

BAB II

TINJAUAN PUSTAKA

II.1. Sistem Pakar

Sistem pakar adalah aplikasi berbasis komputer yang digunakan untuk menyelesaikan masalah sebagaimana yang dipikirkan oleh pakar. Pakar yang dimaksud di sini 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 mendiagnosis penyakit yang diderita pasien serta dapat memberikan penatalaksanaan terhadap penyakit tersebut. Tidak semua orang dapat mengambil keputusan mengenai diagnosis dan memberikan penatalaksanaan suatu penyakit. Contoh yang lain, monitor adalah seorang yang punya keahlian dan pengalaman dalam menyelesaikan kerusakan mesin motor/mobil, psikolog adalah orang yang ahli dalam memahami kepribadian seseorang, dan lain-lain (kusrini ; 2010 : 3).

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.

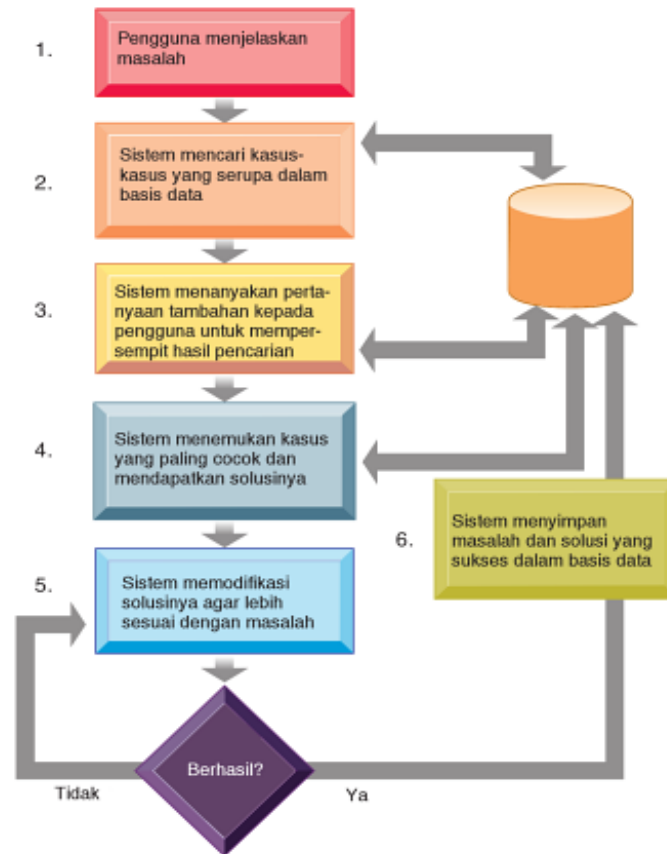
Sebuah sistem pakar memiliki 2 komponen utama yaitu basis pengetahuan dan mesin inferensi. Basis pengetahuan merupakan tempat penyimpanan pengetahuan dalam memori komputer, di mana pengetahuan ini diambil dari pengetahuan kaidah produksi.

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

II.2. Sistem Logika Fuzzy

Kebanyakan orang tidak berpikir dengan menggunakan aturan IF THEN yang tradisional atau dengan angka-angka yang tepat. Manusia cenderung mengategorikan hal-hal secara tidak tepat, dengan menggunakan aturan pengambilan keputusan yang mungkin memiliki banyak perbedaan pengertian. Sebagai contoh, seorang laki-laki atau perempuan dapat saja kuat atau cerdas. Sebuah perusahaan dapat digolongkan sebagai besar, sedang, atau kecil. Suatu temperature dapat digolongkan sebagai panas, sejuk, dingin, atau hangat. Kategori-kategori ini merepresentasikan suatu rentang nilai. (Kennec, dkk ; 2010 ; 124).

Logika fuzzy (*fuzzy logic*) adalah teknologi berbasis aturan yang dapat merepresentasikan ketidak presisian seperti yang telah disebutkan, dengan menciptakan aturan yang menggunakan nilai subjektif atau nilai yang mendekati. *Logika fuzzy* dapat menjelaskan fenomena atau proses tertentu secara linguistic, kemudian merepresentasikannya dalam sejumlah kecil aturan yang fleksibel. Organisasi dapat menggunakan *logika fuzzy* untuk menciptakan sistem piranti lunak yang menangkap pengetahuan tersirat yang mengandung *ambiguitas linguistic*. (Kennec, dkk ; 2010 ; 125).



Gambar II.1. Cara kerja penalaran berbasis kasus
 (Sumber : Sistem Informasi Manajemen, Edisi 10, Kennech C, dkk ; 2010)

Lihat bagaimana *logika fuzzy* merepresentasikan temperature yang bervariasi dalam aplikasi computer untuk mengendalikan temperature ruangan secara otomatis. Istilah-istilah atau dikenal sebagai fungsi keanggotaan didefinisikan secara tidak presisi, sejuk adalah antara 50 derajat dan 70 derajat (*Fahrenheit*), padahal temperature sejuk lebih tepatnya berkisar antara 60 dan 67 derajat. Ingatlah bahwa kategori sejuk bertumpang tindih dengan kategori dingin atau normal. Untuk mengendalikan lingkungan ruangan menggunakan logika fuzzy, programmer juga perlu mengembangkan definisi ketidakpresisian yang

serupa untuk kelembapan dan faktor-faktor lainnya, seperti misalnya angin dan temperature di luar ruangan.

