

BAB II

LANDASAN TEORI

A. Kecerdasan Buatan

Kecerdasan buatan adalah suatu ilmu yang mempelajari tentang cara membuat komputer agar memiliki kemampuan bekerja seperti dan sebaik manusia. Adapun definisi lain, kecerdasan buatan merupakan kawasan penelitian, aplikasi, dan instruksi yang terkait dengan pemrograman komputer untuk melakukan sesuatu hal yang dalam pandangan manusia adalah cerdas (H. A. Simon : 1987).

Ada tiga tujuan kecerdasan buatan, yaitu : membuat komputer lebih cerdas, mengerti tentang kecerdasan, dan membuat mesin lebih berguna. Teknologi kecerdasan buatan dipelajari dalam beberapa bidang penelitian, seperti (Kusrini, 2006 : 5):

1. *Formal tasks* (matematika, *games*).
2. *Perceptron, robotics, natural language, common sense, reasoning*.
3. *Expert tasks* (*financial analysis, medical diagnostics, engineering, scientific analysis, dll*).

Ada beberapa konsep yang harus dipahami dalam kecerdasan buatan (Kusrini, 2006 : 5) :

1. *Turing Test-Metode* Pengujian Kecerdasan.
2. Pemrosesan Simbolik.
3. *Heuristic*.
4. Penarikan Kesimpulan (*Inferencing*).

5. Pencocokan Pola (*Pattern Matching*).

B. Sitem Pakar

Pada dasarnya sistem pakar diterapkan untuk mendukung aktifitas pemecahan masalah. Beberapa aktifitas pemecahan yang dimaksud, yaitu pembuat keputusan (*decision making*), pemaduan pengetahuan (*knowledge fusing*), pembuatan desain (*designing*), perencanaan (*planning*), prakiraan (*forecasting*), pengaturan (*regulating*), pengendalian (*controlling*), diagnosa (*diagnosing*), perumusan (*prescribing*), penjelasan (*explaining*), pemberian nasihat (*advising*), dan pelatihan (*tutoring*) (Kusrini, 2006 : 11).

Sistem pakar merupakan suatu sistem terkomputerisasi yang menirukan seorang pakar dalam mengatasi masalah yang rumit sesuai dengan pengetahuan yang dimilikinya. Penyelesaian masalah dapat diuji dan hasilnya akan sesuai dengan hasil yang dikerjakan oleh seorang pakar.

Untuk membangun sistem pakar yang baik diperlukan beberapa komponen, antara lain (Iswanti, 2008) :

1. Antar Muka Pengguna (*User Interface*)
2. Basis Pengetahuan (*Knowledge Base*)
3. Mekanisme Inferensi (*Inference Machine*)
4. Memori Kerja (*Working Memory*)

Antar Muka Pengguna, sistem pakar menggantikan seorang pakar dalam situasi tertentu, maka sistem harus menyediakan pendukung yang diperlukan oleh pemakai yang tidak memahami masalah teknis. Sistem pakar juga menyediakan komunikasi antar sistem dan pemakaiannya (*user*) yang disebut

sebagai antar muka. Antar muka yang efektif dan ramah penggunaan (*user-friendly*) penting sekali terutama bagi pemakai yang tidak ahli dalam bidang yang diterapkan pada sistem pakar. Sedangkan Basis pengetahuan, merupakan kumpulan pengetahuan bidang tertentu pada tingkatan pakar dalam format tertentu. Pengetahuan ini diperoleh dari akumulasi pengetahuan pakar dan sumber-sumber pengetahuan lainnya. Pada sistem pakar ini basis pengetahuan terpisah dengan mesin inferensi. Pemisahan ini bermanfaat untuk pengembangan sistem pakar secara leluasa disesuaikan dengan perkembangan pengetahuan. Dan Mesin inferensi sesungguhnya adalah program komputer yang menyediakan metodologi untuk melakukan penalaran tentang informasi pada basis pengetahuan dan memori kerja serta untuk merumuskan kesimpulan-kesimpulan. Komponen ini menyajikan arahan-arahan tentang bagaimana menggunakan pengetahuan dari sistem dengan membangun agenda yang mengelola dan mengontrol langkah-langkah yang diambil untuk menyelesaikan masalah ketika dilakukan konsultasi.

