

BAB II

LANDASAN TEORI

2.1 Sekilas Tentang Sistem Pakar

Sistem pakar adalah aplikasi berbasis computer yang digunakan untuk menyelesaikan masalah sebagaimana yang dipikirkan oleh pakar. Pakar yang dimaksud disini adalah orang yang mempunyai keahlian khusus yang dapat menyelesaikan masalah yang tidak dapat diselesaikan oleh orang awam. Sebagai contoh, dokter adalah seorang pakar yang mampu mendiagnosa penyakit yang diderita pasien serta dapat memberikan penatalaksanaan terhadap penyakit tersebut. Tidak semua orang dapat mengambil keputusan mengenai diagnose dan memberikan penatalaksanaan suatu penyakit. Contoh yang lain, montir adalah seorang yang punya keahlian dan pengalaman dalam menyelesaikan kerusakan mesin motor atau mobil. Psikologi adalah orang yang ahli dalam memahami kepribadian seseorang dan lain-lain.

Sistem pakar, yang mencoba memecahkan masalah yang biasanya hanya bisa dipecahkan oleh seorang pakar, dipandang berhasil ketika mampu mengambil keputusan seperti yang dilakukan oleh pakar aslinya baik dari sisi proses pengambilan keputusannya maupun hasil keputusan yang diperoleh.

Sebuah sistem pakar memiliki 2 komponen utama yaitu basis pengetahuan dan mesin inferensi. Basis pengetahuan merupakan tempat penyimpanan pengetahuan dalam memori computer, dimana pengetahuan ini diambil dari pengetahuan pakar.

Ada banyak cara untuk mempresentasikan pengetahuan, diantaranya adalah logika (logic), jaringan semantik (semantic nets). Object Atribut Value (OAV), bingkai (frame), dan kaidah produksi (production rule) (Kusrini, 2007). Buku ini hanya membahas metode representasi pengetahuan dengan menggunakan kaidah produksi.

Mesin inferensi merupakan otak dari aplikasi sistem pakar. Bagian inilah yang menuntun user untuk memasukkan fakta sehingga diperoleh suatu kesimpulan. Apa yang dilakukan oleh mesin inferensi ini didasarkan pada pengetahuan yang ada dalam basis pengetahuan.

2.1.1 Data

Data merupakan representasi fakta mengenai suatu objek atau kejadian. Misalnya:

1. Data mengenai biodata seseorang

Nama : Paijo
Alamat : Sukoharjo
Jenis Kelamin : Laki-Laki

2. Data mengenai identitas suatu barang

Nama : Kursi
Bahan : Kayu Kamper
Warna : Hijau Tua

3. Data mengenai suatu transaksi penjualan

Nota : 05
Tanggal : 1 Januari 2008
Pembeli : Paijo
Barang : Kursi, Meja
Harga : Rp. 200.000,- , Rp. 500.000,-
Jumlah : 4, 1

2.1.2 Informasi

Informasi merupakan data yang sudah diolah sedemikian rupa sehingga sesuai dengan yang dibutuhkan oleh penggunanya, sebagai contoh :

1. Informasi pelanggan yang sering membeli di Toko X (sebagai dasar penentuan pelanggan yang akan diberi bingkisan untuk lebaran)

Nama	Alamat	Jenis Kelamin
Paijo	Sukoharjo	Laki-Laki
Imin	Bantul	Laki-Laki

2. Informasi mengenai jumlah keuntungan dari penjualan bulan Januari 2007 adalah:

Keuntungan penjualan bulan Januari 2007: Rp. 2.000.000,-

Pengetahuan merupakan saringan/intisari dari informasi. Pengetahuan ini lebih umum, tetapi mungkin tidak lengkap dan lebih fuzzy. Pengetahuan bisa berisi fakta, informasi, konsep, prosedur, model, dan heuristic yang dapat digunakan untuk menyelesaikan suatu masalah. Contoh pengetahuan diantaranya :

1. Orang yang beralamat Sukoharjo suka mebel dengan warna-warna cerah.
2. Jika seseorang membeli meja, kemungkinan besar akan membeli kursi juga.

Pengetahuan diklasifikasikan menjadi:

1. Pengetahuan procedural (procedural knowledge) – lebih menekankan pada bagaimana melakukan sesuatu. Contoh:
 2. Pengetahuan tentang bagaimana mencuci dengan menggunakan mesin
 3. Pengetahuan tentang bagaimana membuat pudding
 4. Pengetahuan tentang bagaimana cara mengobati luka bakar
5. Pengetahuan deklaratif (declarative knowledge) – menjawab pertanyaan apakah sesuatu bernilai salah atau benar. Contoh:
 - a. Jangan berikan pisau pada anak di bawah umur 3 tahun
 - b. Buah apel berwarna hijau dan berbentuk bulat
 - c. Ada asosiasi positif antara merokok dan kanker
6. Pengetahuan tacit (tacit knowledge) – pengetahuan yang tidak bisa diungkapkan dengan bahasa
 - a. Bagaimana cara mengayuh sepeda
 - b. Bagaimana cara berjinjit untuk menari balet

2.1.3 Representasi Pengetahuan

Representasi pengetahuan merupakan metode yang digunakan untuk mengkodekan pengetahuan dalam sebuah system pakar. Representasi dimaksudkan untuk menangkap sifat-sifat penting masalah dan membuat informasi itu dapat diakses oleh prosedur pemecahan masalah.