Aturan-aturan tersebut mungkin mencakup satu pernyataan berikut jika temperature sejuk atau dingin dan kelembapan rendah, sementara angin di luar ruangan tinggi dan temperature luar ruangan rendah, naikkan panas dan kelembapan dalam ruangan. Komputer akan mengombinasikan pembacaan fungsi keanggotan dengan melakukan pembobotan dan, menggunakan semua aturannya, menaikkan atau menurunkan temperature dan kelembapan.

Logika fuzzy menyediakan solusi bagi masalah-masalah yang sulit dipecahkan hanya dengan menggunakan aturan IF THEN. Di Jepang, sistem kereta bawah tanah sendai menggunakan control *logika fuzzy* untuk mengatur percepatan kereta dengan mulus sehingga para penumpang yang berdiri tidak perlu perpegangan saat kereta itu berjalan. Mitsubishi Heavy Industries di Tokyo telah berhasil mengurangi pemakaian listrik untuk penyejuk ruangan sampai 20 persen dengan mengimplementasikan program-program control *logika fuzzy*. Perangkat focus otomatis pada kamera hanya mungkin dibuat dengan menggunakan *logika fuzzy*. Pada contoh tersebut, *logika fuzzy* telah memungkinkan perubahan setahap demi setahap pada masukan menghasilkan perubahan yang mulus pada keluaran, dan bukan keluaran yang putus-putus. Hal tersebut membuat *logika fuzzy* sangat bermanfaat bagi konsumen aplikasi elektronik dan rekaya. (Kennec, dkk ; 2010 : 126).

Logika fuzzy juga dapat bermanfaat di bidang manajemen untuk pengambilan keputusan dan control organisasi. Sebuah perusahaan di Wall Street

menciptakan sistem yang memilih perusahaan-perusahaan yang berpotensi untuk diakuisisi dengan menggunakan bahasa yang dimengerti oleh para pedagang saham. Sistem logika fuzzy juga telah dikembangkan untuk mendeteksi kemungkinan kebohongan dalam klaim kesehatan yang diajukan oleh penyedia perawatan kesehatan di Amerika Serikat. (Kennec, dkk ; 2010 : 127).

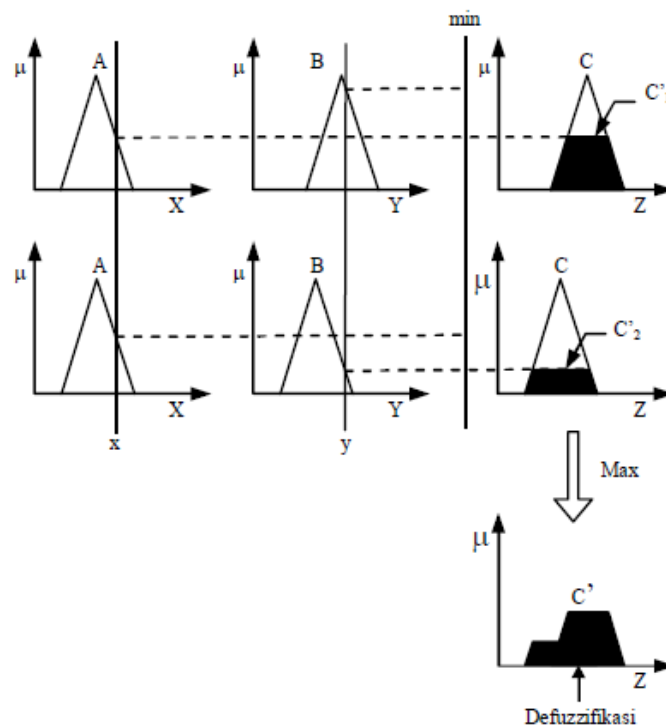
II.2.1. Logika Pengambilan Keputusan

Logika pengambilan keputusan atau dapat disebut penyimpulan fuzzy (*fuzzy inference*) mengaplikasikan aturan-aturan fuzzy pada masukan fuzzy kemudian mengevaluasi setiap aturan. Prinsip logika fuzzy digunakan untuk mengkombinasi aturan-aturan JIKA-MAKA (IFTHEN) yang terdapat dalam basis aturan kedalam suatu pemetaan dari suatu himpunan *fuzzy input* kesuatu himpunan *fuzzy output*. Logika pengambilan keputusan merupakan langkah kedua dalam pemrosesan logika fuzzy. Terdapat beberapa metode pengambilan keputusan dalam logika fuzzy diantaranya yaitu metode Mamdani. (Sutikno ; 2010 : 2).

