

BAB II

LANDASAN TEORI

A. Tinjauan Pustaka

Tinaliah (2015) “Aplikasi Sistem Pakar Untuk Diagnosa Penyakit Hewan Ternak Sapi Dengan Bayesian Network”, pada jurnal ini membahas perancangan aplikasi sistem pakar menggunakan metode Bayesian Network berbasis web native, asil diagnosa akan menampilkan gejala penyakit yang telah dipilih pengguna, tipe penyakit, solusi penanganannya, dan nilai kepastian probabilitasnya yang akan diberikan oleh aplikasi, sehingga dapat memberikan kepastian kepada pengguna akan probabilitas dari penyakit yang diderita oleh hewan ternak sapi.

Relita Buaton dkk (2013), “Perancangan Sistem Pakar Diagnosa Penyakit pada Hewan Ternak Sapi berbasis Web”, pada jurnal ini membahas perancangan sistem pakar diagnosa penyakit pada hewan ternak sapi berbasis web dengan metode *forward chaining*.

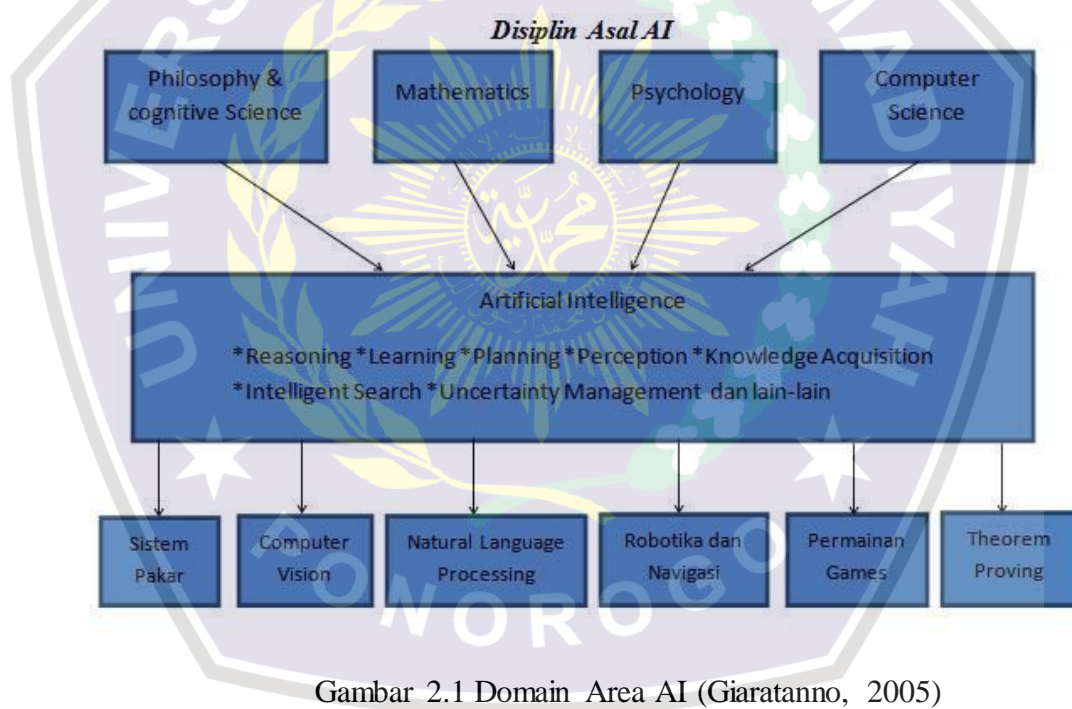
Pada tugas akhir ini penulis mencoba mengembangkan aplikasi sistem pakar dengan metode *forward chaining* berbasis *web*, perbedaan dengan kedua jurnal di atas adalah pengembangan aplikasi sistem pakar menggunakan *framework CodeIgniter* bukan menggunakan *PHP Native*.

B. Artificial Intelligence (AI)

Artificial Intelligence (AI) merupakan inovasi baru pada ilmu komputer yang sangat penting di era kini dan masa yang akan datang. Berasal dari

bahasa latin yaitu “intelligo” yang berarti “saya paham”, AI mempunyai tujuan untuk memahami dan melakukan aksi. Perkembangan AI bermula dari kemunculan komputer sekitar tahun 1940-an. Pada masa ini, perhatian difokuskan pada kemampuan komputer dalam mengerjakan sesuatu yang dapat dilakukan oleh manusia (Warwick, 2012). Dalam hal ini, komputer tersebut dapat mengadaptasi kemampuan kecerdasan dan perilaku manusia dalam memecahkan masalah.

