

BAB II

LANDASAN TEORI

2.1. Sistem Pakar

2.1.1. Definisi Sistem Pakar

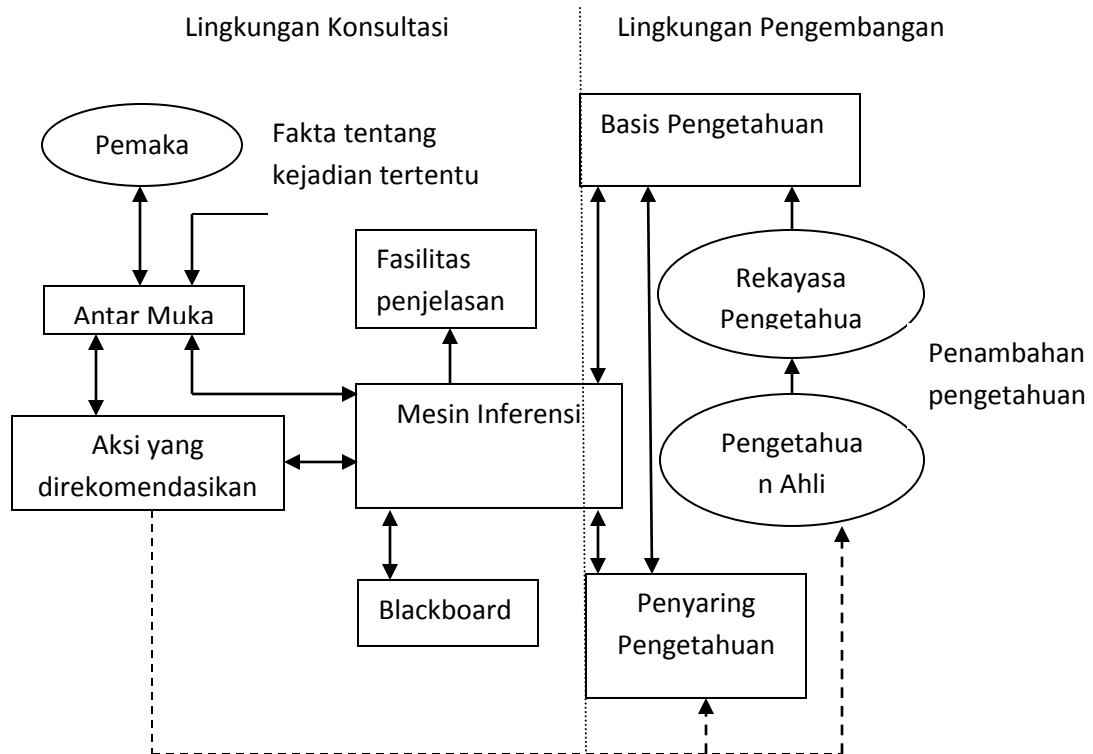
Sistem pakar merupakan cabang dari kecerdasan buatan, yang bertujuan untuk mengambil pengalaman para pakar dan untuk diimplementasikan ke sistem komputer. Pengetahuan khusus disimpan dalam komputer, oleh sistem dieksekusi yang menghasilkan kesimpulan yang spesifik dari permasalahannya. Tujuan dari sistem pakar adalah untuk membantu dan mendukung penilaian pengguna dan bukan menggantikan penilaian manusia. Bahkan, sistem pakar menawarkan solusi kepada pengguna ketika para pakar tidak ada (Vizureanu, 2010).

Sistem pakar adalah produk dari kecerdasan buatan, cabang ilmu komputer yang berupaya mengembangkan program cerdas. Sistem pakar daerah penerapan dan pemecahan masalah yang berbeda dalam berbagai bidang arsitektur, arkeologi, perdagangan, pendidikan, produksi barang dan diagnosis.

2.1.2. Struktur Sistem Pakar

Sistem pakar terdiri dari 2 bagian pokok yaitu : lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembangan digunakan

sebagai pembangun sistem pakar baik dari segi pembangun komponen maupun basis pengetahuan. Lingkungan konsultasi digunakan oleh seseorang yang bukan ahli untuk berkonsultasi. (Sutojo, Mulyanto, Suhartono, 2010). Berikut bagan komponen sistem pakar :



Gambar 2 Bagan Komponen Sistem Pakar

Komponen-komponen yang ada pada sistem pakar adalah sebagai berikut

Keterangan :

1. Subsistem penambahan pengetahuan. Bagian ini digunakan untuk memasukkan pengetahuan, mengkonstruksi atau memperluas pengetahuan dalam basis pengetahuan. Pengetahuan itu bisa berasal dari ahli, buku, basisdata, penelitian dan gambar.