Fungsi implikasi yang digunakan pada pengambilan keputusan dengan metode Mamdani dengan menggunakan MIN dan dalam melakukan komposisi dengan menggunakan MAX. Metode komposisi ini sering disebut MAX-MIN. Contoh dalam penggunaan pengambilan keputusan dengan metode Mamdani ditunjukkan pada gambar II.2. Dengan memisalkan fungsi keanggotaan masukan dan keluaran menggunakan fungsi segitiga dan mempunyai 2 aturan fuzzy, yaitu:

- a. IF Kesalahan adalah Nol dan Beda kesalahan adalah Positif maka Keluaran adalah Positif.

- b. IF Kesalahan adalah Nol dan Beda kesalahan adalah Nol maka Keluaran adalah Nol.



Gambar II.2. Proses pengambilan keputusan metode Mamdani
(Sumber : Sutikno ; 2010 : 35)

Langkah pertama pengambilan keputusan metode Mamdani adalah melakukan proses fuzzifikasi untuk memetakan data tegas masukan kesalahan dan beda kesalahan kedalam data fuzzy sesuai dengan tipe dan bentuk fungsi keanggotaan. Langkah kedua adalah melakukan proses terhadap kedua data fuzzy tersebut dengan operator AND yang akan mengambil nilai paling minimal dari dua data tersebut. Langkah ketiga dengan implikasi MIN akan memotong fungsi keanggotaan keluaran setelah melalui operator AND sehingga didapatkan daerah fuzzy. Ketiga proses tersebut juga diterapkan pada aturan-aturan fuzzy berikutnya. Setelah semua aturan fuzzy telah dieksekusi, dilakukan proses komposisi dengan

metode MAX yaitu solusi himpunan fuzzy diperoleh dengan cara mengambil nilai maksimum aturan, kemudian menggunakannya untuk memodifikasi daerah fuzzy, dan mengaplikasikan ke *output* dengan menggunakan operator OR (*union*). Jika proposisi telah dievaluasi, maka *output* akan berisi suatu himpunan fuzzy yang merefleksikan kontribusi dari tiap-tiap proposisi. Setelah proses implikasi dan komposisi telah dilakukan maka proses selanjutnya adalah proses defuzzifikasi. (Sutikno ; 2010 : 3).

II.2.2. Metode Sugeno

Penalaran dengan metode *output* (konsekuen) sistem tidak berupa himpunan *fuzzy*, melainkan berupa konstanta atau persamaan *linear*. Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985. *Michio Sugeno* mengusulkan penggunaan *singleton* sebagai fungsi keanggotaan dari konsekuen. *Singleton* adalah sebuah himpunan *fuzzy* dengan fungsi keanggotaan pada titik tertentu mempunyai sebuah nilai dan 0 di luar titik tersebut. Pada metode *Sugeno* dua bagian pertama dari proses penarikan kesimpulan *fuzzy*, *fuzzifikasi input* dan menerapkan operator *fuzzy* semua sama dengan metode *Mamdani*. Perbedaan utama antara metode *Mamdani* dan *Sugeno* adalah *output membership function* dari metode *Sugeno* berbentuk *linier* atau konstan (Mustaziri, *eprints.undip.ac.id* ; 2012 : 32).

Aturan pada model *fuzzy* Sugeno mempunyai bentuk :

If Input 1 = x and Input 2 = y

then Output is $z = ax + by + c$

Untuk model *Sugeno* orde-Nol, *Output* level z adalah konstan ($a=b=0$). *Output* level z_i dari setiap aturan merupakan berat dari aturan w_i (*firing strength*). Sebagai contoh, untuk aturan AND dengan *Input* 1 = x dan *Input* 2 = y , maka *firing strength* adalah : $w_i = \text{AndMethod}(F1(X), F2(Y))$ dimana $F1,2(.)$ adalah *membership function* untuk *Input* 1 dan 2. Keuntungan metode *Sugeno* :

1. Komputasinya lebih efisien .
2. Bekerja paling baik untuk teknik linear.
3. Bekerja paling baik untuk teknik optimasi dan adaptif .
4. Menjamin kontinuitas permukaan *output* .
5. Lebih cocok untuk analisis secara matematis.

Ada 2 model *fuzzy* dengan metode *Sugeno* yaitu sebagai berikut:

1. Model *Fuzzy Sugeno* Orde-Nol

Secara umum bentuk model *fuzzy SUGENO* Orde-Nol adalah:

IF (x_1 is A_1) • (x_2 is A_2) • (x_3 is A_3) • • (x_N is A_N) THEN $z=k$ dengan A_i adalah himpunan *fuzzy* ke- i sebagai anteseden, dan k adalah suatu konstanta (tegas) sebagai konsekuen.