Bidang Ilmu Kecerdasan Buatan Secara garis besar bidang ilmu yang dipelajari dalam bidang AI bisa dilihat pada gambar berikut ini :



Penjelasan lebih lanjut dari cabang ilmu AI di atas adalah :

1. Sistem Pakar Bidang ilmu ini mempelajari bagaimana membangun sistem atau komputer yang memiliki keahlian untuk memecahkan masalah dan menggunakan penalaran dengan meniru atau mengadopsi keahlian yang dimiliki oleh pakar/ahli. Dengan adanya sistem pakar, permasalahan yang

seharusnya bisa diselesaikan oleh seorang ahli dapat dilakukan oleh orang awam. Sedangkan untuk para ahli, sistem pakar dapat bekerja sebagai asisten yang seolah-olah sudah berpengalaman sehingga memudahkan pekerjaannya.

2. **Computer Vision** Computer vision adalah ilmu dan teknologi mesin yang melihat, dimana mesin mampu mengekstrak informasi dari gambar yang diperlukan untuk menyelesaikan tugas tertentu. Computer vision merupakan kombinasi antara pengolahan citra dan pengenalan pola. Apabila cabang ilmu ini dikombinasikan dengan AI secara umum akan menghasilkan sebuah visual intelligence system.
3. **Natural Language Processing (NLP)** NLP merupakan domain dari AI yang mempelajari bagaimana bahasa alam diolah sedemikian rupa sehingga user dapat berkomunikasi dengan computer. Konsentrasi ilmu ini adalah computer dapat melakukan ekstraksi informasi dari input yang berupa bahasa alami dan atau menghasilkan output yang juga berupa bahasa alami. Contoh dari NLP adalah robot ELIZA.
4. **Robotika dan Sistem Navigasi** Bidang ilmu inilah yang mempelajari bagaimana merancang robot yang berguna bagi industry dan mampu meringankan beban manusia, bahkan nantinya bisa menggantikan fungsi manusia. Robot dilengkapi dengan actuator seperti lengan, roda, kaki, juga sensor yang memungkinkan mereka untuk menerima dan bereaksi terhadap lingkungan sekitar. Robot humanoid pertama diciptakan oleh Al-Jajari (1136-1206) yang berfungsi sebagai musisi.

5. Permainan GamesGame biasanya memiliki karakter yang dikontrol oleh user, dan karakterlawan yang dikontrol oleh game itu sendiri. Dalam membuat game kita harusmerancang aturan-aturan yang nantinya akan dikerjakan oleh karakter lawan.Game akan menjadi menarik apabila karakter lawan dapat merespon denganbaik terhadap apa yang dilakukan oleh player. Tujuan pembuatan game adalahmembuat karakter lawan mempunyai strategi yang cerdas untuk mengalahkanplayer. Pada bidang ini AI dibutuhkan yaitu untuk merancang danmenghasilkan game yang menyenangkan serta antarmuka antara user –machine yang cerdas dan menarik untuk dimainkan.

C. Pengertian Sistem

Sistem adalah suatu kesatuan utuh yang terdiri dari beberapa bagian yang saling berhubungan dan berinteraksi untuk mencapai tujuan tertentu. (Teguh Wahyono, 2004)

Menurut Zulkifli (1997), Sistem adalah himpunan suatu benda nyata atau abstrak yang terdiri dari bagian-bagian atau komponen-komponen yang saling berkaitan, berhubungan, berketergantungan, dan saling mendukung, yang secara keseluruhan bersatu dalam satu kesatuan untuk mencapai tujuan tertentu secara efisien dan efektif.

Menyangkut pengertian tentang sistem menurut Jogiyanto (2009) mengemukakan sebagai kumpulan dari komponen yang saling berhubungan satu dengan yang lainnya membentuk satu kesatuan untuk mencapai tujuan tertentu.

D. Sistem Pakar

1. Definisi

Pengertian Sistem Pakar Sistem pakar (expert system) menurut Durkin adalah suatu program komputer yang dirancang untuk memodelkan kemampuan penyelesaian masalah yang dilakukan oleh seorang pakar. Biasanya sistem pakar berupa perangkat lunak pengambil keputusan yang mampu mencapai tingkat performa yang sebanding dengan seorang pakar dalam bidang masalah yang khusus dan sempit berdasarkan ilmu, fakta dan teknik berpikir yang berasal dari seorang pakar.