Adapun karakteristik dari metode representasi pengetahuan adalah :

1. Harus bisa deprogram dengan bahasa pemrograman atau dengan shells dan hasilnya disimpan dalam memori.
2. Dirancang sedemikian sehingganya isinya dapat digunakan untuk proses penalaran.
3. Model representasi pengetahuan merupakan sebuah struktur data yang dapat dimanipulasi oleh mesin inferensi dan pencarian untuk aktivitas pencocokan pola. Seperti telah disampaikan, bahwa dalam system pakar ada beberapa metode representasi pengetahuan. Jika pengetahuan berupa pengetahuan yang bersifat deklaratif, maka metode representasi pengetahuan yang cocok adalah jaringan semantic, frame dan logika predikat. Tetapi jika pengetahuannya berupa pengetahuan procedural yang merepresentasikan aksi dan prosedur, maka metode representasi pengetahuan yang cocok adalah kaidah produksi.

2.1.4 Kaidah Produksi

Pengetahuan dalam kaidah produksi direpresentasikan dalam bentuk:

JIKA [antecedent] MAKA [konsekuen]

JIKA [kondisi] MAKA [aksi]

JIKA [premis] MAKA [konklusi]

Contoh:

Aturan 1:

JIKA terjadi luka

MAKA berikan Betadine

Aturan 2:

JIKA tidak punya uang Cash

MAKA ambil tabungan

Aturan 3:

JIKA bersin bersin

MAKA terserang influenza



Aturan 4:

JIKA bensin habis

MAKA Motor tidak akan hidup

Aturan terkadang menggunakan operator logika AND atau OR. Misalnya:

Aturan 5:

JIKA dana mencukupi

DAN pengiriman bisa dilakukan kurang dari 1 bulan

MAKA beli laser printer

Aturan 6:

JIKA kontraktor tidak bisa menyelesaikan pekerjaan tepat waktu

ATAU biaya melebihi anggaran

MAKA kontrak batal

Aturan dalam kaidah produksi diklasifikasikan menjadi kaidah derajat pertama dan kaidah meta. Kaidah derajat pertama adalah aturan yang bagian konklusinya tidak menjadi premis bagi kaidah lain. Sebaliknya, kaidah meta merupakan kaidah yang konklusinya merupakan premis bagi kaidah yang lain. Kaidah meta merupakan kaidah yang berisi penjelasan bagi kaidah yang lain. Aturan

7 merupakan contoh kaidah meta dan aturan 8 merupakan contoh kaidah derajat pertama.

Aturan 7:

JIKA Pusing

DAN Cepat lelah

DAN Sering kesemutan

MAKA Anemia

Aturan 8:

JIKA Anemia

DAN Batuk kronis

MAKA TBC

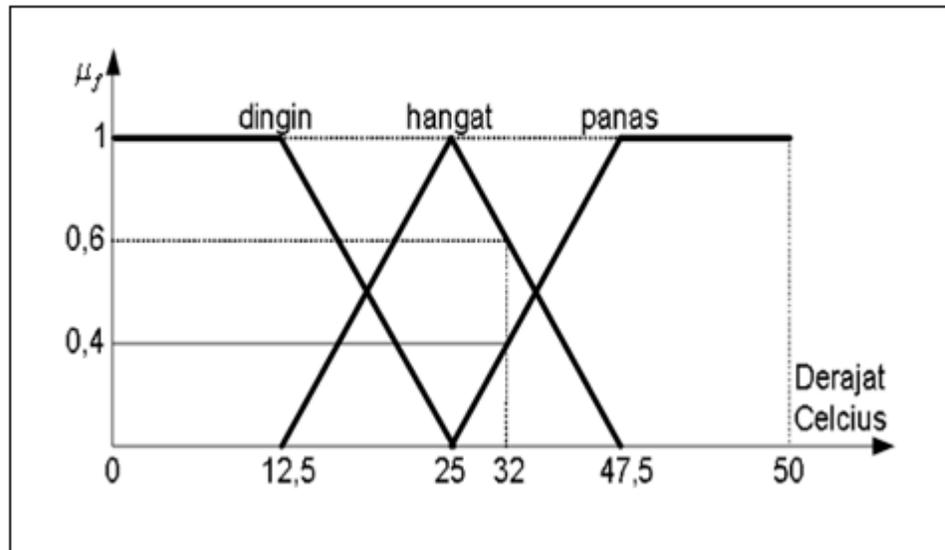
Untuk hasil yang lebih akurat dalam memecahkan masalah dalam suatu domain, biasanya dibutuhkan aturan yang cukup banyak karena masing-masing aturan berisi detail pengetahuan. Jumlah aturan akan menggambarkan kompleksitas sistem pakar.