2. Model *Fuzzy Sugeno* Orde-Satu

Secara umum bentuk model *fuzzy SUGENO* Orde-Satu adalah:

IF (x_1 is A_1) • • (x_N is A_N) THEN $z = p_1*x_1 + \dots + p_N*x_N + q$.

Dengan A_i adalah himpunan *fuzzy* ke- i sebagai anteseden, dan p_i adalah suatu konstanta (tegas) ke- i dan q juga merupakan konstanta dalam konsekuen. Apabila komposisi aturan menggunakan metode *SUGENO*, maka defuzzifikasi dilakukan dengan cara mencari nilai rata-ratanya (Mustaziri, *eprints.undip.ac.id* ; 2012 : 33).

Sistem *fuzzy Sugeno* memperbaiki kelemahan yang dimiliki oleh sistem *fuzzy* murni untuk menambah suatu perhitungan matematika sederhana sebagai bagian THEN. Pada perubahan ini, sistem *fuzzy* memiliki suatu nilai rata-rata tertimbang (*Weighted Average Values*) di dalam bagian aturan *fuzzy IF-THEN*.

Sistem *fuzzy Sugeno* juga memiliki kelemahan terutama pada bagian *THEN*, yaitu dengan adanya perhitungan matematika sehingga tidak dapat menyediakan kerangka alami untuk merepresentasikan pengetahuan manusia dengan sebenarnya. Permasalahan kedua adalah tidak adanya kebebasan untuk menggunakan prinsip yang berbeda dalam logika *fuzzy*, sehingga ketidakpastian dari sistem *fuzzy* tidak dapat direpresentasikan secara baik dalam kerangka ini (Mustaziri ; 2012 : 32).

II.3. Data, Informasi, dan Pengetahuan

Data merupakan representasi fakta mengenai suatu objek atau kejadian.
(kusrini;2008:4) 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

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).

Tabel II.1. Data Pelanggan

Nama	Alamat	Jenis Kelamin
Paijo	Sukoharjo	Laki-Laki
Imin	Bantul	Laki-Laki
Tentrem	Turi	Perempuan

(Sumber : Kusri ; 2010 : 4)

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 di antaranya :

1. Orang yang beralamat Sukoharjo suka mebel dengan warna-warni 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 :
 - a) Pengetahuan tentang bagaimana mencuci dengan menggunakan mesin.
 - b) Pengetahuan tentang bagaimana membuat puding.
 - c) Pengetahuan tentang bagaimana cara mengobati luka bakar.
2. 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.
3. Pengetahuan tacit (*tacit knowledge*) pengetahuan yang tidak bisa diungkapkan dengan bahasa
 - a) Bagaimana cara mengayuh sepeda
 - b) Bagaimana cara berjinjit untuk mencari balet

Pengetahuan bisa dimasukkan secara manual, semi otomatis maupun otomatis. Secara manual pengetahuan dapat diperoleh dari hasil wawancara, pelacakan proses penalaran ataupun observasi. Secara semiotomatis pengetahuan dimasukkan dengan sedikit bantuan dari *knowledge engineer*, tetapi sumbernya masih dari pakar, sedangkan cara otomatis dapat dilakukan dengan minimal input dari *knowledge engineer* maupun dari pakar.

II.4. Representasi Pengetahuan

Representasi pengetahuan merupakan metode yang digunakan untuk mengkodekan pengetahuan dalam sebuah sistem pakar. Representasi dimaksudkan untuk menangkap sifat-sifat penting masalah dan membuat informasi itu dapat diakses oleh prosedur pemecahan masalah (Kusrini ; 2010 : 6).

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 sehingga 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 sistem 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.

II.5. Pengertian Jaringan

Jaringan komputer adalah sistem yang terdiri dari komputer-komputer, serta piranti-piranti yang saling terhubung sebagai satu kesatuan. Dengan

dihubungkan pirang-piranti tersebut, alhasil dapat saling berbagi sumber daya antar satu piranti dengan piranti lainnya (Wahana Komputer ; 2010 ; 2).

Dalam istilah komputer, jaringan merupakan penghubung antara dua komputer atau lebih yang tujuan utamanya adalah berbagi data. Betulkah jaringan komputer itu *hardware* dan *software*? Jawabnya adalah betul. Jaringan komputer adalah gabungan antara *hardware* dan *software*.

Jaringan komputer bisa diklasifikasikan menurut beberapa kategori, nanti akan dijelaskan lebih lanjut di sub bab mengenal jenis jaringan komputer. Karena sebuah sistem, jaringan komputer terdiri atas komponen-komponen, dan perangkat jaringan lainnya yang bekerja bersama-sama untuk mencapai suatu tujuan yang sama. Tujuan dari jaringan komputer adalah :

1. Membagi sumber daya, contohnya berbagi pemakaian printer CPU (*central processing unit*), memori, hard disk
2. Komunikasi, contohnya suatu elektronik, instant messaging, chatting.
3. Akses informasi, contohnya *web browsing*.