Sistem pakar terdiri dari dua komponen utama yaitu : basis pengetahuan (knowledge base) dan alat pengambilan kesimpulan (inference engine).

2. Konsep Dasar Sistem Pakar

Konsep dasar dari sistem pakar meliputi yaitu keahlian (*expertise*), ahli (*experts*), pemindahan keahlian (*transferring expertise*), inferensi (*inferencing*), aturan (*rules*) dan kemampuan menjelaskan (*explanation capability*).

Keahlian (*expertise*) adalah suatu kelebihan penguasaan pengetahuan dibidang tertentu yang diperoleh dari pelatihan, membaca atau pengalaman. Contoh bentuk pengetahuan yang termasuk keahlian adalah:

- a. Fakta-fakta pada lingkup permasalahan tertentu.
- b. Teori-teori pada lingkup permasalahan tertentu.

- c. Prosedur-prosedur dan aturan-aturan berkenaan dengan lingkup permasalahan tertentu.
- d. Strategi-strategi global untuk menyelesaikan masalah.
- e. Meta-knowledge (pengetahuan tentang pengetahuan)

Ada dua macam pengetahuan yaitu pengetahuan dari sumber yang ahli dan pengetahuan dari sumber yang tidak ahli. Ahli (expert) adalah seorang individu yang memiliki kemampuan pemahaman yang superior dari suatu masalah. Pengetahuan dari sumber yang ahli dapat digunakan untuk mengambil keputusan dengan cepat dan tepat.

Seorang ahli mampu menjelaskan suatu tanggapan, mempelajari hal baru seputar topik permasalahan, menyusun kembali pengetahuan jika dipandang perlu, memecah aturan-aturan jika dibutuhkan, dan menentukan relevansi tidaknya keahlian mereka. Pengalihan keahlian dari para ahli ke komputer untuk kemudian dialihkan lagi ke orang lain yang bukan ahli.

3. Tujuan Sistem Pakar

Adapun tujuan dari sistem pakar adalah untuk memindahkan kemampuan dari seorang ahli atau sumber keahlian lain ke dalam komputer dan kemudian memindahkannya kepada pemakai yang bukan seorang ahli. Proses ini meliputi empat aktifitas, sebagai berikut (Kusumadewi, 2003):

- a. Akuisisi pengetahuan (*knowledge acquisition*) yaitu kegiatan mencari dan mengumpulkan pengetahuan dari para ahli atau sumber keahlian yang lain.

- b. Representasi pengetahuan (*knowledge representation*) yaitu proses menyusun dan menyimpan pengetahuan yang didapatkan dari seorang ahli ke dalam computer. Pengetahuan berupa fakta dan aturan yang disimpan computer sebagai sebuah komponen disebut basis pengetahuan.
- c. Inferensi pengetahuan (*knowledge inferencing*) yaitu proses penalaran dengan melakukan inferensi berdasarkan pengetahuan yang sudah ditanamkan pada computer.
- d. Pemindahan pengetahuan (*knowledge transfer*) yaitu kegiatan memindahkan pengetahuan dari computer ke pemakai yang bukan ahli.

4. Ciri-Ciri Sistem Pakar

Sistem pakar yang baik harus memenuhi ciri-ciri sebagai berikut (Kusrini, 2006):

- a. Terbatas pada bidang yang spesifik
- b. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti
- c. Dapat mengemukakan rangkaian alasan yang diberikannya dengan cara yang mudah dipahami
- d. Knowledge base dan inference engine terpisah
- e. Memiliki fasilitas informasi yang handal
- f. Mudah dimodifikasi
- g. Dapat digunakan pada berbagai jenis computer
- h. Memiliki kemampuan untuk belajar beradaptasi

