

BAB II

TINJAUAN PUSTAKA

II.1. Artificial Intelligence (Kecerdasan Buatan)

Kecerdasan Buatan berasal dari bahasa Inggris “*Artificial Intelligence*” atau disingkat AI, yaitu *Intelligence* adalah kata sifat yang berarti cerdas, sedangkan *Artificial* artinya buatan. Kecerdasan buatan yang dimaksud di sini merujuk pada mesin yang mampu berpikir, menimbang tindakan yang akan diambil, dan mampu mengambil keputusan seperti yang dilakukan oleh manusia. (Suriyanti; 2013 : 36)

Kecerdasan Buatan adalah ide-ide untuk membuat suatu perangkat lunak komputer yang memiliki kecerdasan sehingga perangkat lunak komputer tersebut dapat melakukan suatu pekerjaan yang dilakukan oleh manusia. Adapun pekerjaan itu adalah berupa konsultasi yang dapat memberikan suatu informasi berupa saran-saran yang akan sangat berguna. Kecerdasan Buatan memungkinkan komputer untuk berpikir dengan cara menyederhanakan program. Dengan cara ini, Kecerdasan Buatan dapat menirukan proses belajar manusia sehingga informasi baru dapat diserap dan digunakan sebagai acuan di masa-masa mendatang.

Kecerdasan atau kepandaian itu didapat berdasarkan pengetahuan dan pengalaman, untuk itu agar perangkat lunak yang dikembangkan dapat mempunyai kecerdasan maka perangkat lunak tersebut harus diberi suatu pengetahuan dan kemampuan untuk menalar dari pengetahuan yang telah didapat dalam menemukan solusi atau kesimpulan layaknya seorang pakar dalam bidang tertentu yang bersifat spesifik. Teori ini dapat dinyatakan dalam bahasa program

komputer dan dibuktikan melalui eksekusinya pada komputer nyata.(Ismail Juriwansyah; 2014 : 9)

II.2. Sistem

Suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial, dan merupakan kegiatan strategi dari suatu organisasi, serta menyediakan laporan-laporan yang diperlukan oleh pihak luar.(Abdul Hamid; 2015 : 2)

II.2.1. Karakteristik Sistem

Menurut Kusrini (2010:6), Sistem mempunyai beberapa karakteristik atau sifat-sifat tertentu antara lain :

a. Komponen sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama untuk membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupaya suatu subsistem atau bagian dari sistem. Setiap sistem tidak peduli berapapun kecilnya, selalu mengandung komponen-komponen atau subsistem.

b. Batasan sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

c. Lingkungan luar sistem (*Environment*)

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut

d. Penghubung sistem (*Interface*)

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui perhubungan ini memungkinkan sumber-sumber daya mengalir dari subsistem yang lainnya.

e. Masukan sistem (*Input*)

Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*).

f. Keluaran sistem (*Output*)

Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

g. Pengolah sistem (*Process*)

Suatu sistem yang dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran.

h. Sasaran sistem (*Objective*)

Suatu sistem pasti mempunyai tujuan atau sasaran. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

II.2.2. Klasifikasi Sistem

Menurut Kusrini (2010:7), Sistem dapat diklasifikasikan dari beberapa sudut pandangan, diantaranya sebagai berikut :

1. Sistem abstrak dan sistem fisik.

Sistem abstrak adalah sistem yang berisi gagasan atau konsep. Misalnya, sistem teologi yang berisi gagasan tentang hubungan manusia dan Tuhan. Sistem fisik merupakan sistem yang secara fisik dapat dilihat. Misalnya sistem komputer, sistem sekolah, sistem akuntansi, dan sistem transportasi.

2. Sistem alamiah dan sistem buatan manusia.

Sistem alamiah adalah sistem yang terjadi karena alam (tidak dibuat manusia). Misalnya, sistem tata surya. Sistem buatan manusia adalah sistem yang dibuat oleh manusia. Misalnya, sistem komputer dan sistem mobil.

3. Sistem tertentu dan sistem tak tentu.

Sistem tertentu adalah sistem yang operasinya dapat diprediksi secara tepat. Misalnya, sistem komputer. Sistem tak tentu adalah sistem yang tak dapat diramal dengan pasti karena mengandung unsur probabilitas. Misalnya, sistem arisan dan sistem sediaan.

4. Sistem tertutup dan sistem terbuka

Sistem tertutup adalah sistem yang tidak bertukar materi, informasi, atau energi dengan lingkungan. Misalnya, reaksi kimia dalam tabung terisolasi. Sistem terbuka adalah sistem yang berhubungan dengan lingkungan dan dipengaruhi oleh lingkungan.