Agar dapat mencapai tujuan yang sama, setiap bagian dari jaringan komputer meminta dan memberikan layanan (*service*). Pihak yang meminta layanan disebut *client* dan yang memberikan layanan disebut *server*. Arsitektur ini disebut dengan sistem *client server*, dan digunakan pada jaringan komputer.

II.5.1. Jenis Jaringan

Jaringan komputer mempunyai berbagai macam tipe yang masing-masing memiliki kelebihan dan kekurangan. Agar lebih jelas, berikut penjabarannya (Wahana Komputer ; 2010 ; 3).

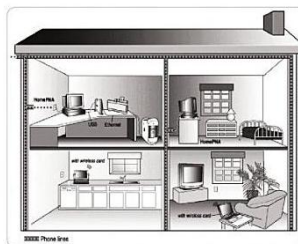
1. Berdasarkan Ruang Lingkup

Ada banyak tipe jaringan komputer, itu juga bisa dibedakan berdasarkan beberapa parameter yang berbeda. Parameter pertama adalah berdasarkan ruang lingkup. Ada beberapa tipe jaringan komputer berdasarkan ruang lingkungnya, yaitu :

- 1) *Local Area Network* (LAN)
- 2) *Metropolitan Area Network* (MAN)
- 3) *Wide Area Network* (WAN)

a. LAN (*Local Area Network*)

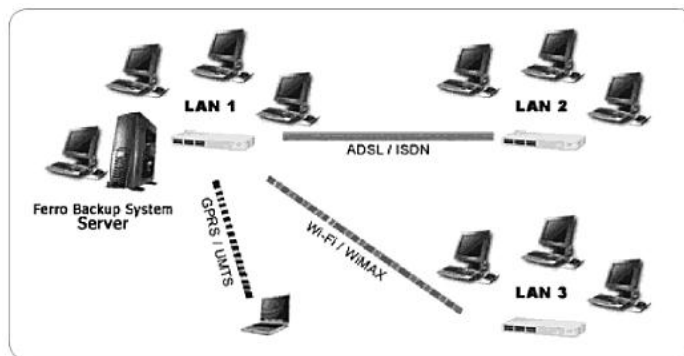
LAN (*Local Area Network*) adalah sebuah jaringan komputer yang cakupan areanya kecil, seperti di sebuah rumah, kantor, atau sekolah. Karakteristik khusus dari LAN yang membedakan dengan jaringan WAN adalah transfer data yang lebih besar, cakupan atau geografis yang lebih sempit dan tidak perlunya jalur komunikasi *leased line*.



Gambar II.3. Jaringan LAN Didalam Rumah
(Sumber : *Wahana Komputer ; 2010 ; 3*)

b. WAN (*Wide Area Network*)

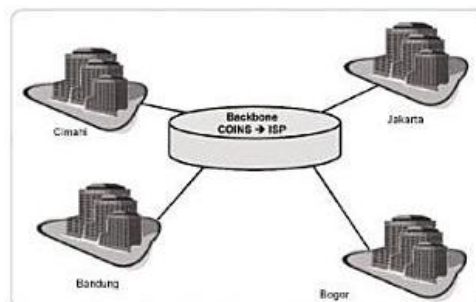
Sebuah *Wide Area Network* (WAN) adalah jaringan komputer yang cakupannya cukup luas, seperti antar regional atau antar negara. Dapat dilihat pada Gambar II.4.



Gambar II.4. Jaringan WAN
(Sumber : Wahana Komputer ; 2010 ; 4)

c. MAN (*Metropolitan Area Network*)

Metropolitan Area Network (MAN) adalah jaringan komputer yang cakupan luasnya mencapai satu atau lebih kota. Sebuah MAN biasanya menghubungkan antara beberapa LAN lokal menggunakan teknologi *backbone*, seperti *fiber optic*, dan menyediakan layanan ke banyak jaringan seperti untuk internet. Dapat dilihat pada Gambar II.5.



Gambar II.5. Jaringan MAN
(Sumber : Wahana Komputer ; 2010 ; 5)

II.6. Jaringan *Client Server*

Jaringan *Client Server* menghubungkan komputer server dengan komputer client/workstation. Komputer *Server* adalah komputer yang menyediakan fasilitas bagi komputer-komputer *client/workstation* yang

terhubung dalam jaringan. Sedangkan komputer adalah komputer yang menggunakan fasilitas yang disediakan oleh komputer server. Komputer *server* pada sebuah jaringan tipe *client server* disebut dengan *dedicated server*, karena komputer yang digunakan hanya sebagai penyedia fasilitas untuk komputer *client/workstation*.

II.7. Desain atau Perancangan Sistem

Desain atau perancangan dalam pengembangan perangkat lunak merupakan upaya untuk mengkonstruksi sebuah sistem yang memberikan kepuasan (mungkin informal) akan spesifikasi kebutuhan fungsional memenuhi target, memenuhi kebutuhan secara implisit dan eksplisit dari segi performansi maupun penggunaan sumber daya, kepuasan batasan pada proses desain dari segi biaya, waktu, dan perangkat. (Rosa A.S, M. Shalahuddin ; 2011 : 21).