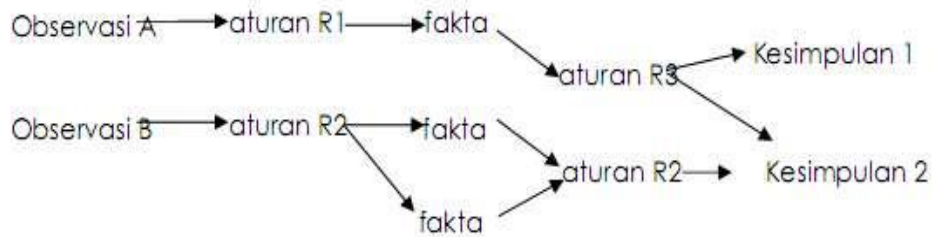
E. Metode *Forward Chaining*

Forward chaining merupakan metode inferensi yang melakukan penalaran dari suatu masalah kepada solusinya. Jika klausa premis sesuai dengan situasi (bernilai *TRUE*), maka proses akan menyatakan konklusi. *Forward chaining* adalah data-driven karena inferensi dimulai dengan informasi yang tersedia dan baru konklusi diperoleh. Jika suatu aplikasi menghasilkan tree yang lebar dan tidak dalam, maka gunakan forward chaining. (Sumber: Efrain Turban, 2005). *Forward Chaining* digunakan jika:

1. Banyak aturan berbeda yang dapat memberikan kesimpulan yang sama.
2. Banyak cara untuk mendapatkan sedikit konklusi.
3. Benar-benar sudah mendapatkan pelbagai fakta, dan ingin mendapatkan konklusi dari fakta-fakta tersebut.

Adapun tipe sistem yang dapat menggunakan teknik pelacakan forward chaining, yakni :

1. Sistem yang direpresentasikan dengan satu atau beberapa kondisi.
2. Untuk setiap kondisi, sistem mencari rule-rule dalam knowledge base untuk rule-rule yang berkorespondensi dengan kondisi dalam bagian if.
3. Setiap *rule* dapat menghasilkan kondisi baru dari konklusi yang diminta pada bagian then. Kondisi baru ini dapat ditambahkan ke kondisi lain yang sudah ada.
4. Setiap kondisi yang ditambahkan ke sistem akan diproses. Jika ditemui suatu kondisi, sistem akan kembali ke langkah 2 dan mencari rule-rule dalam knowledge base kembali. Jika tidak ada konklusi baru, sesi ini berakhir.



Gambar 2.2 Diagram *Forward Chaining*

F. Sapi

Sapi adalah ternak terpenting dari jenis-jenis hewan ternak yang dipelihara manusia sebagai sumber daging, susu, tenaga kerja, dan kebutuhan manusia lainnya. Ternak sapi menghasilkan 50 % kebutuhan daging di dunia, 95 % kebutuhan susu, dan kulitnya menghasilkan sekiranya 85 % kebutuhan kulit untuk sepatu. Sapi adalah salah satu genus dari famili Bovidae. Ternak atau hewan-hewan lainnya yang termasuk famili ini ialah Bison, banteng (Bibos), kerbau (Bubalus), kerbau Afrika (Syncherus), dan Anoa. Oleh karena itu satu genus dengan Eropa dan Bos Taurus dan sapi-sapi tropis atau Bos indicus. (Pane, 1993)

Usaha ternak merupakan suatu proses mengkombinasikan faktor-faktor produksi berupa lahan, ternak, tenaga kerja, dan modal untuk menghasilkan produk peternakan. Keberhasilan usaha ternak sapi bergantung pada tiga unsur, yaitu bibit, pakan, dan manajemen atau pengelolaan. Manajemen mencakup pengelolaan perkawinan, pemberian pakan, perkandangan, dan kesehatan ternak. Manajemen juga mencakup penanganan hasil ternak, pemasaran, dan pengaturan tenaga kerja (Abidin, 2002)

G. *CodeIgniter*

Menurut Hakim (2010:8) *CodeIgniter* adalah sebuah *framework* PHP yang dapat membantu mempercepat developer dalam pengembangan aplikasi *web* berbasis PHP dibanding jika menulis semua kode program dari awal. *CodeIgniter* pertama kali dibuat oleh Rick Ellis, CEO *Ellislab, Inc.* (<http://ellislab.com>), sebuah perusahaan yang memproduksi CMS (*Content Management System*) yang cukup handal, yaitu *Expression Engine* (<http://www.expressionengine.com>). Saat ini, *CodeIgniter* dikembangkan dan dipelihara oleh *Expression Engine Development Team*. Adapun beberapa keuntungan menggunakan *CodeIgniter*, diantaranya:

1. Gratis

CodeIgniter berlisensi dibawah Apache/BSD opensource.

2. Ditulis Menggunakan PHP 4

Meskipun *CodeIgniter* dapat berjalan di PHP 5, namun sampai saat ini kode program *CodeIgniter* masih dibuat dengan menggunakan PHP 4.