II.3. Pakar

Pakar adalah orang yang memiliki pengetahuan khusus, pendapat, pengalaman dan metode, serta kemampuan untuk mengaplikasikan keahliannya tersebut guna menyelesaikan masalah. (Muhammad Arhami; 2005 : 12)

II.3.1 Sistem Pakar

Sistem pakar adalah suatu sistem komputer yang menyamai (*Emulates*) kemampuan pengambilan keputusan dari seorang pakar. Istilah *emulates* berarti bahwa sistem pakar diharapkan dapat bekerja dalam semua hal seperti seorang pakar. Suatu *emulates* jauh lebih kuat daripada suatu simulasi yang hanya membutuhkan sesuatu yang bersifat nyata dalam beberapa bidang atau hal. (Muhammad Arhami; 2005 : 2)

Seorang pakar dengan sistem pakar mempunyai banyak perbedaan. Di antaranya perbandingan sistem pakar dengan sistem konvensional. Perbedaan ini ditujukan pada Tabel II.1 sebagai berikut:

Tabel II.1 Perbandingan antara Sistem Konvensional dengan Sistem Pakar

Sistem Konvensional	Sistem Pakar
Informasi dan pemrosesannya biasanya digabungkan dalam satu 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 diperlakukan atau bagaimana output dihasilkan.	Penjelasan merupakan bagian terpenting dari semua sistem pakar.
Perubahan program sangat menyulitkan.	Perubahan dalam aturan-aturan mudah untuk dilakukan.
Sistem hanya bisa beroperasi setelah	Sistem dapat beroperasi hanya dengan

lengkap atau selesai.	aturan-aturan yang sedikit (sebagai prototipe awal).
Eksekusi dilakukan berdasarkan langkah demi langkah (<i>algorithmic</i>).	Eksekusi dilakukan dengan menggunakan <i>heuristic</i> dan logika pada seluruh basis pengetahuan.
Perlu informasi lengkap agar bisa beroperasi.	Dapat beroperasi dengan informasi yang tidak lengkap atau mengandung ketidakpastian.
Memanipulasi 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: (Muhammad Arhami; 2005 : 8)

II.3.2. Keuntungan Sistem Pakar

Sistem pakar (*expert system*) merupakan paket perangkat lunak atau paket program komputer yang ditujukan sebagai penyedia nasihat dan sarana bantu dalam memecahkan masalah di bidang-bidang spesialisasi tertentu seperti sains, rekayasa, matematika, kedokteran, pendidikan dan sebagainya.

Ada beberapa keuntungan dalam menggunakan sistem pakar, di antaranya adalah:

1. Menjadikan pengetahuan dan nasihat lebih mudah didapat.
2. Meningkatkan output dan produktivitas.
3. Menyimpan kemampuan dan keahlian pakar.
4. Meningkatkan penyelesaian masalah, menerusi paduan pakar, penerangan, sistem pakar khas.

5. Meningkatkan reliabilitas.
6. Memberikan respons (jawaban) yang cepat.
7. Merupakan panduan yang intelligence (cerdas).
8. Dapat bekerja dengan informasi yang kurang lengkap dan mengandung ketidakpastian.
9. *Intelligence database* (basis data cerdas), bahwa sistem pakar dapat digunakan untuk mengakses basis data dengan cara cerdas.

(Muhammad Arhami : 9-10).

II.3.3. Kelemahan Sistem Pakar

1. Masalah dalam mendapatkan pengetahuan di mana pengetahuan tidak selalu bisa didapatkan dengan mudah, karena kadangkala pakar dari masalah yang kita buat tidak ada.
2. Untuk membuat suatu sistem pakar yang benar-benar berkualitas tinggi sangatlah sulit dan memerlukan biaya yang sangat besar untuk pengembangan dan pemeliharaannya.
3. Boleh jadi sistem yang tidak dapat membuat keputusan.
4. Sistem pakar tidaklah 100% menguntungkan, walaupun seorang tetap tidak sempurna atau tidak selalu benar.

II.3.4. Konsep Dasar Sistem Pakar

Pengetahuan dari suatu sistem pakar mungkin dapat direpresentasikan dalam sejumlah cara. Salah satu metode yang paling umum untuk

merepresentasikan pengetahuan adalah dalam bentuk tipe aturan (*rule*) **IF...THEN** (Jika...Maka).

Turban (1995) menyatakan bahwa konsep dasar dari suatu sistem pakar mengandung beberapa unsur/elemen, yaitu keahlian, ahli, pengalihan keahlian, inferensi, aturan dan kemampuan menjelaskan.