II.8. Mengenal HTML

HTML memiliki beberapa kode yang memungkinkan programmer atau desainer mendesain halaman web yang tampilannya tidak seperti biasa, Anda akan belajar bagaimana membuat tabel, frame, memasukkan JavaScript, dan mengenal PHP.

II.8.1. Memahami Tabel di HTML

Tabel penting peranannya dalam halaman web, selain untuk menampilkan teks atau gambar dalam format lajur dan kolom, Anda bisa juga menggunakan tabel untuk membantu layout tampilan halaman.

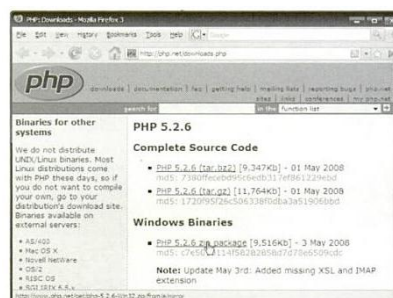
Tabel merupakan sebuah kotak yang terdiri dari baris (*row*) dan kolom (*column*). Untuk membuat tabel, Anda menggunakan tag `<table>` dan menutupnya dengan tag `</table>`. Anda bisa juga menambahkan atribut lain di tag `<table>` pembuka. Misalnya menentukan border, warna, dan sebagainya.

Di dalam tag `<table>` ada beberapa tag lain yang perlu Anda pahami, yaitu :

- Tag `<tr>`, artinya tag untuk menuliskan baris biasa di tabel, TR singkatan dari *table row*.
- Tag `<td>`. Artinya tag untuk menuliskan kotak di dalam baris, makanya tag `<td>` ada di dalam tag `<tr>`. TD singkatan dari *table data*.
- Tag `<th>` artinya tag untuk menuliskan kotak biasa seperti `<td>`, namun untuk header tabel. TH singkatan dari *table header*.

II.9. Mengenal Server Environment

Kelebihan PHP yang paling terasa adalah tersedianya PHP parser di banyak platform. Anda bisa menjalankan skrip PHP di banyak *server*, seperti Apache dan IIS dan di banyak sistem operasi. Untuk melihat halaman download PHP dapat dilihat pada Gambar II.6.

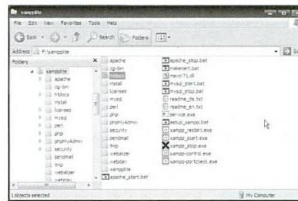


Gambar II.6. Halaman PHP
(Sumber : Ali Zaki ; 2010 ; 32)

II.9.1. Instalasi Server Environment

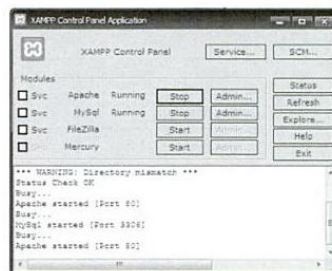
File *xampp Lite* yang sudah di *download* berupa file *7zip executable*. Eksekusi file tersebut dan kemudian tentukan tempat tujuan ekstraksi dengan mengisikannya di kota *Extract to*.

Xampplite akan langsung terekstrak ke tempat tujuan. Ada banyak file di dalam *folder xampplite*. *Folder* penting adalah *htdocs* di mana file-file halaman web harus diletakkan di situ. Yang kedua adalah file *xampp-control.exe* yang berguna untuk mengatur perilaku *server*, seperti mengaktifkan dan menonaktifkan komponen-komponen tertentu. Untuk melihat tempat folder terinstall xampplite dapat dilihat pada Gambar II.7.



Gambar II.7. Tempat Penyimpanan xampplite
(Sumber : Ali Zaki ; 2010 ; 35)

Agar bisa membuat kode PHP dieksekusi oleh server, Anda perlu mengaktifkan modul Apache dan MySQL dari *xampp-control.exe*. Klik tombol *Start* untuk mengaktifkan modul tersebut. Dapat dilihat pada Gambar II.8.



Gambar II.8. Mengaktifkan PHP dan MySQL
(Sumber : Ali Zaki ; 2010 ; 35)

II.10. Mengenal MySQL

MySQL merupakan *database server open source* yang cukup populer keberadaannya. Dengan berbagai keunggulan yang dimiliki, membuat software database ini banyak digunakan oleh para praktisi untuk membangun suatu project. Adanya fasilitas API (*Application Programming Interface*) yang dimiliki oleh MySQL, memungkinkan bermacam-macam aplikasi komputer yang ditulis dengan berbagai bahasa pemrograman dapat mengakses basis data MySQL (Wahana Komputer ; 2010 : 2).