3. Berukuran Kecil

Ukuran *CodeIgniter* yang kecil merupakan keunggulan tersendiri. Dibanding dengan *framework* lain yang berukuran besar.

4. Menggunakan Konsep MVC

CodeIgniter menggunakan konsep MVC yang memungkinkan pemisahan layer application-logic dan presentation.

5. URL yang Sederhana

Secara default, URL yang dihasilkan *CodeIgniter* sangat bersih dan Search Engine Friendly (SEF).

6. Memiliki Paket Library yang Lengkap

CodeIgniter mempunyai library yang lengkap untuk mengerjakan operasi-operasi yang umum dibutuhkan oleh sebuah aplikasi berbasis web, misalnya mengakses database, mengirim email, memvalidasi form, menangani session dan sebagainya.

7. Extensible

Sistem dapat dikembangkan dengan mudah menggunakan plugin dan helper, atau dengan menggunakan hooks.

8. Tidak Memerlukan Template Engine

Meskipun CodeIgniter dilengkapi dengan template parser sederhana yang dapat digunakan, tetapi hal ini tidak mengharuskan kita untuk menggunakannya.

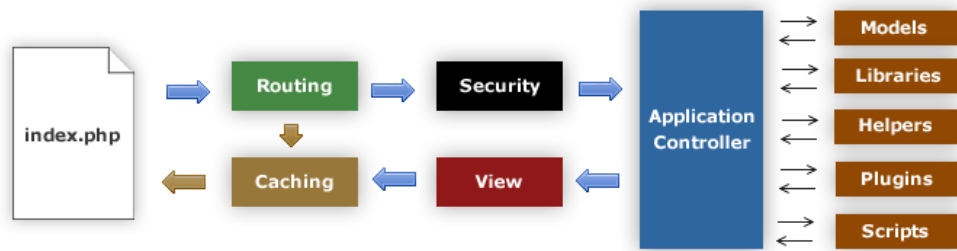
9. Dokumentasi Lengkap dan Jelas

Dari sekian banyak framework, CodeIgniter adalah satu-satunya framework dengan dokumentasi yang lengkap dan jelas.

10. Komunitas

Komunitas CodeIgniter saat ini berkembang pesat. Salah satu komunitasnya bisa dilihat di (<http://codeigniter.com/forum/>).

Proses aliran data aplikasi pada sistem dapat diilustrasikan seperti terlihat pada gambar 2.5.



Gambar 2.3 *Application Flowchart*

Keterangan :

1. Index.php berfungsi sebagai front controller, menginisialisasi base resource untuk menjalankan CodeIgniter.
2. Router memeriksa HTTP request untuk menentukan apa yang harus dilakukan dengannya.
3. Jika Cache aktif, maka hasilnya akan langsung dikirimkan ke browser dengan mengabaikan aliran data normal.
4. Security. Sebelum Controller dimuat, HTTP request dan data yang dikirimkan user akan difilter untuk keamanan.
5. Controller memuat model, core libraries, plugins, helpers dan semua resource yang diperlukan untuk memproses request.
6. Akhirnya View yang dihasilkan akan dikirimkan ke browser. Jika Cache aktif, maka View akan disimpan sebagai Cache dahulu, sehingga pada request berikutnya langsung dapat ditampilkan.

H. Web

1. Definisi *Web*

Web adalah salah satu aplikasi yang berisikan dokumen-dokumen multimedia (teks, gambar, suara, animasi, video) di dalamnya yang menggunakan protokol *HTTP* (*hypertext transfer protocol*) dan untuk mengaksesnya menggunakan perangkat lunak yang disebut *browser*.(Rudianto, 2011).

Browser (perambah) adalah aplikasi yang mampu menjalankan dokumen-dokumen *web* dengan cara diterjemahkan. Prosesnya dilakukan oleh komponen yang terdapat di dalam aplikasi *browser* yang biasa disebut *web engine*. Semua dokumen *web* ditampilkan oleh *browser* dengan cara diterjemahkan.

Situs *Web* adalah dokumen-dokumen *web* yang terkumpul menjadi satu kesatuan yang memiliki *Unified Resource Locator* (URL atau *domain*) dan biasanya di-*publish* di *internet* atau *intranet*. (Rudianto. 2011).