2.2 Logika Fuzzy

Logika *Fuzzy* adalah suatu cara yang tepat untuk memetakan suatu ruang *input* ke dalam ruang *output*. Untuk sistem yang sangat rumit, penggunaan logika *fuzzy* (*fuzzylogic*) adalah salah satu pemecahannya. Sistem tradisional dirancang untuk mengontrol keluaran tunggal yang berasal dari beberapa masukan yang tidak

saling berhubungan. Karena ketidaktergantungan ini, penambahan masukan yang baru akan memperumit proses kontrol dan membutuhkan proses perhitungan kembali dari semua fungsi. Kebalikannya, penambahan masukan baru pada sistem *fuzzy*, yaitu sistem yang bekerja berdasarkan prinsip-prinsip logika *fuzzy*, hanya membutuhkan penambahan fungsi keanggotaan yang baru dan aturan-aturan yang berhubungan dengannya.

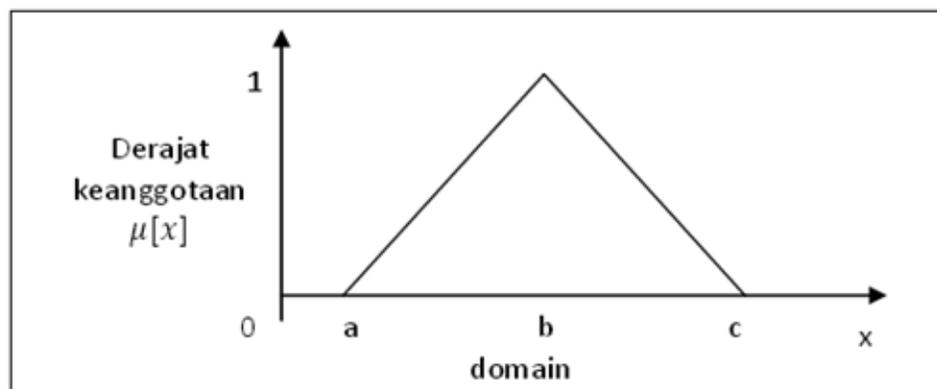
Pada suatu kendali proses, data yang dihasilkan berbentuk tegas/nyata (crisp). Fuzifikasi diperlukan untuk mengubah masukan tegas/nyata (crisp inputs) yang bersifat bukan Fuzzy ke dalam himpunan Fuzzy menjadi nilai Fuzzy dari beberapa variabel linguistic masukan yang telah didefinisikan. Hasil perubahan data inilah yang dinamakan masukan Fuzzy (Fuzzy Inputs). Berikut ini adalah contoh fuzzifikasi. Jika suhu air merupakan suatu variabel linguistik, dengan nilai linguistik $T(\text{suhu}) = \{\text{dingin, hangat, panas}\}$, dimana semesta pembicaraannya terletak antara suhu 0°C sampai dengan suhu 50°C . Distribusi fungsi keanggotaan segitiga pada Fuzzy diperlihatkan pada Gambar 2.1



Gambar 2.1 Proses Himpunan Fuzzy (Kusrini, 2007)

Jika masukan adalah suhu sebesar 32°C, maka proses Fuzzifikasi menghasilkan nilai linguistik dingin dengan derajat keanggotaan 0, hangat dengan derajat keanggotaan 0,6 dan panas derajat keanggotaannya 0,4.

Kurva Segitiga pada dasarnya merupakan gabungan antara dua garis (linear) seperti terlihat pada Gambar 2.2.



Gambar 2.2 Representasi Kurva Segitiga (Kusrini, 2007)

Fungsi Keanggotaan:

$$\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq c \\ \frac{(x-a)}{(b-a)}; & a \leq x \leq b \\ \frac{(c-x)}{(c-b)}; & b \leq x \leq c \end{cases} \quad (2.6)$$

Keterangan :

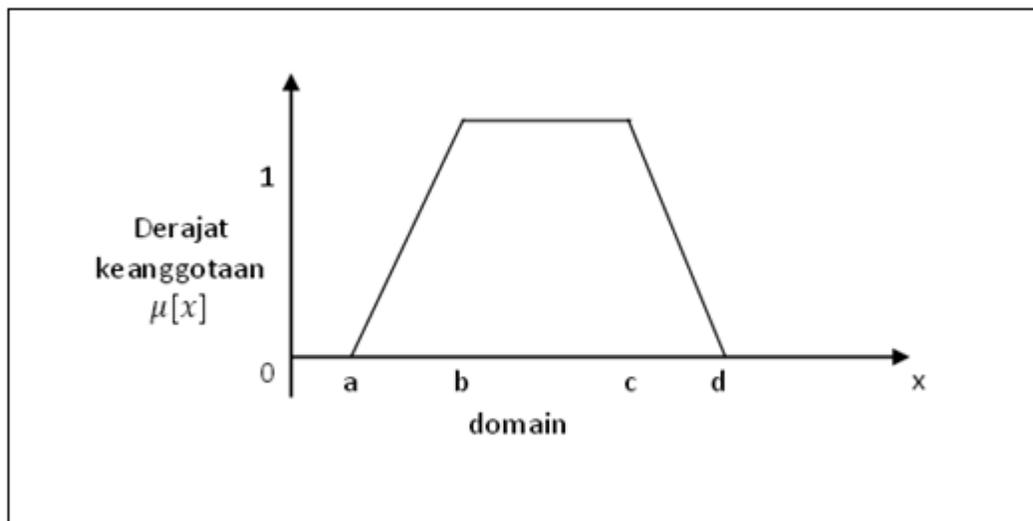
A = nilai domain terkecil yang mempunyai derajat keanggotaan nol

B = nilai domain yang mempunyai derajat keanggotaan Satu

C = nilai domain terbesar yang mempunyai derajat keanggotaan nol

X= nilai input yang akan diubah kedalam bilangan fuzzy

Kurva trapezium pada dasarnya seperti bentuk segitigakarena merupakan gabungan antara dua garis (linear), hanya saja ada beberapa titik yang memiliki nilai keanggotaan 1. Representasi kurva trapesium ditunjukkan pada Gambar 2.3.