II.10.1. Database

Database adalah sebuah struktur yang umumnya terbagi dalam 2 hal, yaitu sebuah *database flat* dan sebuah *database relasional*. *Database* relasional lebih mudah dipahami daripada *database flat* karena database relasional mempunyai bentuk yang sederhana serta mudah dilakukan operasi data. MySQL sendiri adalah sebuah database relasional. Database yang memiliki struktur relasional terdapat tabel-tabel untuk menyimpan data. Pada setiap tabel terdiri dari kolom dan baris serta sebuah kolom untuk mendefinisikan jenis informasi apa yang harus disimpan (Wahana Komputer ; 2010 : 2).

Mengapa menggunakan database, itu pertanyaan yang akan keluar dari pikiran Anda pada saat pertama kali ingin mempelajari database. Database akan menjadi sangat berguna saat Anda perlu menyimpan informasi yang dikategorikan secara logis. Contoh, jika Anda ingin menyimpan informasi tentang PT. Wahana Komputer dengan database, Anda bisa mengelompokkan berbagai hal dalam bisnis menjadi beberapa tabel.

II.10.2. Database Relasional

Ketika menggunakan *software* sistem manajemen *database* terkomputerisasi, umumnya menggunakan database relasional. Prinsip *database* relasional adalah informasi dibagi menjadi beberapa data yang terpisah secara logis. Data-data yang terpisah tersebut diletakkan dalam bentuk tabel. Tabel adalah objek dasar yang merupakan jantung dari database relasional. Tabel adalah dasar penyimpanan informasi dan pengambilannya (*retrieval*). Ketika informasi sudah tersimpan di tabel-tabel yang terpisah, Anda nantinya dapat melihat (*view*), mengedit (*edit*), menambah (*add*), dan menghapus (*delete*) informasi dengan berbagai metode. Selain itu, Anda juga bisa mengambil informasi menggunakan *query* dan menampilkan informasi menggunakan report.

Keunggulan penyimpanan data menggunakan database relasional sangat banyak dibandingkan dengan penyimpanan ke satu tabel ukuran besar dua dimensi seperti di dokumen Word atau *spreadsheet* Excel.

Salah satu keunggulan utama database adalah berkurangnya redudansi data. Konsekuensinya tidak hanya ruang penyimpanan hard disk menjadi berkurang, tetapi kecepatan pemrosesan data juga berkurang. Selain itu, ada kelebihan lainnya, yaitu :

1. Fleksibilitas

Jika data berubah, Anda dapat mengupdate nilainya hanya di satu tempat.

Dengan demikian, semua *query*, form, dan lainnya akan berubah secara konsekuen dengan nilai yang baru tersebut.

2. *Simple*

Model penyimpanan tabel yang merupakan dasar dari sistem relasional memang simple, dan merupakan metode penyimpanan data yang tak redundan. Tiap tabel didesain relasional untuk objek tunggal yang mengandung data, yang konsekuen terhadap aspek tertentu dari database, seperti pegawai, produk atau order.

3. *Power*

Menyimpan data dalam tabel-tabel yang terpisah lebih memudahkan adanya pengelompokkan, *searching*, dan pengambilan data menggunakan banyak cara yang tak terbatas.

4. Kemudahan manajemen

Dengan tabel yang kecil dan tidak kompleks, informasi jadi mudah dilacak dan diatur.

Misalnya, Anda menggunakan database untuk menyimpan data order dari pelanggan. Jika menggunakan tabel tunggal, ada banyak record yang dibuat untuk tiap order yang dilakukan pelanggan. Dengan demikian, tiap kali ada order, tiap kali pula sebuah informasi dibuat, walaupun pelanggannya sama. Karena itu, jika suatu saat ada informasi yang berubah misalnya informasi pelanggan, semua record yang mengandung informasi pelanggan harus diubah. Ini akan sangat merepotkan. Begitu pula informasi tentang pegawai pemroses order juga akan terus berulang (Wahana Komputer ; 2010 : 3).

Akan lebih efisien seandainya ada satu tabel untuk pelanggan dan satu tabel untuk pegawai. Kemudian, ada field identitas pelanggan yang ditambahkan

ke tabel pelanggan dan tabel order sehingga ada hubungan atau koneksi antar keduanya. Hubungan ini disebut relationalship. Begitu pula identitas pegawai bisa dihubungkan dengan tabel order menggunakan relationship. Secara lengkap, komponen utama database dijelaskan seperti berikut ini.

1. Tabel

Tabel adalah inti konsep database. Tujuannya adalah menyimpan informasi. Tabel satu dengan lainnya bisa dihubungkan. Satu database bisa mengandung banyak tabel dalam jumlah tak terbatas. Jumlah record dalam tabel juga umumnya tak dibatasi, batasnya hanya pada kapasitas disk yang digunakan. Tiap tabel memiliki beberapa kolom, kolom menentukan nilai-nilai apa yang bisa disimpan, misalnya, tabel yang menyimpan informasi produk menyimpan data seperti nama, harga dan berat produk.