2. Definisi Situs *Web*

Situs *web* merupakan kumpulan dari halaman *web* yang sudah dipublikasikan di jaringan *internet* dan memiliki domain atau URL (*Uniform Resource Locator*) yang dapat diakses semua pengguna *internet* dengan cara mengetikkan alamatnya. Berikut adalah contoh alamat situs *web*: www.amiko.ac.id, <http://rudyantoarief.com>. (Rudianto. 2011).

I. PHP (Hypertext Preprocessor)

1. Definisi PHP (Hypertext Preprocessor)

PHP adalah bahasa pemrograman *script* yang paling banyak dipakai saat ini. *PHP* banyak dipakai untuk memrogram situs *web* dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain. Contoh terkenal dari aplikasi *PHP* adalah forum (*phpBB*) dan *Media Wiki* (*software* di belakang *Wikipedia*). *PHP* juga dapat dilihat sebagai pilihan lain dari *ASP.NET/C#/VB.NET Microsoft*, *ColdFusion Macromedia*, *JSP/Java Sun Microsystems*, dan *CGI/Perl*. Contoh aplikasi lain yang lebih kompleks berupa *CMS* yang dibangun menggunakan *PHP* adalah *Mambo*, *Joomla*, *Postnuke*, *Xaraya*, dan lain-lain. (Rudianto. 2011).



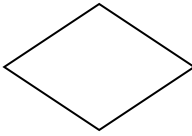

2. Kelebihan PHP (Hypertext Preprocessor)

1. Bahasa pemrograman *PHP* adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. *Web Server* yang mendukung *PHP* dapat ditemukan dimana - mana dari mulai *apache*, *IIS*, *Lighttpd*, hingga *Xitami* dengan konfigurasi yang relatif mudah.
3. *PHP* adalah bahasa *open source* yang dapat digunakan di berbagai mesin (*Linux*, *Unix*, *Macintosh*, *Windows*) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah sistem.

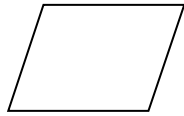
J. Flowchart

Flowchart dapat menunjukkan secara jelas arus pengendalian suatu *algoritma*, yakni melaksanakan suatu rangkaian kegiatan secara *logis* dan *sistematis*. Suatu diagram alur dapat memberi gambaran dua *dimensi* berupa simbol-simbol grafis. Masing-masing simbol telah ditetapkan lebih dahulu fungsi dan artinya. Simbol-simbol tersebut dipakai untuk menunjukkan berbagai kegiatan operasi dan jalur pengendalian. Arti khusus dari sebuah *flowchart* adalah simbol-simbol yang digunakan untuk menggambarkan urutan proses yang terjadi di dalam suatu program komputer secara *sistematis* dan *logis*. (Sutabri, 2004).

Tabel 2.1. Simbol *Flowchart*

Simbol <i>Flowchart</i>	Fungsi
	TERMINAL Simbol ini digunakan untuk mengawali atau mengakhiri suatu proses/kegiatan.
	PREPARATION Simbol ini digunakan untuk mempersiapkan harga awal/nilai awal suatu variabel yang akan diproses.
	DECISION Simbol ini digunakan untuk pengujian suatu kondisi yang sedang diproses.
	PROSES Simbol ini digunakan untuk menggambarkan

suatu proses yang sedang dieksekusi.



INPUT/OUTPUT

Simbol ini digunakan untuk menggambarkan proses input (*read*) maupun proses output (*print*).



SUBROUTINE

Simbol ini digunakan untuk menggambarkan proses pemanggilan subprogram dari main program.



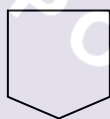
FLOW LINE

Simbol ini digunakan untuk menggambarkan arus proses dari suatu kegiatan ke kegiatan lain.



CONNECTOR

Simbol ini digunakan sebagai penghubung antara suatu proses dengan proses lainnya yang ada di dalam suatu lembar halaman.



PAGE CONNECTOR

Simbol ini digunakan sebagai penghubung antara suatu proses dengan proses lainnya, tetapi berpindah halaman.



MANUAL OPERATION

Simbol ini digunakan untuk menggambarkan suatu kegiatan atau proses yang bersifat manualisasi.



PRINTER

Digunakan untuk menggambarkan suatu kegiatan mencetak suatu informasi dengan mesin printer.



CONSOLE

Simbol ini digunakan untuk menggambarkan suatu kegiatan menampilkan data atau informasi melalui monitor atau *CRT (Cathode Ray Tube)*.



DISK

Simbol ini digunakan untuk menggambarkan suatu kegiatan membaca atau menulis data menggunakan media *magnetic disk*.



