatat di kamus data, maka bagian penjelasan dapat diisi dengan keterangan-keterangan tentang arus data tersebut. (Tata Sutabri, S.Kom : 2004 : 170-172)

Kamus data juga mempunyai suatu bentuk mempersingkat arti/makna dari simbol yang dijelaskan, yang disebut notasi. Notasi atau simbol yang digunakan dibagi menjadi 2 macam, yaitu sebagai berikut:

1. Notasi Tipe Data

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

A. Notasi	Keterangan
X	Setiap karakter
9	Angka Numerik
A	Karakter alphabet
Z	Angka nol ditampilkan sebagai spasi kosong

.	Titik, sebagai pemisah ribuan
,	Koma, sebagai pemisah pecahan
-	Hypen, sebagai tanda penghubung (contoh : 021-7500282)
/	Slash, sebagai tanda pembagi (contoh : 24/10/1967)

2. Notasi Struktur Data

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

Notasi	Keterangan
=	Terdiri dari
+	And (dan)
( )	Pilihan (boleh Ya atau Tidak)
{}	Iterasi / pengulangan proses
[]	Pilih salah satu pilihan
I	Pemisah pilihan didalam tanda []
*	Keterangan atau catatan
@	Petunjuk (key field).

(Tata Sutabri ; 2005 : 172)

II. 12. ERD (*Entity Relation Diagram*)

Perancangan basis data dengan menggunakan model *entity relationship* adalah dengan menggunakan *Entity Relationship Diagram* (ERD). Terdapat tiga

notasi dasar yang bekerja pada model E-R yaitu : *entity sets*, *relationship sets dan attributes*. Sebuah *entity* adalah sebuah "benda" (*thing*) atau "objek"(*object*) di dunia nyata yang dapat dibedakan dari semua objek lainnya. *Entity sets* adalah sekumpulan entity yang mempunyai tipe yang sama. Kesamaan tipe ini dapat dilihat dari *atribut/property* yang dimiliki oleh setiap *entity*. Misal :

1. Kumpulan orang yang menyimpan uang pada suatu bank dapat didefenisikan sebagai entity set nasabah.
2. Kumpulan orang yang belajar di perguruan tinggi didefenisikan sebagai mahasiswa. (Kusrini, M.kom : 2007 : 21)