Gambar 2.3 Representasi Kurva Trapezium (Kusrini, 2007)

Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq d \\ \frac{(x-a)}{(b-a)}; & a \leq x \leq b \\ 1; & b \leq x \leq c \\ \frac{(d-x)}{(d-c)}; & x \geq d \end{cases} \quad (2.7)$$

Keterangan :

A = nilai domain terkecil yang mempunyai derajat keanggotaan nol

B = nilai domain terkecil yang mempunyai derajat keanggotaan satu

C = nilai domain terbesar yang mempunyai derajat keanggotaan satu

D = nilai domain terbesar yang mempunyai derajat keanggotaan nol

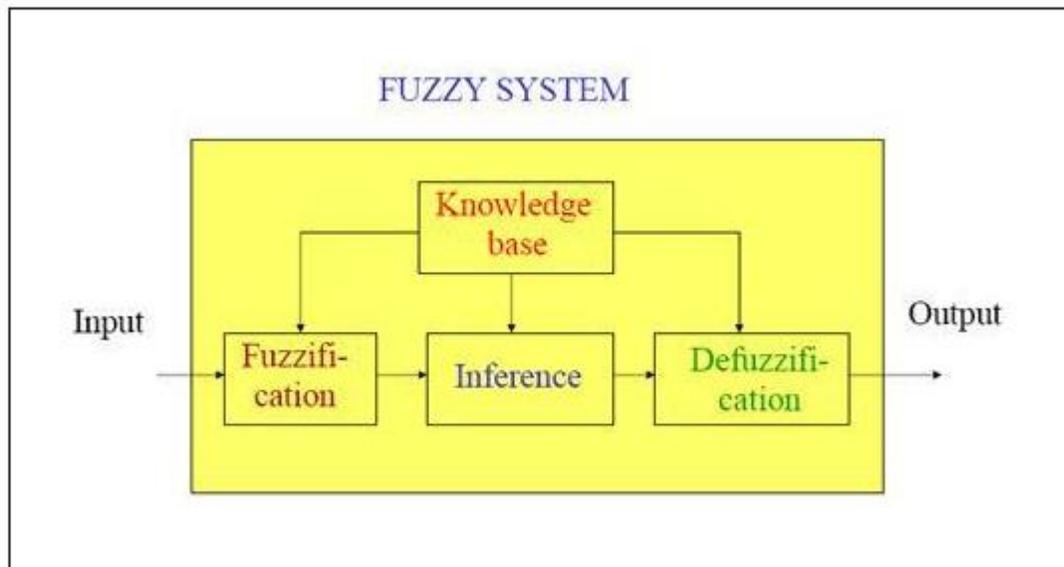
x = nilai input yang akan di ubah ke dalam bilangan fuzzy

Inferensi adalah melakukan penalaran menggunakan fuzzyinput dan fuzzyrules yang telah ditentukan sehingga menghasilkanfuzzyoutput. Secara sintaks, suatu fuzzyrule (aturan fuzzy) dituliskan sebagaiberikut:IF antecedent THEN consequent.

Salah satu aplikasi logika fuzzyyang telah berkembang amat luas dewasa ini adalah sistem inferensi fuzzy(Fuzzy Inference System/FIS), yaitu kerangka komputasi yang didasarkan pada teori himpunan fuzzy, aturan fuzzy berbentuk IF-THEN, dan penalaran fuzzy.Misalnyadalam penentuan status gizi, produksi barang, sistem pendukung keputusan, penentuan kebutuhan kalori harian, dan sebagainya.

Ada tiga metode dalam sistem inferensi fuzzy yang sering digunakan, yaitu metode Tsukamoto, metode Mamdani, dan metode Takagi Sugeno. Dalam penelitian ini akan dibahas penentuan status gizi menggunakan metode Mamdani. Sistem ini berfungsi untuk mengambil keputusan melalui proses tertentu dengan mempergunakan aturan inferensi berdasarkan logika fuzzy.

Dibawah ini kerangka system fuzzy (Kusrini, 2007), yang terdiri dari input – fuzzifikasi- inferensial – knowledge – defuzzifikasi – output.



Gambar 2.4 Proses Fuzzifikasi (Kusrini, 2007)

Keterangan:

1. Fuzzifikasi adalah proses untuk mengubah variabel non fuzzy (variabel numerik) menjadi variabel fuzzy (variabel linguistik).