MANUAL INPUT

Simbol ini digunakan untuk menggambarkan proses pemasukan data melalui media keyboard.






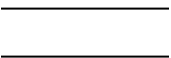
TAPE

Simbol ini digunakan untuk menggambarkan suatu kegiatan membaca atau menulis data menggunakan media *magnetic tape*.

K. Data Flow Diagram (DFD)

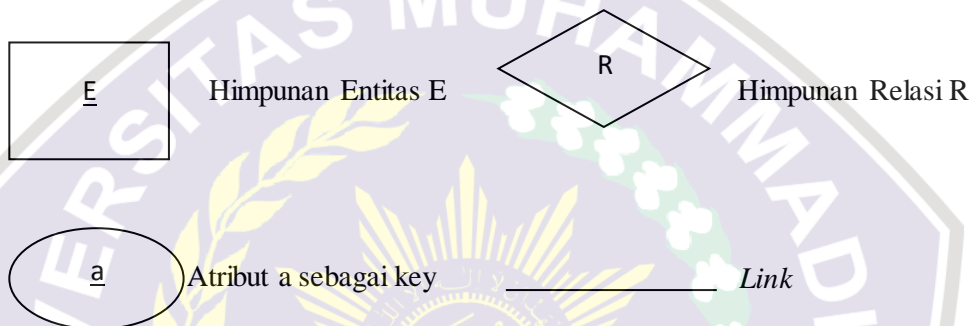
Data Flow Diagram(DFD) adalah representasi grafik dari sebuah sistem. *DFD* menggambarkan komponen-komponen sebuah sistem, aliran-aliran data di mana komponen-komponen tersebut, dan asal, tujuan, dan penyimpanan dari data tersebut. Kita dapat menggunakan *DFD* untuk dua hal utama, yaitu untuk membuat dokumentasi dari sistem informasi yang ada, atau untuk menyusun dokumentasi untuk sistem informasi yang baru. Empat simbol yang digunakan : (Sutabri, 2004)

Tabel 2.2. Simbol *Data Flow Diagram (DFD)*

Notasi	Fungsi
	Simbol <i>entitaseksternal</i> atau <i>terminator</i> menggambarkan asal atau tujuan data di luar sistem
	Simbol <i>lingkaran</i> menggambarkan entitas atau proses dimana aliran data masuk ditransformasikan ke aliran data keluar
	Simbol aliran data menggambarkan aliran data
	Simbol <i>file</i> menggambarkan tempat data disimpan

L. Entity Relationship Diagram (ERD)

Model *entity-relationship* yang berisi komponen-komponen. Himpunan *entitas* dan himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang mempresentasikan seluruh fakta dari dunia nyata yang kita tinjau, dapat digambarkan dengan lebih sistematis dengan menggunakan diagram *entity-relationship* (diagram *E-R*). Notasi-notasi simbolik didalam diagram *E-R* yang dapat kita gunakan adalah :



Gambar 2.4 Kardinalitas relasi

Berikut adalah contoh penggambaran relasi antar himpunan *entitas* lengkap dengan kardinalitas relasi dan atribut-atributnya :

1. Relasi satu-ke-satu (*one-to-one*)
2. Relasi satu-ke-banyak (*one-to-many*).
3. Relasi banyak-ke-banyak (*many-to-many*).

M. Basis Data

1. Pengertian Basis Data

Basis dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan data merupakan representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa,

pembeli, pelanggan), barang, hewan peristiwa, konsep, keadaan, dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya. Basis data (*database*) merupakan kumpulan data yang saling berhubungan (punya relasi). (Yakub, 2012). Menurut Janner (2007), Basis Data adalah koleksi data yang bisa mencari secara menyeluruh dan secara sistematis memelihara dan *me-retrieve* informasi.

2. Manfaat Basis Data

- a. Kecepatan dan kemudahan (*Speed*), pemanfaatan basis data memungkinkan untuk dapat, menyimpan, merubah, dan menampilkan kembali data tersebut dengan lebih cepat dan mudah.
- b. Efisiensi ruang penyimpanan (*space*), dengan basis data efisiensi atau optimalisasi penggunaan ruang penyimpanan dapat dilakukan, karena penekanan jumlah redudansi data, baik dengan sejumlah pengkodean atau dengan membuat label-label yang saling berhubungan.
- c. Keakuratan (*accuracy*), pembentukan relasi antar data bersama dengan penerapan aturan atau batasan (*constraint*) tipe, domain dan keunikan data dapat diterapkan dalam sebuah basis data.
- d. Ketersediaan (*availability*), dapat memilah data utama atau master, transaksi, data histori hingga data kadaluwarsa. Data yang jarang atau tidak digunakan lagi dapat diatur dari sistem basis data yang akif.