Keahlian merupakan suatu penguasaan pengetahuan di bidang tertentu yang didapatkan dari pelatihan, membaca atau pengalaman. Contoh bentuk pengetahuan yang merupakan keahlian adalah:

1. Fakta-fakta pada lingkup permasalahan tertentu.
2. Teori-teori pada lingkup permasalahan tertentu.
3. Prosedur-prosedur dan aturan-aturan berkenaan dengan lingkup permasalahan tertentu.
4. Strategi-strategi global untuk menyelesaikan masalah.
5. *Meta-Knowledge* (Pengetahuan tentang pengetahuan).

Menurut Turban (1995), terdapat tiga orang yang terlibat dalam lingkungan sistem pakar, yaitu:

1. Pakar

Pakar adalah orang yang memiliki pengetahuan khusus, pendapat, pengalaman dan metode, serta kemampuan untuk mengaplikasikan keahliannya tersebut guna menyelesaikan masalah.

2. *Knowledge Engineer* (Rekayasa Sistem)

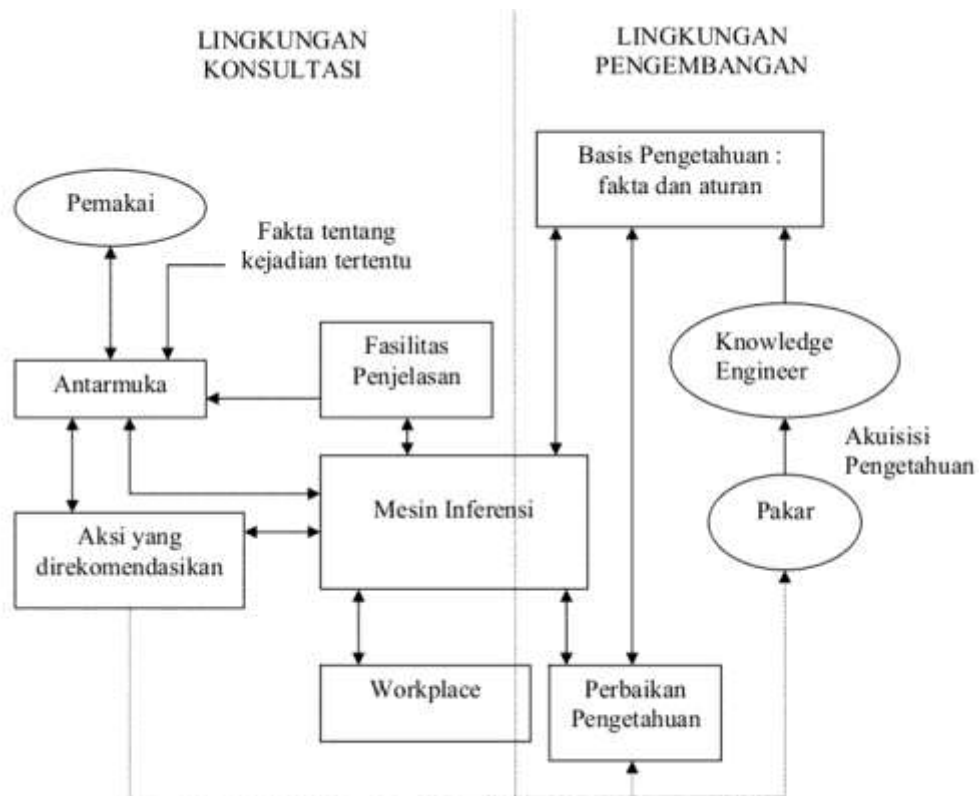
Knowledge Engineer adalah orang yang membantu pakar dalam menyusun area permasalahan dengan menginterpretasikan dan mengintegrasikan jawaban-jawaban pakar atas pertanyaan yang diajukan, menggambarkan analogi, mengajukan *counter example* dan menerangkan kesulitan-kesulitan konseptual.

3. Pemakai

Sistem pakar memiliki beberapa pemakai, yaitu pemakai bukan pakar, pelajar, pembangun sistem pakar yang ingin meningkatkan dan menambah basis pengetahuan dan pakar. (Muhammad Arhami; 2005 : 11-13)

II.3.5. Struktur Sistem Pakar

Sistem pakar disusun oleh dua bagian utama, yaitu lingkungan pengembangan (*Development Environment*) dan lingkungan konsultasi (*Consultation Environment*) (Turban, 1995). Lingkungan pengembangan sistem pakar digunakan untuk memasukkan pengetahuan pakar ke dalam lingkungan sistem pakar, sedangkan lingkungan konsultasi digunakan oleh pengguna yang bukan pakar guna memperoleh pengetahuan pakar. Komponen-komponen sistem pakar dalam kedua bagian tersebut dapat dilihat dalam Gambar II.1 berikut ini:



Gambar II.1 Arsitektur Sistem Pakar

Sumber (Muhammad Arhami; 2005: 13-14)

Keterangan:

1. *Antarmuka (Interface)*

Interface merupakan mekanisme yang digunakan oleh pengguna dan sistem pakar untuk berkomunikasi. Selain itu antarmuka menerima informasi dari sistem dan menyajikannya ke dalam bentuk yang dapat dimengerti oleh pemakai.