Setiap baris dari tabel bisa mengandung nilai untuk kolom-kolom yang ditentukan di tabel. Selain itu, tiap kolom bisa ditentukan tipe data apa saja yang didukung. Tipe data menentukan jenis data yang bisa disimpan. Tipe data bisa dibatasi seperti jenis numerik atau tanggal, atau lain sebagainya.

2. Query

Anda dapat mengambil informasi tertentu yang disimpan di tabel atau multitabel menggunakan sebuah query. Untuk membuat query, Anda harus menentukan parameter-parameter dari informasi yang ingin dicari. Misalnya Anda ingin mengambil data pelanggan yang telah membeli barang tipe X selama 3 bulan yang lalu. Dengan demikian, Anda harus mengisikan variabel nama barang

dan periode. Informasi yang ditampilkan juga nantinya bisa diurutkan, di *filter* dan diatur cara menampilkannya.

Dengan menjalankan *query*, Anda bisa menampilkan data berupa nilai-nilai yang sesuai dengan kriteria.

II.11. UML (*Unified Modelling Language*)

Pemodelan perangkat lunak bekerja dengan cara yang cukup serupa layaknya seorang arsitek atau insinyur teknik sipil yang akan membuat sebuah bangunan / gedung berskala besar. Saat seorang arsitek atau insinyur teknik sipil akan membuat sebuah bangunan / gedung berskala besar, ia biasanya membuat denah-denah atau maket-maket yang menggambarkan bentuk jadi dari bangunan / gedung. Kita sebagai seorang perancang sistem perangkat lunak juga bertindak dengan cara yang serupa, hanya saja yang kita rancang bukan bangunan, melainkan sistem perangkat lunak. Menggambarkan komponen-komponen sistem perangkat lunak dalam bentuk-bentuk geometri tertentu misalnya untuk menggambarkan suatu kelas (*class*) dalam aplikasi, menggunakan antarkelas (*asosiasi*), menggunakan garis lurus (Adi Nugroho ; 2009 : 6).

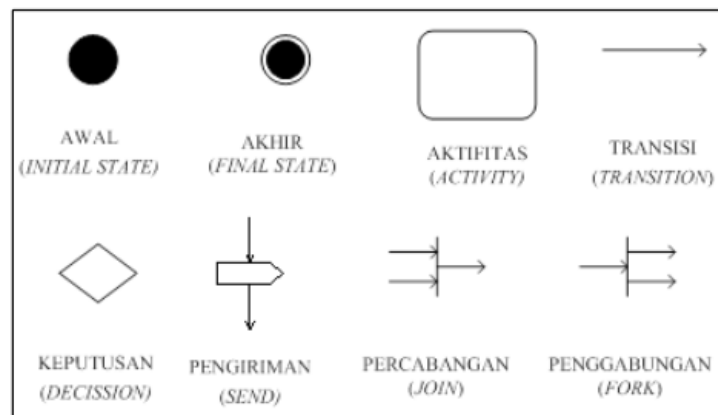
II.11.1. Use Case Diagram

Dalam konteks UML, tahap konseptualisasi dilakukan dengan pembuatan use case diagram yang sesungguhnya merupakan deskripsi peringkat tinggi bagaimana perangkat lunak (*aplikasi*) akan digunakan oleh penggunanya. Selanjutnya, use case diagram tidak hanya sangat penting pada tahap analisis, tetapi juga sangat penting untuk perancangan (*design*), untuk mencari (*mencoba*

menemukan) kelas-kelas yang terlibat dalam aplikasi, dan untuk melakukan pengujian (testing) (Adi Nugroho ; 2009 : 7).

II.11.2. Diagram Activity

Activity diagram menggambarkan urutan aktifitas yang digunakan untuk menjelaskan aktifitas dari sebuah operasi. Pada activity diagram terdapat keadaan aksi yang berisi spesifikasi dari aktifitas tertentu. Diagram ini berisi, pilihan keputusan dan kondisi serta spesifikasi message yang dikirim atau diterima sebagai gambaran dari aksi.



Gambar II.9. Diagram Activity

(Sumber : MHD Imam Alfarisyi, dkk ; 2011 : 56)

II.11.3. Sequence

Diagram interaksi yang menekankan pada waktu pengiriman *message*. *Sequence* diagram menunjukkan sekumpulan objek dan pengiriman serta penerimaan *message* antar objek. Objek yang umumnya memiliki nama atau instansi dari *class*, tapi dapat pula merupakan turunan dari *things* lain, seperti *collaboration*, *component* dan *node*. Diagram ini digunakan untuk mengilustrasikan *dynamic view* dari sistem.

II.11.4. State

Sebuah *state* diagram menggambarkan keadaan mesin, transisi, *event* dan *activity*. Diagram ini adalah pelengkap khusus untuk mendeskripsikan sebuah *class* yang menggambarkan *state* dari objek dari *class* dan *event* yang menyebabkan *state* berubah. *Event* tersebut dapat berasal dari objek yang mengirimkan suatu *message* atau dari kondisi yang terpenuhi.