Memori kerja, merupakan bagian sistem pakar yang menyimpan fakta-fakta yang diperoleh saat dilakukan proses konsultasi. Fakta-fakta inilah yang nantinya akan diolah oleh mesin inferensi berdasarkan pengetahuan untuk menentukan suatu keputusan pemecahan masalah. Dan Fasilitas penjelasan dapat membrikan informasi kepada pemakai mengenai jalannya penalaran sehingga dihasilkan suatu keputusan. Tujuan adanya fasilitas penjelasan dalam sistem pakar antara lain membuat sistem menjadi lebih cerdas, menunjukkan adanya proses analisa dan yang tidak kalah pentingnya adalah memuaskan psikologis pemakai. Sedangkan Akuisisi pengetahuan adalah

proses pengumpulan, perpindahan, dan transformasi dari keahlian/kepakaran pemecahan masalah yang berasal dari beberapa sumber pengetahuan ke dalam bentuk yang dimengerti oleh komputer. Dengan demikian maka seorang pakar akan dengan mudah menambahkan pengetahuan ataupun kaidah baru pada sistem pakar. Untuk menjamin bahwa pengetahuan pada sistem pakar *up to date* dan valid, maka fasilitas akuisisi pengetahuan hanya bisa diakses oleh pakar. Pengguna awam tidak berhak memakai fasilitas akuisisi pengetahuan. (Iswanti, 2008)

C. *Smartphone*

1. Kerusakan

- a. Kerusakan *power on key*.
- b. Kerusakan *IC Power*.
- c. Kerusakan *IC CPU*.
- d. Kerusakan *IC Flash*.
- e. Kerusakan *IC UEM*.
- f. Kerusakan *EEPROM*.
- g. Kerusakan *IC Audio*.
- h. Kerusakan *Charge interface*.
- i. Kerusakan *keyboard interface sirquit*.
- j. Kerusakan *mic/speaker interface*.

2. Gejala

- a. Ponsel mati
- b. Tombol power tidak dapat ditekan atau dinyalakan

- c. Ponsel mati Dan sebelumnya muncul contact service
- d. LCD blank
- e. Restrart
- f. Tiba-tiba ponsel mati sendiri
- g. Tidak dapat mencharge
- h. Muncul pesan “*Insert Sim Card*”.
- i. Ponsel mati dan sebelumnya muncul contact service
- j. Tidak ada jaringan.
- k. Salah satu menu hilang.
- l. Muncul phone restic (cek IMEI).
- m. Ponsel dapat dinyalakan dan LCD blank dan muncul pesan contact service
- n. Speaker mati.
- o. Keypad pada ponsel tidak berfungsi.
- p. Speaker/mic mati.

D. PHP (*Hypertext Preprocessor*)

1. Pengertian PHP (*Hypertext Preprocessor*)

PHP (Hypertext Preprocessor) adalah bahasa *server side scripting* yang menyatu dengan HTML untuk membuat halaman *web* yang dinamis (Arief, 2011). Karena *PHP* merupakan *server side scripting* maka sintaks dan perintah-perintah *PHP* yang dieksekusi di *server* kemudian hasilnya dikirimkan ke *browser* dalam format *HTML*. Dengan demikian kode program yang ditulis dalam *PHP* tidak akan terlihat oleh *user* sehingga

keamanan halaman *web* lebih terjamin. *PHP* dirancang untuk membentuk halaman *web* yang dinamis, yaitu halaman *web* yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman *web*.

PHP termasuk dalam *open source product*, sehingga *source code PHP* dapat diubah dan didistribusikan secara bebas. Versi terbaru *PHP* dapat diunduh secara gratis di situs resmi *PHP*: <http://www.PHP.net>. *PHP* juga dapat berjalan pada berbagai *web server* seperti *IIS (Internet Information Server)*, *Apache*, dan *Xitami*. *PHP* juga mampu lintas *platform*. Artinya *PHP* dapat dibangun sebagai modul pada *web server Apache* dan sebagai *binary* yang dapat berjalan sebagai *CGI (Common Gateway Interface)*. (Rudiyanto, 2011).