2. Basis pengetahuan. Berisi pengetahuan-pengetahuan yang dibutuhkan untuk memahami, memformulasikan dan menyelesaikan masalah.
3. Motor inferensi (*inference engine*). Program yang berisi metodologi yang digunakan untuk melakukan penalaran terhadap informasi-informasi dalam basis pengetahuan dan blackboard, serta digunakan untuk memformulasikan konklusi. Ada 3 elemen utama dalam motor inferensi yaitu :
 - *Interpreter* : mengeksekusi item-item agenda yang terpilih dengan menggunakan aturan-aturan dalam basis pengetahuan yang sesuai
 - *Scheduler* : akan mengontrol agenda
 - *Consistency Enforcer* : akan berusaha memelihara konsistensi dalam mempresentasikan solusi yang bersifat darurat.
4. *Blackboard*. Merupakan area dalam memori yang digunakan untuk merekam kejadian yang sedang berlangsung, termasuk keputusan sementara. Ada 3 tipe keputusan yang dapat direkam yaitu :
 - Rencana : bagaimana menghadapi masalah
 - Agenda : aksi-aksi yang potensial yang sedang menunggu untuk dieksekusi
 - Solusi : calon aksi yang akan dibangkitkan
5. Antarmuka. Digunakan untuk media komunikasi antara user dan program

6. Subsistem penjelasan. Digunakan untuk melacak respon dan memberikan penjelasan tentang kelakuan sistem pakar secara interaktif melalui pertanyaan :

- Mengapa suatu pertanyaan ditanyakan sistem pakar ?
- Bagaimana konklusi dicapai ?
- Mengapa ada alternatif yang dibatalkan ?
- Rencana apa yang digunakan untuk mendapatkan solusi

Sistem penyaring pengetahuan. Sistem ini digunakan untuk mengevaluasi kerja sistem pakar itu sendiri untuk melihat apakah pengetahuan-pengetahuan yang ada masih cocok untuk digunakan dimasa mendatang.

2.1.3. Karakteristik Sistem Pakar

Adapun ciri-ciri sistem pakar sebagai berikut :

1. Terbatasnya pada bidang keahlian yang spesifik
2. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti
3. Dapat menggemukkan rangkaian alasan-alasan yang diberikan dengan cara yang dapat dipahami
4. Berdasarkan rule atau kaidah tertentu
5. Dirancang untuk dikembangkan secara bertahap
6. Keluaran bersifat anjuran atau nasehat
7. Keluaran tergantung dari dialog dengan user

2.2. K-Nearest Neighbour

2.2.1. Definisi K-Nearest Neighbour

K-Nearest Neighbour (k -NN) merupakan sebuah metode algoritma untuk melakukan klasifikasi terhadap objek baru berdasarkan (k) tetangga terdekatnya (Gorunescu, 2011). k -NN termasuk algoritma *supervised learning*, dimana hasil dari *query instance* yang baru, diklasifikasikan berdasarkan mayoritas dari kategori pada k -NN. Kelas yang paling banyak muncul yang akan menjadi kelas hasil klasifikasi.

Kelebihan k -NN :

- Lebih efektif di data training yang besar.
- Dapat menghasilkan data yang lebih akurat.

Kekurangan k -NN :

- Perlu ditentukan nilai k yang paling optimal yang menyatakan jumlah tetangga terdekat.
- Biaya komputasi cukup tinggi karena perhitungan jarak harus dilakukan pada setiap *query instance* bersama-sama dengan seluruh instan dari training sample.

2.2.2. Prinsip Kerja K-Nearest Neighbour

Prinsip kerja *k-Nearest Neighbour* (k -NN) adalah dengan mencari jarak terdekat antara data yang akan dievaluasi dengan k tetangga

(*Neighbour*) terdekatnya dalam data pelatihan. Berikut urutan proses kerja k -NN :

1. Menentukan parameter k (jumlah tetangga paling dekat). Parameter kepada testing ditentukan berdasarkan nilai k optimum pada saat training. Nilai k optimum diperoleh dengan mencoba-coba.
2. Menghitung kuadrat jarak *euclidean* (*euclidean distance*) masing-masing obyek terhadap data sampel yang diberikan.

$$d_i = \sqrt{\sum_{i=1}^p (x_{2i} - x_{1i})^2}$$

Keterangan:

x_1 = Sampel Data

x_2 = Data Uji / Testing

i = Variabel Data

d = Jarak

p = Dimensi Data

3. Mengurutkan objek-objek tersebut ke dalam kelompok yang mempunyai jarak *euclidean* terkecil
4. Mengumpulkan kategori y (klasifikasi *nearest neighbor*)

2.3. PHP

PHP adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk memprogram situs web dinamis. PHP dapat digunakan untuk membangun sebuah CMS. Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web (Aditya, 2011).