2. Inferencing (Ruled Based), pada umumnya aturan-aturan fuzzy dinyatakan dalam bentuk “IF.....THEN” yang merupakan inti dari relasi fuzzy.
3. Defuzifikasi adalah proses pengubahan data-data fuzzy tersebut menjadi data-data numerik yang dapat dikirimkan ke peralatan pengendalian.
4. Setelah diperoleh nilai linguistik, maka dilanjutkan dengan membuat aturan (Ruled Based) dengan menggunakan pola IF <premis> Then <konsekuensi>. Aturan tersebut nanti akan diperoleh dari hasil wawancara dengan dua orang pakar. Aturan yang telah dibuat kemudian dimasukkan ke dalam metode Mamdani. Lalu range nilai-nilai pada input nanti tergantung pada dosen pengajar sebagai pakarnya, termasuk KNOWLEDGE IF-THEN setelah dibuat awal nanti dapat diubah dari pakar.

2.3 Metode Mamdani

Metode Mamdani sering juga dikenal dengan nama Metode Max-Min. Untuk mendapatkan output, diperlukan 4 tahapan: (Ebrahim Mamdani-1975)

- a. Pembentukan himpunan fuzzy. Pada proses fuzzifikasi langkah yang pertama adalah menentukan variabel fuzzy dan himpunan fuzzinya. Kemudian tentukan derajat kesepadanan (degree of match) antara data masukan fuzzy dengan himpunan fuzzy yang telah didefinisikan untuk setiap variabel masukan sistem dari setiap aturan fuzzy. Pada metode Mamdani, baik variabel input maupun variabel output menjadi satu atau lebih himpunan fuzzy.

- b. Aplikasi fungsi implikasi pada metode mamdani. Fungsi implikasi yang digunakan adalah min. Lakukan implikasi fuzzy berdasar pada kuat penyulutan dan himpunan fuzzy terdefinisi untuk setiap variabel keluaran di dalam bagian konsekuensi dari setiap aturan. Hasil implikasi fuzzy dari setiap aturan ini kemudian digabungkan untuk menghasilkan keluaran infrensi fuzzy. (Kusumadewi, 2003).
- c. Komposisi Aturan. Tidak seperti penalaran monoton, apabila sistem terdiri dari beberapa aturan, maka infrensi diperoleh dari kumpulan dan korelasi antar aturan. Ada 3 metode yang digunakan dalam melakukan inferensi sistem fuzzy, yaitu: max, additive dan probabilistik OR.
- d. Penegasan (defuzzy). Input dari proses defuzzifikasi adalah suatu himpunan fuzzy yang diperoleh dari komposisi aturan-aturan fuzzy, sedangkan output yang dihasilkan merupakan suatu bilangan pada domain himpunan fuzzy tersebut.

2.4 Aplikasi Web dan Database

2.4.1 Aplikasi Web

Perkembangan dunia internet telah melahirkan suatu fasilitas layan baru, yaitu web, yang merupakan layanan terpenting internet. Fasilitas web mengizinkan pengakses untuk mengakses dan berinteraksi dengan teks, grafik, animasi, foto, suara dan video. 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 sebagai pihak terkait.

Web dikenal dengan sistem client-server. Komputer pengguna disebut komputer client, sedangkan komputer yang di akses disebut server. Web menggunakan protokol yang disebut HTTP (*HyperText Transfer Protocol*) yang berjalan pada TCP/IP. Perekat antarhalaman web disebut *Hyperlink* dan *Hypertext*. Agar dapat menjelajahi web, kita membutuhkan perangkat lunak yang disebut web browser. Kemudian, membangun sebuah web ditulis pada bahasa komputer yang dikenal *Hypertext Markup Language* (HTML).

Pada prakteknya, bahasa komputer berbasis web tidak hanya HTML, melainkan juga melibatkan bahasa pemrograman lain seperti PHP atau Perl. Tujuannya adalah untuk membentuk halaman yang bersifat dinamis. Aplikasi web itu sendiri dibagi menjadi dua, Web Statis dan Web Dinamis. Web statis dibentuk dengan menggunakan HTML saja. Kekurangan aplikasi seperti ini terletak pada keharusan untuk memelihara program secara terus-menerus untuk mengikuti setiap perubahan yang terjadi. Kelemahan ini diatasi dengan model aplikasi web dinamis. Dengan memperluas kemampuan HTML, yakni dengan menggunakan perangkat lunak tambahan, perubahan informasi dalam halaman-halaman web dapat ditangani melalui perubahan data, bukan melalui perubahan program. Sebagai implementasinya, aplikasi dapat dikoneksikan ke basis data. Dengan demikian, perubahan informasi dapat dilakukan oleh operator atau yang bertanggung jawab terhadap kemuktakiran data, dan tidak menjadi tanggung jawab pemrograman atau Webmaster.

2.4.2 Database

Menurut Bambang Hariyanto (2008), data adalah rekaman mengenai fenomena/fakta yang ada atau yang terjadi. Data pada pokoknya adalah refleksi fakta yang ada. Data mengenai fakta-fakta penting organisasi harus dikelola secara baik sehingga dapat dipakai/diakses secara efisien sehingga efektif mendukung operasi dan pengendalian organisasi. Data merupakan sumber daya penting pada manajemen modern. Untuk itu, organisasi perlu melakukan penataan dan manajemen data yang baik agar data yang dimiliki organisasi dapat berdaya guna secara maksimal.