Keunggulan *PHP* adalah :

- a. *PHP* memiliki tingkat akses yang lebih cepat
- b. *PHP* mampu berjalan di beberapa *server web*, seperti *Apache*, *Microsoft IIS*, dan *Xitami*
- c. *PHP* mendukung akses ke beberapa *database* baik yang bersifat gratis atau komersial, seperti *mSQL*, *MySQL*, dan *MicrosoftSQL Server*
- d. *PHP* bersifat *Open Source*

2. *Syntaks PHP*

Sintaks *PHP* ditulis dalam apitan tanda khusus *PHP*. Ada empat macam pasangan tag *PHP* yang digunakan :

- a. `<?PHP....?>`

b. `<script language="PHP">...</script>`

c. `<?...?>`

d. `<%....%>`

Point a dan *b* merupakan cara yang paling umum yang digunakan sekalipun cara *c* lebih praktis karena cara *c* tidak selalu diaktifkan pada konfigurasi *file PHP* yang terdapat pada direktori *c:\apache\php*. Cara *d* juga memungkinkan kemudahan bagi anda yang sudah terbiasa dengan *ASP (Active Server Pages)*. Namun, bila itu dikenal harus dilakukan pengaktifan file konfigurasi *PHP*.

E. 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*.

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*. (Arief, 2011).

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>. (Arief, 2011).

F. *Flowchart* (Diagram Alur)

1. Pengertian *flowchart* (Diagram Alur).

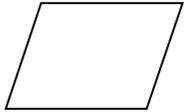
Karena komputer membutuhkan hal-hal yang rinci, maka bahasa pemrograman bukanlah alat baik untuk merancang sebuah *algoritma* awal. Alat yang banyak dipakai untuk membuat *algoritma* adalah diagram alur (*flowchart*).

Diagram alur 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).

2. Simbol-simbol *flowchart*.

Sudah dikemukakan di atas bahwa diagram alur atau *flowchart* memiliki beberapa simbol yang biasa digunakan untuk menggambarkan rangkaian proses yang harus dilaksanakan. Simbol-simbol tersebut dijelaskan di bawah ini: (Sutabri, 2004).

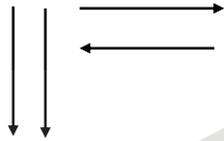
Tabel 2.1. Simbol *Flowchart*

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 Menggambarkan suatu proses yang sedang dieksekusi.
	INPUT/OUTPUT Menggambarkan proses input (<i>read</i>) maupun proses output (<i>print</i>).



SUBROUTINE

Menggambarkan proses pemanggilan subprogram dari main program.



FLOW LINE

Menggambarakan arus proses dari suatu kegiatan ke kegiatan lain.



CONECTOR

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



PAGE CONECTOR

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



MANUAL OPERATION

Menggambarkan suatu kegiatan atau proses yang bersifat manualisasi.



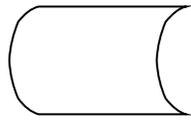
PRINTER

Menggambarkan suatu kegiatan mencetak suatu informasi dengan mesin printer.



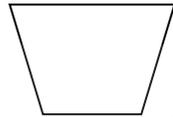
CONSOLE

Menggambarkan suatu kegiatan menampilkan data atau informasi melalui monitor atau *CRT* (*Cathode Ray Tube*).



DISK

Menggambarkan suatu kegiatan membaca atau menulis data menggunakan media *magnetic disk*.



MANUAL INPUT

Menggambarkan proses pemasukan data melalui media keyboard.



TAPE

Menggambarkan suatu kegiatan membaca atau menulis data menggunakan media *magnetic tape*.

Sumber : Sutabri. 2004

3. Jenis *flowchart*.

Bentuk diagram alur (*flowchart*) yang sering digunakan dalam proses pembuatan suatu program komputer adalah sebagai berikut:

a. Program *flowchart*.

Simbol-simbol yang menggambarkan proses secara rinci dan detil antara intruksi yang satu dengan intruksi yang lainnya dalam suatu program komputer yang bersifat *logic*.

b. Sistem *flowchart*.

Simbol-simbol yang menggambarkan urutan prosedur secara detil dalam suatu sistem komputerisasi. Bersifat fisik.

4. Teknik pembuatan *flowchart*.

Sebelum kita membuat sebuah program komputer, yang harus kita lakukan sebelumnya adalah membuat *flowchart*. Jenis *flowchart* yang sering digunakan adalah program *flowchart*.

Teknik pembuatan program *flowchart* ini dibagi menjadi dua bagian, yaitu:

a. *General way*.

Teknik pembuatan *flowchart* dengan cara ini lazim digunakan untuk menyusun logika suatu program. Teknik ini menggunakan pengulangan proses secara tidak langsung (*Non-Direct-Loop*).

b. *Iteration way*.