Kelebihan bahasa pemrograman PHP :

1. Web server yang mendukung PHP dapat ditemukan dimana-mana dari mulai Apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
2. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan developer yang siap membantu dalam pengembangan.
3. Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak.

PHP adalah bahasa open source yang dapat digunakan di berbagai sistem operasi (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah sistem.

2.4. My SQL

My SQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang multi-thread dan multi-user dengan sekitar 6 juta

instalasi di seluruh dunia. MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL (Aditya, 2011).

MySQL memiliki beberapa kelebihan antara lain :

- Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
- Perangkat lunak sumber terbuka. MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
- *Multi-user*. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
- *Performance tuning*. MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
- Keamanan. MySQL memiliki beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.

Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

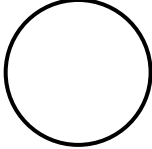
2.5. Diagram Alir Data (DAD)




Diagram Alir Data (DAD) atau *Data Flow Diagram* (DFD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*) (Rosa, 2011).

DAD dapat digunakan untuk mempresentasikan sebuah sistem atau perangkat lunak pada beberapa level abstraksi. DAD dapat dibagi menjadi beberapa level yang lebih detail untuk mempresentasikan aliran informasi atau fungsi yang lebih detail. DAD menyediakan mekanisme untuk pemodelan fungsional ataupun pemodelan aliran informasi. Oleh karena itu, DAD lebih sesuai digunakan untuk memodelkan fungsi-fungsi perangkat lunak yang akan diimplementasikan menggunakan pemrograman terstruktur karena pemrograman terstruktur membagi-bagi bagiannya dengan fungsi-fungsi dan prosedur-prosedur.

Notasi-notasi pada DAD (Edward Yourdon dan Tom DeMarco) adalah sebagai berikut (Rosa, 2011) :

Tabel 2 Simbol DAD

Notasi	Keterangan
	<p>Proses atau fungsi/prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur didalam kode program.</p>

	<p>Catatan: nama yang diberikan pada sebuah proses biasanya berupa kata kerja.</p>
	<p>File atau basis data atau penyimpanan (<i>storage</i>); pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel pada basis data (<i>Entity Relationship Diagram</i>(ERD), <i>Conceptual Data Model</i> (CDM), <i>Physical Data Model</i> (PDM)).</p> <p>Catatan: nama yang diberikan pada sebuah penyimpanan biasanya kata benda.</p>
	<p>Entitas luar (<i>external entity</i>) atau masukan (input) atau keluaran (output) atau orang yang memakai/berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan.</p> <p>Catatan: nama yang digunakan pada masukan (input) atau keluaran (output) biasanya berupa kata benda.</p>
	<p>Aliran data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke</p>

	<p>masukan (input) atau keluaran (output).</p> <p>Catatan: nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data siswa” atau tanpa kata data misalnya “siswa”.</p>
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Berikut ini adalah tahapan-tahapan perancangan dengan menggunakan DAD (Rosa, 2011):

1. Membuat DAD level 0 atau sering disebut juga *Context Diagram*. DAD level 0 menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain. DAD level 0 digunakan untuk menggambarkan interaksi antara sistem yang akan dikembangkan dengan entitas luar.
2. Membuat DAD Level 1
DAD level 1 digunakan untuk menggambarkan modul-modul yang ada dalam sistem yang akan dikembangkan. DAD level 1 merupakan hasil *breakdown* DAD level 0 yang sebelumnya sudah dibuat.
3. Membuat DAD Level 2
Modul-modul pada DAD level 1 dapat di-*breakdown* menjadi DAD level 2. Modul mana saja yang harus di-*breakdown* lebih detail tergantung pada tingkat kedetailan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu untuk di-

breakdown lagi. Untuk sebuah sistem, jumlah DAD level 2 sama dengan jumlah modul pada DAD level 1 yang di-*breakdown*.

4. Membuat DAD Level 3 dan seterusnya

DAD level 3, 4, 5, dan seterusnya merupakan *breakdown* dari modul pada DAD level di-atasnya. *Breakdown* pada level 3, 4, 5, dan seterusnya aturannya sama persis dengan DAD level 1 atau level 2.

Pada satu diagram DAD sebaiknya jumlah modul tidak boleh lebih dari 20 buah. Jika lebih dari 20 buah modul, diagram akan terlihat rumit dan susah untuk dibaca sehingga menyebabkan sistem yang akan dikembangkan juga menjadi rumit.