2. *Basis Pengetahuan*

Basis pengetahuan mengandung pengetahuan untuk pemahaman, formulasi, dan penyelesaian masalah. Komponen sistem pakar ini disusun atas dua

elemen, yaitu fakta dan aturan. Fakta merupakan informasi tentang obyek dalam area permasalahan tertentu. Sedangkan aturan merupakan informasi tentang cara bagaimana memperoleh fakta baru dari fakta yang telah diketahui.

3. Akuisisi Pengetahuan (*Knowledge Acquisition*)

Akuisisi pengetahuan adalah akumulasi transfer dan transformasi keahlian dalam menyelesaikan masalah dari sumber pengetahuan ke dalam program komputer.

4. Mesin Inferensi

Komponen ini mengandung mekanisme pola pikir dan penalaran yang digunakan oleh pakar dalam menyelesaikan suatu masalah. Mesin inferensi adalah program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan *workplace*, dan untuk memformulasikan kesimpulan (Turban, 1995).

5. Workplace

Workplace merupakan area dari sekumpulan memori kerja (*working memory*). *Workplace* digunakan untuk merekam hasil-hasil antara dan kesimpulan yang dicapai. (Muhammad Arhami; 2005 : 14-22)

II.4. Mesin Air

Mesin air adalah salah satu peralatan listrik yang cukup vital di banyak perumahan. Kerusakan pada unit ini akan mengakibatkan terganggunya pasokan air dari dalam tanah, dan tentu akan sangat mengganggu kegiatan harian di rumah.

Sehingga berangkat dari fakta inilah kita sebaiknya memperhatikan aspek pengoperasian dari mesin pompa air ini. Secara umum, diluar instalasi pipa, mesin pompa air terdiri dari dari tiga bagian besar yaitu :

1. Motor Listrik (Dinamo)

Motor listrik berfungsi sebagai penggerak pompa air, dimana motor listrik ini mengubah energi listrik menjadi energi gerak / putar. Antara motor listrik dan pompa air dihubungkan oleh satu *shaft*.

2. Pompa air dan tabung akumulator.

Pompa air mempunyai bagian yang disebut *impeller* yang juga ikut berputar, sedemikian sehingga air terhisap dari sumbernya melalui pipa masuk (*suction*) dan kemudian didorong keluar dengan tekanan tertentu melalui pipa keluar (*discharge*). Sebelum air keluar di pipa *discharge*, maka air itu melewati dulu sebuah tabung yang berfungsi sebagai akumulator. Cara kerja akumulator ini adalah menyimpan air pada saat tekanan pompa tinggi dan mengeluarkan air saat tekanannya turun. Ada beberapa mesin pompa air yang tidak menggunakan tabung ini.

II.5. Pengertian Dempster-Shafer

Teori *Dempster-Shafer* (DST) merupakan teori matematika dari *evidence*. Teori ini dapat memberikan sebuah cara untuk menggabungkan *evidence* dari beberapa sumber dan mendatangkan atau memberikan tingkat kepercayaan (direpresentasikan melalui fungsi kepercayaan) dimana mengambil dari seluruh

evidence yang tersedia. Teori tersebut pertama kali dikembangkan oleh Arthur P. Dempster dan Glenn Shafer.

Ada berbagai macam penalaran dengan model yang lengkap dan sangat konsisten, tetapi pada kenyataannya banyak permasalahan yang tidak dapat terselesaikan secara lengkap dan konsisten. Ketidakkonsistenan yang tersebut adalah akibat adanya penambahan fakta baru. Penalaran yang seperti itu disebut dengan penalaran *non monotonis*. Untuk mengatasi ketidakkonsistenan tersebut maka dapat menggunakan penalaran dengan teori *Dempster-Shafer*. Secara umum teori *Dempster-Shafer* ditulis dalam suatu interval :

$$[Belief, Plausibility]$$

Belief (*Bel*) adalah ukuran kekuatan *evidence* dalam mendukung suatu himpunan proposisi. Jika bernilai 0 maka mengindikasikan bahwa tidak ada *evidence*, dan jika bernilai 1 menunjukkan adanya kepastian. *Plausibility* (*Pl*) dinotasikan sebagai :