Teknik pembuatan *flowchart* dengan cara ini biasanya dipakai untuk logika program yang cepat dan bentuk permasalahannya kompleks. Pengulangan proses yang terjadi bersifat langsung (*Direct-Loop*). (Sutabri, 2004).

G. *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 :

Tabel 2.2 Simbol *Data Flow Diagram (DFD)*

Notasi	Fungsi
	Simbol <i>entitas eksternal</i> atau <i>terminator</i> menggambarkan asal atau tujuan data di luar sistem
	Simbol lingkaran 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

Sumber : Sutabri. 2004

Ada 3 (tiga) jenis *DFD*, yaitu :

1. *Diagram contex.*

Jenis pertama *Context diagram*, adalah data *flow diagram* tingkat atas (*DFD Top Level*), yaitu diagram yang paling tidak detail, dari sebuah sistem informasi yang menggambarkan aliran-aliran data ke dalam dan ke luar sistem dan ke dalam dan ke luar *entitas-entitas eksternal*. (*CD* menggambarkan sistem dalam satu lingkaran dan hubungan dengan *entitas* luar. Lingkaran tersebut menggambarkan keseluruhan proses dalam sistem).

Beberapa hal yang harus diperhatikan dalam menggambar *CD* :

a. Terminologi sistem :

- 1) Batas sistem adalah batas antara daerah kepentingan sistem.
- 2) Lingkungan sistem adalah segala sesuatu yang berhubungan atau mempengaruhi sistem tersebut.
- 3) *Interface* adalah aliran yang menghubungkan sebuah sistem dengan lingkungan sistem tersebut.

b. Menggunakan satu simbol proses.

Yang masuk didalam lingkaran konteks (simbol proses) adalah kegiatan pemrosesan informasi (Batas Sistem). Kegiatan informasi adalah mengambil data dari *file*, *mentransformasikan* data, atau melakukan *filig* data, misalnya mempersiapkan dokumen, memasukkan, memeriksa, mengklasifikasi, mengatur, menyortir, menghitung, meringkas data, dan melakukan *filig* data (baik yang melakukan secara manual maupun yang dilakukan secara *terotomasi*).

c. Nama/keterangan di simbol proses tersebut sesuai dengan fungsi sistem tersebut.

d. Antara *Entitas Eksternal/Terminator* tidak diperbolehkan komunikasi langsung.

e. Jika terdapat *termintor* yang mempunyai banyak masukan dan keluaran, diperbolehkan untuk digambarkan lebih dari satu sehingga mencegah penggambaran yang terlalu rumit, dengan memberikan tanda *asterik* (*) atau garis silang (#).

- f. Jika Terminator mewakili individu (personil) sebaiknya diwakili oleh peran yang dipertainkan *personil* tersebut.
- g. Aliran data ke proses dan keluar sebagai output keterangan aliran data berbeda.

2. *DFD* fisik.

DFD fisik adalah representasi grafik dari sebuah sistem yang menunjukkan *entitas-entitas internal* dan *eksternal* dari sistem tersebut, dan aliran-aliran data ke dalam dan keluar dari *entitas-entitas* tersebut. *Entitas-entitas internal* adalah *personal*, tempat (sebuah bagian), atau mesin (misalnya, sebuah komputer) dalam sistem tersebut yang *mentransformasikan* data. Maka *DFD* fisik tidak menunjukkan apa yang dilakukan, tetapi menunjukkan dimana, bagaimana, dan oleh siapa proses-proses dalam sebuah sistem dilakukan.

Perlu diperhatikan didalam memberikan keterangan di lingkaran-lingkaran (simbol proses) dan aliran-aliran data (simbol aliran data) dalam *DFD* fisik menggunakan label/keterangan dari kata benda untuk menunjukkan bagaimana sistem *mentransmisikan* data antara lingkaran-lingkaran tersebut.

3. *DFD* logis.

DFD logis dalah representasi grafik dari sebuah sistem yang menunjukkan proses-proses dalam sistem tersebut dan aliran-aliran data ke dalam dan ke luar dari proses-proses tersebut. Kita menggunakan *DFD* logis untuk membuat dokumentasi sebuah sistem informasi karena *DFD* logis dapat mewakili logika tersebut, yaitu apa yang dilakukan oleh

sistem tersebut, tanpa perlu menspesifikasi dimana, bagaimana, dan oleh siapa proses-proses dalam sistem tersebut dilakukan. Keuntungan dari *DFD* logis dibandingkan dengan *DFD* fisik adalah dapat memusatkan perhatian pada fungsi - fungsi yang dilakukan sistem.