Database adalah kumpulan data yang saling berhubungan yang merefleksikan fakta-fakta yang terdapat di organisasi. Saat satu kejadian muncul di dunia nyata mengubah *state* organisasi/perusahaan/sistem maka satu perubahan pun harus dilakukan terhadap data yang disimpan di *database*. *Database* merupakan komponen utama sistem informasi karena semua informasi untuk pengambilan keputusan berasal dari data di *database*. Pengelolaan *database* yang buruk dapat mengakibatkan ketidakterediaan data penting yang digunakan untuk menghasilkan informasi yang diperlukan dalam pengambilan keputusan.

Sistem manajemen basisdata atau DBMS (*Database Management System*) adalah perangkat lunak untuk mendefinisikan, menciptakan, mengelola, dan mengendalikan pelaksanaan basisdata. Fungsi sistem manajemen basisdata saat ini yang penting adalah menyediakan basis untuk sistem informasi manajemen.

2.5 Software yang Digunakan dalam Pembuatan Aplikasi

2.5.1 HTML

HTML kependekan dari *Hyper Text Markup Language*. Dokumen HTML adalah file teks murni yang dapat di buat dengan editor teks sembarang. Dokumen ini dikenal sebagai web page. Dokumen HTML merupakan dokumen yang disajikan dalam browser web surfer. Dokumen ini umumnya berisi informasi atau interface aplikasi di dalam internet. Ada dua cara untuk membuat sebuah web page, dengan HTML editor atau dengan editor teks biasa (misalnya notepad).

Dokumen HTML disusun oleh elemen-elemen. “Elemen” merupakan istilah bagi komponen-komponen dasar pembentuk dokumen HTML. Beberapa contoh elemen adalah head, body, table, paragraf, dan list. Elemen dapat berupa teks murni, atau bukan teks, atau keduanya.(Abdul Kadir, 2008)

2.5.2 CSS

CSS merupakan singkatan dari *Cascading Style Sheet*, merupakan fitur baru dari HTML 4.0. hal ini diperlukan setelah melihat perkembangan HTML menjadi kurang praktis karena web pages terlalu banyak dibebani hal-hal yang berkaitan dengan faktor tampilan seperti font dan lain-lain. Untuk itu jika kumpulan isi gaya (*style*) tersebut dikelola secara terpisah maka manajemen pages menjadi lebih mudah dan efisien. (Bambang Hariyanto, 2008)

2.5.3 PHP

Menurut Budi Raharjo (2011), PHP merupakan salah satu bahasa pemrograman berbentuk skrip yang ditempatkan dalam server dan diproses di server. Hasilnyalah yang dikirim ke klien, tempat pemakai menggunakan *browser*. Secara khusus, PHP dirancang untuk membentuk aplikasi web dinamis. Maksudnya, PHP mampu menghasilkan *website* yang secara terus-menerus hasilnya bisa berubah-ubah sesuai dengan pola yang diberikan. Hal tersebut tergantung pada permintaan *client browser*-nya (bisa menggunakan *browser* Opera, Internet Explorer, Mozilla, dan lain-lain). Umumnya, pembuatan web dinamis berhubungan erat dengan *Database* sebagai sumber data.

PHP mempunyai fungsi yang sama dengan skrip-skrip seperti ASP (*Active Server Page*), Cold Fusion, ataupun Perl. Namun, Perlu diketahui bahwa PHP sebenarnya bisa dipakai secara *Command Line*. Artinya, skrip PHP dapat dijalankan tanpa melibatkan *web server* maupun *browser*.

Model kerja HTML diawali dengan permintaan suatu halaman web oleh *browser*. Berdasarkan URL (*Uniform Resource Locator*) atau dikenal dengan sebutan alamat internet, *browser* mendapatkan alamat dari *web browser*, mengidentifikasi alamat 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* (atau yang biasa disebut browser saja). *Browser* yang mendapatkan isinya segera melakukan proses penerjemah kode HTML dan menampilkan ke layar pemakai.

Bagaimana halnya yang diminta adalah sebuah halaman PHP? Prinsipnya serupa dengan kode HTML. Hanya saja, ketika berkas PHP yang diminta didapatkan oleh web server, isinya segera dikirimkan ke mesin PHP dan mesin inilah yang memproses dan memberikan hasilnya (berupa kode HTML) ke *Web Server*. Selanjutnya, *web server* menyampaikan ke klien.

2.5.4 Java Script

JavaScript adalah kode untuk menyusun halaman web yang memungkinkan dijalankan di sisi klien (pada browser yang digunakan pemakai). Karena dijalankan di sisi klien, maka JavaScript digunakan untuk membuat tampilan lebih dinamis. (Abdul Kadir, 2011).