$$Pl(s) = 1 - Bel(\neg s)$$

Plausibility juga bernilai 0 sampai 1. Jika yakin akan $\neg s$, maka dapat dikatakan bahwa $Bel(\neg s)=1$, dan $Pl(\neg s)=0$. Pada teori *Dempster-Shafer* dikenal adanya *frame of discrement* yang dinotasikan dengan θ . *Frame* ini merupakan semesta pembicaraan dari sekumpulan *hipotesis*. Tujuannya adalah mengaitkan ukuran kepercayaan elemen-elemen θ . Tidak semua *evidence* secara langsung mendukung tiap-tiap elemen. Untuk itu perlu adanya probabilitas fungsi densitas (m). Nilai m tidak hanya mendefinisikan elemen-elemen θ saja, namun juga

semua subsetnya. Sehingga jika θ berisi n elemen, maka subset θ adalah 2^n . Jumlah semua m dalam subset θ sama dengan 1. Apabila tidak ada informasi apapun untuk memilih hipotesis, maka nilai :

$$m_{\{\theta\}} = 1,0$$

Apabila diketahui X adalah subset dari θ , dengan m_1 sebagai fungsi densitasnya, dan Y juga merupakan subset dari θ dengan m_2 sebagai fungsi densitasnya, maka dapat dibentuk fungsi kombinasi m_1 dan m_2 sebagai m_3 , yaitu

$$m_3(Z) = \frac{\sum X \cap Y = Z m_1(X) \cdot m_2(Y)}{1 - \sum X \cap Y = \emptyset m_1(X) \cdot m_2(Y)}$$

Sumber (Anis Mistani; 2014 : 76)

II.6. PHP

PHP merupakan singkatan dari *Hypertext Preprocessor*, merupakan bahasa script server-side yang disisipkan pada *HTML (Hypertext Markup language)*. Kemampuan *PHP* yang paling diandalkan dan signifikan adalah dukungan kepada banyak database. Membuat halaman web yang menggunakan data dari data base dengan sangat mudah untuk dilakukan.

PHP merupakan salah satu software pemrograman web. Software ini disebarkan dan dilisensikan sebagai satu perangkat lunak open source. *PHP* merupakan bahasa script serverside yang disisipkan pada *HTML* atau *HyperText Markup Language*. Interpreter *PHP* dalam mengeksekusi kode *PHP* pada sisi server (disebut *server-side*) dan berbeda dengan mesin maya Java yang mengeksekusi program pada sisi *client*. (Sri Yani Putri; 2013 : 122-123)

II.6.1. Kelebihan dan Kekurangan PHP

Banyak sekali kelebihan yang dimiliki *PHP* dibandingkan dengan bahasa program lainnya, diantaranya:

1. Bisa membuat *web* menjadi dinamis.
2. *PHP* bersifat *Open Source* yang berarti dapat digunakan oleh siapa saja secara gratis.
3. Program yang dibuat dengan *PHP* bisa dijalankan oleh semua sistem operasi karena *PHP* berjalan secara *Web Base* yang artinya semua sistem bahkan HP yang mempunyai *Web Browser* dapat menggunakan program *PHP*.
4. Aplikasi *PHP* lebih cepat dibandingkan dengan *ASP* maupun *Java*.

Selain kelebihan *PHP*, *PHP* juga mempunyai kekurangan. Namun masalah kekurangannya sangat sedikit, diantaranya:

1. *PHP* tidak mengenal *Package*.
2. Jika tidak di *encoding*, maka kode *PHP* dapat dibaca semua orang dan untuk meng *encodingnya* dibutuhkan tool dari *Zend* yang mahal sekali biayanya.
3. *PHP* memiliki kelemahan keamanan.

II.7. Basis Data

Menurut Chou basis data merupakan kumpulan informasi yang bermanfaat yang diorganisasikan kedalam tatacara yang khusus. Menurut Fabbry dan Schwab, basis data adalah sistem berkas terpadu yang dirancang terutama untuk meminimalkan pengulangan data. Menurut Date, sistem basis data pada dasarnya adalah sistem terkomputerisasi yang tujuannya adalah memelihara informasi dan membuat informasi tersebut tersedia saat dibutuhkan. (Abdul Kadir; 2003 : 9)

II.7.1. Model Basis Data

Model basis data menyatakan hubungan antar rekaman yang tersimpan dalam basis data. Beberapa literature menggunakan istilah struktur data logis untuk menyatakan keadaan ini.

Model dasar yang paling umum ada 3 macam, yaitu:

1. Hirarkis
2. Jaringan, dan
3. Relasional

(Abdul Kadir; 2003 : 22)

II.7.2. Perancangan Basis Data

Perancangan basis data, terlepas dari masalah yang ditangani, dibagi menjadi 2 tahapan:

1. Perancangan basis data secara konseptual,
2. Perancangan basis data secara logis, dan
3. Perancangan basis data secara fisis.

Perancangan basis data secara konseptual merupakan upaya membuat model yang masih bersifat konsep.