H. *Entity Relationship Diagram (ERD)*

Entity Relationship Data (ERD) merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. *ERD* untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol. Ada tiga simbol yang digunakan, yaitu:

1. *Entity*

Entity adalah orang, tempat, kejadian atau konsep yang informasinya direkam. Pada bidang administrasi siswa misalnya, *entity* adalah siswa, buku, pembayaran, nilai test. Pada bidang kesehatan, *entity* adalah pasien, obat, kamar, diet. Simbol yang digunakan untuk *entity* adalah persegi panjang.

2. *Attribute*

Setiap *entity* mempunyai *attribute* atau sebutan untuk mewakili suatu *entity*. Seorang siswa dapat dilihat dari atributenya, misalnya nama, nomor siswa, alamat, nama orang tua, hobby. *Attribute* juga disebut sebagai data elemen, data *field*, data item. (Sumber : Kristanto, 2004, 2)

3. *Relationship*

Hubungan yang terjadi antara satu atau lebih *entity*. *Relationship* tidak mempunyai keberadaan fisik, kecuali yang mewarisi hubungan antara *entity* tersebut. *Relationship set* adalah kumpulan *relationship* yang sejenis. Simbol yang digunakan adalah belah ketupat, *diamond* atau *rectangel*. (Marlinda, 2004, 17)

- a. *One to one* (1:1)
- b. *One to many* (1:M) atau *many to one* (M:1)
- c. *Many to many* (M:M)

I. **Basis Data (*Database*)**

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 atau mempunyai relasi. (Yakub, 2008). Menurut Janner (2007:2), 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 reduansi data, baik dengan sejumlah pengkodean atau dengan membuat tabel-tabel 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 aktif.
- 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, 2008)

3. Operasi Basis Data

Pada sebuah *disk (hard disk)*, 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, 2008)

4. Persyaratan Basis Data

Ketentuan yang harus diperhatikan pada pembuatan file basis data agar dapat memenuhi kriteria sebagai basis data, yaitu: redundansi 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, 2008).

5. Sistem Basis Data

a. Pengertian Sistem Basis Data

Sistem basis data (*database*) merupakan sistem yang terdiri dari kumpulan file atau tabel yang saling berhubungan dan memungkinkan beberapa mengakses dan memanipulasinya. Sistem basis data juga merupakan suatu sistem yang menyusun dan mengelola data organisasi perusahaan, sehingga mampu menyediakan informasi yang diperlukan oleh pemakai. Istilah sistem basis data tentu saja berbeda dengan istilah basis data, sistem basis data merupakan lingkup yang lebih luas daripada basis data. (Yakub, 2008)

b. Komponen Sistem Basis Data

Sistem basis data terdapat komponen-komponen utama yaitu; perangkat keras (*hardware*), sistem operasi (*operating system*), basis data (*database*), program aplikasi (*application program*), *Database Management System* (DBMS), dan pemakai (*user*).

c. Database Management System (DBMS)

Database Management System (DBMS) merupakan kumpulan program aplikasi yang digunakan untuk membuat dan mengelola basis data. DBMS berisi suatu koleksi data dan satu set program untuk mengakses data. DBMS merupakan perangkat lunak (*software*) yang menentukan bagaimana data tersebut diorganisasi, disimpan, diubah dan diambil kembali. Perangkat ini juga menerapkan mekanisme pengamanan data, pemakaian data bersama, dan konsistensi data.

d. Pemakai (Users)

Pemakai atau *users* adalah beberapa jenis atau tipe pemakai pada sistem basis data, berdasarkan cara mereka berinteraksi pada basis data, diantaranya program aplikasi, pemakai mahir, pemakai umum dan pemakai khusus.

1) *Programmer* Aplikasi

Programmer aplikasi adalah pemakai yang berinteraksi dengan basis data melalui DML (*Data Manipulation Language*), yang disertakan dalam program yang ditulis dalam bahasa pemrograman induk (seperti *pascal*, *cobol*, *clipper*, *foxpro*, dan lain – lain).

2) *User* Mahir (*Casual User*)

Casual user adalah pemakai yang berinteraksi dengan sistem tanpa menulis modul program. Pemakai menggunakan *query*

(untuk akses data), dengan bahasa *query* yang telah disediakan oleh suatu DBMS.