2.5.5 JQuery

Jquery merupakan salah satu pustaka yang dikembangkan dengan mengembangkan JavaScript. Kehadirannya adalah untuk memudahkan penulisan kode JavaScript. Dengan menggunakan JQuery, penulisan kode JavaScript menjadi lebih sederhana (lebih ringkas). Selain itu, yang lebih penting lagi, pembuatan web yang interaktif dan menarik menjadi jauh lebih mudah diimplementasikan daripada jika Anda menggunakan JavaScript sendiri. (Abdul Kadir, 2011).

2.5.6 MYSQL

MySQL adalah salah satu jenis *database server* yang sangat terkenal. Populerannya disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk

mengakses *databasenya*. Selain itu ia bersifat *open source* (tidak perlu membayar untuk menggunakannya) pada pelbagai platform (kecuali untuk jenis Enterprise, yang bersifat komersial).

MySQL termasuk jenis RDBMS (*Relational Database Management System*). Itulah sebabnya, istilah seperti tabel, baris, dan kolom digunakan pada MySQL. Pada MySQL, sebuah *database* mengandung satu atau beberapa kolom.

2.6 UML

UML itu singkatan dari *Unified Modelling Language*. Sesuai dengan kata terakhir dari kepanjangannya, UML merupakan salah satu bentuk language atau bahasa. Menurut pencetusnya, UML didefinisikan sebagai bahasa visual untuk menjelaskan, memberikan spesifikasi, merancang, membuat model, dan mendokumentasikan aspek-aspek dari sebuah sistem.

Karena tergolong bahasa visual, UML lebih mengedepankan penggunaan diagram untuk menggambarkan aspek dari sistem yang sedang dimodelkan. Memahami UML itu sebagai bahasa visual itu penting, karena penekanan tersebut membedakannya dengan bahasa pemrograman yang lebih dekat ke mesin. Bahasa visual lebih dekat ke mental model pikiran kita, sehingga pemodelan menggunakan bahasa visual bisa lebih mudah dan lebih cepat dipahami dibandingkan apabila dituliskan dalam sebuah bahasa pemrograman.

UML adalah salah satu bentuk notasi atau bahasa yang sama yang digunakan oleh professional dibidang software untuk menggambarkan atau memodelkan sebuah system software. Sebelumnya ada banyak notasi atau bahasa lain untuk mencapai

keperluan yang sama misalnya DFD (Data Flow Diagram). Tetapi sejak matang dan populernya teknologi pemrograman, perancangan, dan analisis berorientasi objek, UML telah menjadi de facto standard language.

Ada tiga cara dalam memakai UML dalam melakukan pemodelan system:

1. UML sebagai ketsa

UML digambarkan dalam sketsa coretan-coretan dalam kertas atau whitboard secara tidak formal. Biasanya digunakan dalam sesi diskusi tim untuk membahas aspek tertentu dalam tahap analisis dan perancangan.

2. UML sebagai blueprint system

Seperti diagram kelistrikan adalah blueprint dari komponen atau produk yang akan dihasilkan, UML juga bisa menggambarkan blueprint yang identik untuk sebuah system software.

3. UML sebagai bahasa pemrograman

UML berfungsi sebagai bahasa pemrograman mencoba melakukan semuanya dengan UML sampai kepada produk jadinya. Analisis dan perancangan dilakukan dengan diagram-diagram yang ada dalam UML, sementara sebuah tool atau generator bisa menghasilkan produk akhir dari diagram-diagram ini. Diagram-diagram yang terdapat dalam UML antara lain:

- *use case diagram*
- *class diagram*
- *statechart diagram*
- *activity diagram*
- *sequence diagram*

- *collaboration diagram*
- *component diagram*
- *deployment diagram*

(Munawar, 2005).

2.6.1 Use Case

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya.

Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

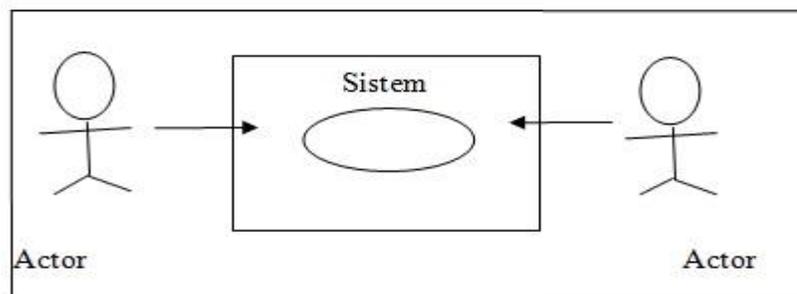
Use case diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem.

Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal.

Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*.

Sebuah *use case* juga dapat meng-*extenduse case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

Diagram use case menunjukkan 3 aspek dari system yaitu Actor, use case dan system/sub system boundary. Actor mewakili peran orang, system yang lain atau alat ketika berkomunikasi dengan use case.