Perancangan basis data secara logis merupakan tahapan untuk memetakan model konseptual ke model basis data yang akan dipakai (model rasional, hirarkis, atau jaringan).

Perancangan basis data secara fisis merupakan tahapan untuk menuangkan perancangan basis data yang bersifat logis menjadi basis data fisis yang tersimpan pada media penyimpan eksternal. (Abdul Kadir; 2003 : 39)

II.8. Normalisasi

Normalisasi Adalah suatu teknik untuk memperbaiki struktur tabel dengan ketergantungan fungsional(KF) dan primary key. Perbaikan dengan memecah sturktur tabel kedalam tabel lebih kecil secara bertahap untuk memenuhi kebutuhanpemakai didalam suatu organisasi. Pemecah tabel menjadi tabel yang lebih kecil disebut *dekomposisi*.

II.8.1. Tujuan Normalisasi

1. Untuk menghilangkan redudan(Rangkap data) yang menyebabkan pemborosan memori.
2. Untuk mengurangi kompleksitas.
3. Untuk menghindari anomali(ketidak konsistenan) saat update data.

II.8.2. Proses Normalisasi

1. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis/diperiksa berdasarkan persyaratan tertentu kebeberapa tingkat(1NF, 2NF, 3NF, dst..)
2. Apabila tabel yang diuji belum memenuhi persyaratan tertentu, maka tabel tersebut perlu dipecah menjadi sederhana sampai memenuhi bentuk yang optimal.

II.9. MySQL

MySQL *database server* adalah RDBMS (*Relasional Database Management Sistem*) yang dapat menangani data yang bervolume besar. meskipun begitu, tidak menuntut *resource* yang besar. MySQL adalah *database* yang paling populer di antara *database* yang lain.

MySQL adalah program *database* yang mampu mengirim dan menerima data dengan sangat cepat dan *multi user*. MySQL memiliki dua bentuk lisensi, yaitu *free software* dan *shareware*. penulis sendiri dalam menjelaskan buku ini menggunakan *database* ini untuk keperluan pribadi atau usaha tanpa harus

membeli atau membayar lisensi, yang berada di bawah lisensi GNU/GPL (*general public license*), yang dapat anda download pada alamat resminya.

Kelebihan lain dari *MySQL* adalah ia menggunakan bahasa *Query* standar yang dimiliki *SQL* (*Structure Query language*). *SQL* adalah suatu bahasa permintaan yang terstruktur yang telah distandarkan untuk semua program pengakses *database* seperti *Oracle*, *Posgres SQL*, *SQL Server*.

(Bunafit Nugroho; 2004 : 29)

II.10. UML

Unified Modelling Language (*UML*) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual (Braun, *et. al.* 2001). Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek. (Haviluddin; 2011 : 1)

II.10.1. Sejarah UML

Sejarah *UML* sendiri terbagi dalam dua fase; sebelum dan sesudah munculnya *UML*. Dalam fase pertama: *UML* sebenarnya sudah mulai diperkenalkan sejak tahun 1990an namun notasi yang dikembangkan oleh para ahli analisis dan desain berbeda-beda, sehingga dapat dikatakan belum memiliki standarisasi. Fase kedua: dilandasi dengan pemikiran untuk mempersatukan metode tersebut dan dimotori oleh Object Management Group (OMG) maka

pengembangan *UML* dimulai pada akhir tahun 1994 ketika Grady Booch dengan metode OOD (*Object-Oriented Design*), Jim Rumbaugh dengan metode OMT (*Object Modelling Technique*) mereka ini bekerja pada Rational Software Corporation dan Ivar Jacobson dengan metode OOSE (*Object-Oriented Software Engineering*) yang bekerja pada perusahaan Objectory Rational.

Sebagai pencetus metode-metode tersebut mereka bertiga berinisiatif untuk menciptakan bahasa pemodelan terpadu sehingga pada tahun 1996 mereka berhasil merilis *UML* versi 0.9 dan 0.91 melalui Request for Proposal (RFP) yang dikeluarkan oleh OMG (Braun, *et.al.* 2001).

Kemudian pada Januari 1997 IBM, ObjecTime, Platinum Technology, Ptech, Taskon, Reich Technologies dan Softeam juga menanggapi Request for Proposal (RFP) yang dikeluarkan oleh OMG tersebut dan menyatakan kesediaan untuk bergabung. (Haviluddin; 2011 : 1).

II.10.2. Tujuan *UML*

Tujuan utama dalam *UML* adalah (Sugrue J. 2009) :

1. Menyediakan bagi pengguna (analisis dan desain sistem) suatu bahasa pemodelan visual yang ekspresif sehingga mereka dapat mengembangkan dan melakukan pertukaran model data yang bermakna.
2. Menyediakan mekanisme yang spesialisasi untuk memperluas konsep inti.
3. Karena merupakan bahasa pemodelan visual dalam proses pembangunannya maka *UML* bersifat independen terhadap bahasa pemrograman tertentu.