- e. Keamanan (*security*), untuk menentukan siapa-siapa yang berhak menggunakan basis data beserta objek-objek di dalamnya dan menentukan jenis-jenis operasi apa saja yang boleh dilakukan.
- f. Kebersamaan pemakai (*sharebility*), basis data dapat digunakan oleh beberapa pemakai dan beberapa lokasi. Basis Data yang dikelola oleh sistem (aplikasi) yang mendukung *multiuser* dapat memenuhi kebutuhan, akan tetapi harus menghindari inkonsistensi data. (Yakub, 2012).

3. Operasi Basis Data

Pada sebuah *disk (harddisk)*, basis dapat diciptakan dapat pula ditiadakan. Pada sebuah *disk* juga dapat menempatkan beberapa basis data, misalnya basis data kepegawaian, akademik, penjualan, perpustakaan dan lain-lain. Sementara dalam sebuah basis data dapat ditempatkan pada satu *file* atau tabel barang, faktur, pelanggan dan transaksi barang. Operasi-operasi dasar yang dapat dilakukan basis data adalah :

- a. Pembuatan basis data baru (*Create Database*)
- b. Penghapusan basis data (*Drop Database*)
- c. Pembuatan file atau tabel baru ke suatu basis data (*Create Table*)
- d. Penghapusan file atau tabel dari suatu basis data (*Drop Table*)
- e. Penambahan atau pengisian data baru di sebuah basis data (*Insert*)
- f. Pengambilan data dari sebuah file atau tabel (*Retrieve* atau *Search*)
- g. Pengubahan data dalam sebuah file atau tabel (*Update*)
- h. Penghapusan data dari sebuah file atau tabel (*Delete*)

Operasi pembuatan basis data dan tabel merupakan operasi awal yang hanya dilakukan sekali dan berlaku seterusnya. Sedangkan untuk operasi pengisian data merupakan operasi rutin yang dilakukan berulang ulang. (Yakub, 2012)

4. Persyaratan Basis Data

Ketentuan yang harus diperhatikan pada pembuatan *file* basis data agar dapat memenuhi kriteria sebagai basis data, yaitu: *redudansi* data, inkonsistensi data, pengaksesan data, data terisolasi untuk standarisasi, masalah keamanan, masalah integritas data, data *multiuser*.

a. *Redudansi* dan Inkonsistensi Data

Penyimpanan data yang sama di beberapa tempat disebut *redudansi*, hal ini akan menyebabkan pemborosan dan menimbulkan inkonsistensi data (data tidak konsisten) karena bila terjadi maka data harus dirubah pada beberapa tempat, hal ini tentunya tidak efisien.

b. Pengaksesan Data

Data di dalam basis data harus siap diakses oleh siapa saja yang membutuhkan dan mempunyai hak untuk mengaksesnya. Oleh karena itu perlu dibuat suatu program pengelolaan atau suatu aplikasi untuk mengakses data yang dikenal sebagai *Database Management System (DBMS)*.

c. Data Terisolasi untuk Standarisasi

Jika data tersebar dalam beberapa *file* dalam bentuk *format* yang tidak sama, maka akan menyulitkan dalam menulis program

aplikasi, baik untuk mengambil dan menyimpan data. Oleh karena itu ada dalam satu *database* harus dibuat satu *format* yang sama, sehingga mudah dibuat program aplikasinya.

d. Masalah Keamanan atau *Security*

Setiap pemakai sistem basis data tidak semua bagian diperbolehkan untuk mengakses semua data, misalnya data mengenai gaji pegawai hanya boleh dibuka oleh bagian keuangan, sedang bagian gudang dan bagian lain tidak diperkenankan untuk membukanya. Keamanan dapat diatur dan disesuaikan baik ditingkat basis data atau aplikasinya.

e. *Multiple User*

Salah satu alasan basis data dibangun karena nantinya data tersebut akan digunakan oleh banyak orang, baik dalam waktu berbeda maupun bersamaan. Oleh karena itu diperlukan basis data yang handal dan dapat mendukung banyak pemakai atau multiuser. (Yakub, 2012).