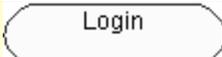
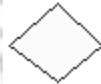


Gambar 2.5 Use Case Model (Munawar, 2005)

2.6.2 Activity Diagram

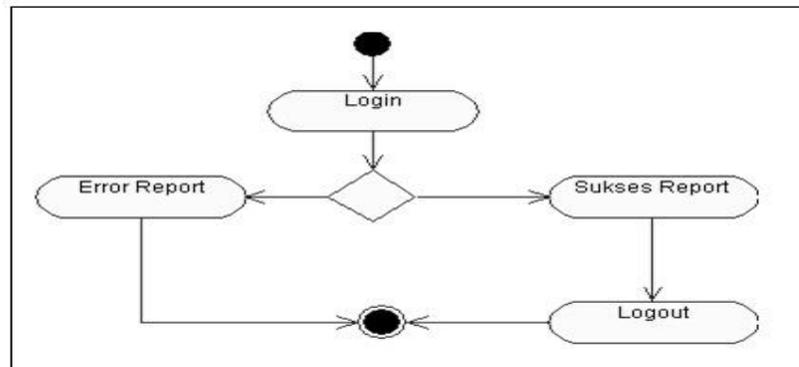
Menurut (Romi Satria Wahono, 2003) Pada dasarnya Diagram aktivitas adalah Diagram *flowchart* yang diperluas yang menunjukkan aliran kendali satu aktivitas ke aktivitas lain. Kegunaan diagram ini adalah untuk memodelkan workflow atau jalur kerja, memodelkan operasi, bagaimana objek-objek bekerja, aksi-aksi dan pengaruh terhadap objek. Simbol-simbol yang terdapat dalam *Activity Diagram*, diantaranya sebagai berikut :

Tabel 2.1 Simbol Activity Diagram (Romi Satrio Wahono, 2003)

Keterangan	Simbol
Titik Awal atau permulaan.	
Titik Akhir atau akhir dari aktivitas.	
Aktiviti, atau aktivitas yang dilakukan oleh aktor.	
Decision, atau pilihan untuk mengambil keputusan.	
Arah tanda panah alur proses.	

Activity diagram menunjukkan apa yang terjadi, tetapi tidak menunjukkan siapa yang melakukan apa. Dalam pemrograman hal tersebut tidak menunjukkan *class* mana yang bertanggungjawab atas setiap *action*. Pada pemodelan bisnis, hal tersebut tidak bisa menunjukkan organisasi mana yang menjalankan sebuah *action*. *Swimlane* adalah sebuah cara untuk mengelompokan *activity* berdasarkan *actor* (mengelompokkan *activity* dari sebuah urutan yang sama). *Actor* bisa ditulis nama *actor* ataupun sekaligus dengan lambang *actor* (*stick figure*) pada *usecase diagram*. *Swimlane* digambarkan secara vertikal, walaupun terkadang digambarkan secara horizontal.

Activity diagram merupakan salah satu diagram yang umum digunakan dalam UML untuk menjabarkan proses atau aktivitas dari aktor. Sebagai contoh, pelanggan melakukan *login* (masuk) pada halaman *website* untuk bergabung, jika pelanggan belum terdaftar, maka akan ditolak oleh sistem dan dikembalikan. Proses penjabarannya adalah sebagai berikut :

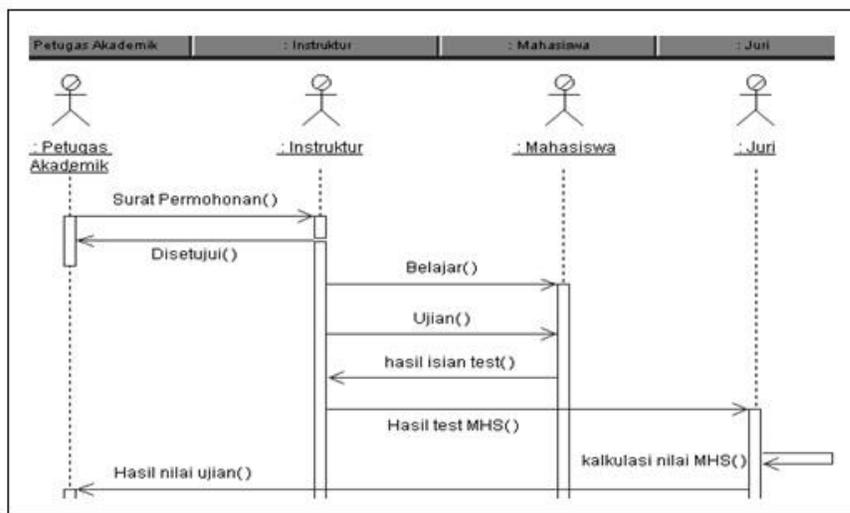


Gambar 2.6 Activity Diagram (Romi Satrio Wahono, 2003)

Di dalam *Activity* diagram tersebut dijelaskan bahwa *user* melakukan proses *login* untuk dapat memasuki area sistem, jika proses *login* dan/atau *user* belum teregistrasi, maka *user* akan ditolak oleh sistem tersebut dan diberi pesan *error*. Selain itu, bila *user* telah teregistrasi dan memasukkan kode *login* dengan benar maka akan diberi akses untuk masuk ke sistem, dan diberikan pesan sukses. *User* dapat *logout* (keluar) untuk mengakhiri sesi.

2.6.3 Sequence Diagram

Sequence diagram menjelaskan secara detail urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case* : interaksi yang terjadi antar class, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi Berikut contoh sederhana *Sequence* diagram pada gambar 2.7.



Gambar 2.7 Contoh *Sequence* Diagram (Sri Dharwiyanti, 2003)