4. Memberikan dasar formal untuk pemahaman bahasa pemodelan.
5. Mendorong pertumbuhan pasar terhadap penggunaan alat desain system yang berorientasi objek (OO).
6. Mendukung konsep pembangunan tingkat yang lebih tinggi seperti kolaborasi, kerangka, pola dan komponen terhadap suatu sistem.
7. Memiliki integrasi praktik terbaik.

(Haviluddin; 2011 : 2)

II.10.3. Notasi UML

Berikut ini beberapa notasi dalam *UML*, diantaranya:

1. *Actor*; menentukan peran yang dimainkan oleh *user* atau sistem lain yang berinteraksi dengan subjek. *Actor* adalah segala sesuatu yang berinteraksi langsung dengan sistem aplikasi komputer, seperti orang, benda atau lainnya. Tugas *actor* adalah memberikan informasi kepada sistem dan dapat memerintahkan sistem untuk melakukan sesuatu tugas.

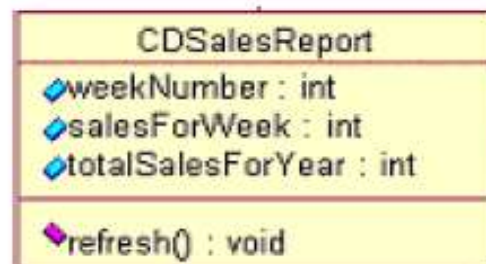


Gambar II.2. Notasi Aktor

(Sumber : Haviluddin; 2011)

2. Class Diagram

Notasi utama dan yang paling mendasar pada diagram *UML* adalah notasi untuk mempresentasikan suatu *class* beserta dengan atribut dan operasinya. *Class* adalah pembentuk utama dari sistem berorientasi objek.



Gambar II.3. Class Diagram

(Sumber : Haviluddin; 2011)

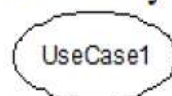
3. Use Case

Use case adalah deskripsi fungsi dari sebuah sistem perspektif pengguna.

Use case bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut skenario.

Use case merupakan awal yang sangat baik untuk setiap fase pengembangan berbasis objek, design, testing, dan dokumentasi yang menggambarkan kebutuhan sistem dari sudut pandang di luar sistem.

Use-case symbol



Gambar II.4. Use Case

(Sumber : Haviluddin; 2011)

4. *Realization*

Realization menunjukkan hubungan bahwa elemen yang ada di bagian tanpa panah akan merealisasikan apa yang dinyatakan oleh elemen yang ada di bagian dengan panah.



Gambar II.5. *Realization*

(Sumber : Haviluddin; 2011)

5. *Interaction*

Interaction digunakan untuk menunjukkan baik aliran pesan atau informasi antar obyek maupun hubungan antar obyek.



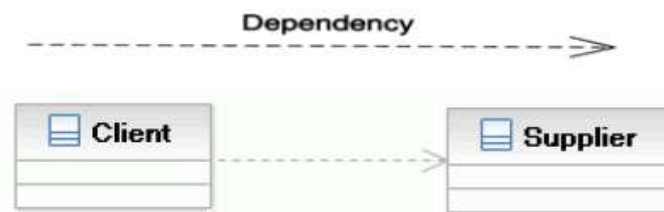
Gambar II.6. *Interaction*

(Sumber : Haviluddin; 2011)

6. *Dependency*

Dependency merupakan relasi yang menunjukkan bahwa perubahan pada salah satu elemen memberi pengaruh pada elemen lain. Terdapat 2 *stereotype* dari *dependency*, yaitu *include* dan *extend*. Include menunjukkan bahwa suatu bagian dari elemen (yang ada digaris tanpa

panah) memicu eksekusi bagian dari elemen lain (yang ada di garis dengan panah). *Extend* menunjukkan bahwa suatu bagian dari elemen di garis tanpa panah bisa disisipkan ke dalam elemen yang ada di garis dengan panah.

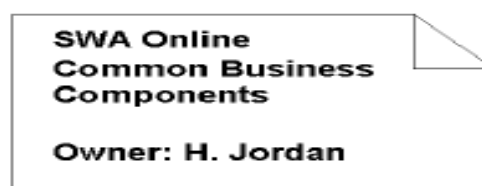


Gambar II.7. *Dependency*

(Sumber : Havaluddin; 2011)

7. *Note*

Note digunakan untuk memberikan keterangan atau komentar tambahan dari suatu elemen sehingga bisa langsung terlampir dalam model. *Note* ini bisa disertakan ke semua elemen notasi yang lain.

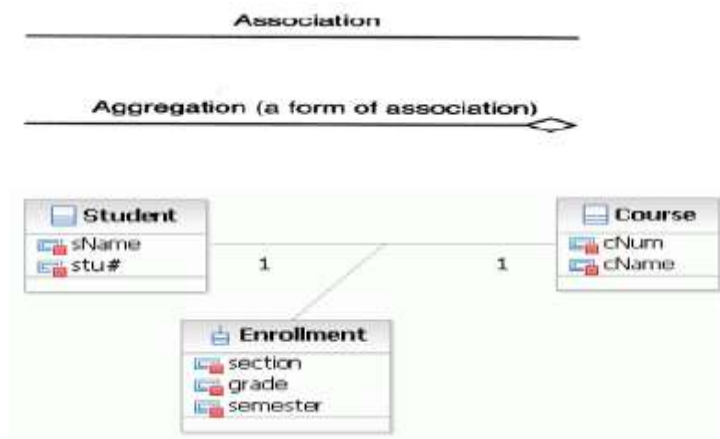


Gambar II.8. *Note*

(Sumber : Havaluddin; 2011)

8. Association

Association menggambarkan navigasi antar *class* (*navigation*), berapa banyak obyek lain yang bisa berhubungan dengan satu obyek (*multiplicity* antar *class*) dan apakah suatu *class* menjadi bagian dari *class* lainnya (*aggregation*).

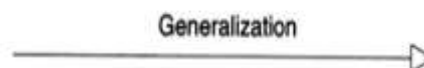


Gambar II.9. Association

(Sumber : Haviluddin; 2011)

9. Generalization

Generalization menunjukkan hubungan antara elemen yang lebih umum ke elemen yang lebih spesifik.

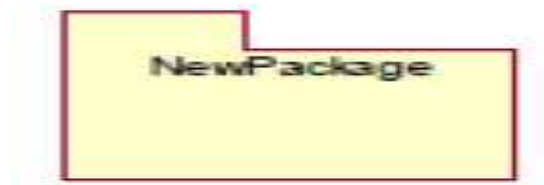


Gambar II.10. Generalization

(Sumber : Haviluddin; 2011)

10. Package

Package adalah mekanisme pengelompokan yang digunakan untuk menandakan pengelompokan elemen-elemen model.

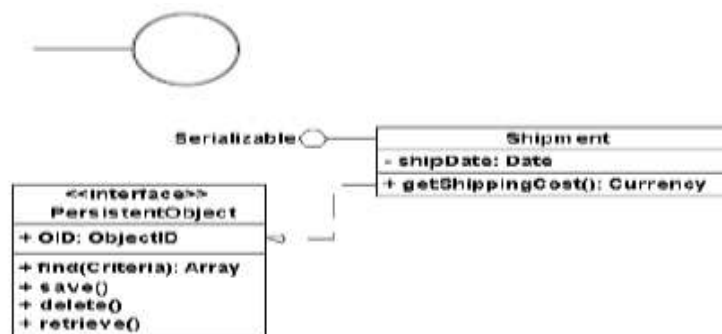


Gambar II.11. Package

(Sumber : Havaluddin; 2011)

11. Interface

Interface merupakan kumpulan operasi berupa implementasi dari suatu *class*. Atau dengan kata lain implementasi operasi dalam *interface* dijabarkan oleh operasi di dalam *class*.



Gambar II.12. Interface

(Sumber : Havaluddin; 2011)

II.11. Pengertian Web

Menurut Ahmad Hasnan, *web* merupakan tempat untuk memberikan informasi berbasis elektronik berupa multimedia, teks, grafis serta segala data dan informasi yang dapat dimasukkan dalam format – format tersebut, umumnya bahasa yang digunakan untuk memberikan informasi berbasis *HTML*. Saat ini *world wide web* merupakan bagian yang paling terkenal dari internet sebab kemudahan dan fleksibilitas dalam penyampaian informasi.

Web bekerja didasarkan pada jaringan internet, internet adalah *network of interconnected networks*. Artinya jaringan yang menghubungkan jaringan – jaringan lainnya. Ketika berselancar di internet dan melihat berbagai macam bentuk dan jenis *web* disitulah anda dapat melihatan begitu hebatnya *web*, sebab *web* dapat menampung berbagai jenis data seperti text, grafis, animasi, foto, suara dan video. Halaman *web* ditulis dengan bahasa pemrograman yang disebut *Hypertext Markup Language* atau *HTML*, file *HTML* berisi markup tags, markup tags inilah yang memberi tahu *browser* bagaimana halaman *web* ditampilkan, terdapat dua extension file *HTML* yang dapat digunakan yaitu *.htm* dan *.html*.