3) *User Umum (Naive User)*

Naive user adalah pemakai yang berinteraksi dengan sistem basis data melalui pemanggilan satu program aplikasi permanen, yang telah ditulis atau disediakan sebelumnya.

4) *User Khusus (Specialized User)*

Specialized User adalah pemakai yang menulis aplikasi basis data non konvensional untuk keperluan khusus, seperti untuk aplikasi sistem pakar, pengolahan citra, dan lain-lain.

6. **Administrator Basis Data**

Sebuah lingkup basis data seharusnya mempunyai satu orang atau sekelompok orang pada bagian struktur basis data untuk menangani administrasi basis data yang biasa disebut administrator basis data atau *Database Administrator (DBA)*. *Administrator* basis data adalah orang yang bertanggung jawab dan bekerjasama dengan analisis sistem dan *user-user* lain guna melengkapi berbagai tugas seperti; mendefinisikan data, pemodelan data, desain basis data, serta menjamin keserasian integritas data. (Yakub, 2008).

J. **XAMPP**

Xampp merupakan *software* yang berisi paket pendukung seperti interpreter PHP, Web Server dan data My Sql. Xampp merupakan paket php yang berbasis *open source*. Informasi lengkap mengenai produk ini

dapat diakses di situs resmi websitenya, yaitu: <http://www.apachefriends.com> xampp berfungsi sebagai pengembang aplikasi (*project*) berbasis PHP. Xampp mengkombinasikan beberapa paket software berbeda kedalam satu paket. Paket-paket yang dimaksud adalah Apache, My Sql, PHP, Perl, FileZilla FTP Sever, Php my Admin, Open SSL, Free type, Webalizer, mood_perl, Truck MMChace, mcrypt, SQL Lit, JP Grapt, Mercury Mail Transport Sistem, PHPB lender PHP Compiler (Riyanto;2009).

K. Adobe Dreamweaver CS6

Adobe Dreamweaver merupakan sebuah editor profesional yang menggunakan HTML untuk mendesain web secara visual dan mengelola situs atau halaman web. Adobe Dreamweaver CS6 memiliki beberapa kemampuan yang tidak hanya untuk mendesain web, tetapi juga untuk menyunting kode serta pembuatan aplikasi web antara lain JSP, PHP, ASP, XML, dan ColdFusion (ELCOM, 2013). Software Dreamweaver dikeluarkan oleh Adobe System. Aplikasi ini banyak digunakan oleh para programmer, desainer dan developer web dikarenakan kemudahan dalam penggunaannya, kelengkapan fituranya dan juga dukungannya terhadap teknologi terkini.

Adobe Dreamweaver CS6 memiliki antarmuka pengguna / *user interface* baru serta kemampuan untuk menyunting kode dengan lebih baik. Adobe Dreamweaver CS6 juga dapat melakukan print code pada jendela Code View, serta memiliki fasilitas code hits.

Kelebihan-kelebihan yang dimiliki oleh adobe dreamweaver CS6 adalah sebagai berikut (imulti.org):

1. Kemampuannya membuat halaman web yang terlihat konsisten. Adobe Dreamweaver sudah terinstall beberapa template yang elegan dan menarik.
2. Kemudahan dan efisiensi dalam penggunaan. Adobe Dreamweaver menyediakan fitur editor WYSIWYG (*What You See is What You Get*) atau dalam bahasa kesehariannya disebut Design View. Maksudnya adalah, tampilan hasil akhir web nantinya akan sama dengan tampilan pada saat proses perancangan halaman web. Selain itu dreamweaver memiliki kemampuan memperlihatkan 3 proses yang berbeda, yaitu:
 - a. Code View : Berfungsi untuk hanya menampilkan script html saja.
 - b. Desain View : Berfungsi menampilkan kode-kode html yang tulis menjadi sebuah design / template yang nantinya akan ditampilkan di browser.
 - c. Split View : Berfungsi menampilkan gabungan antara Code View dan Desain View pada saat bersamaan. Sehingga *user* bisa langsung melihat perubahan pada saat anda mengubah htmlnya.
3. Mudah untuk mengupload melalui FTP. Dreamweaver sudah dilengkapi dengan fitur FTP jadi setelah anda selesai membangun sebuah web, anda bisa langsung menguploadnya melalui FTP . FTP (File Transfer Protocol) adalah sebuah protokol Internet yang berjalan di dalam lapisan aplikasi yang merupakan standar untuk pentransferan berkas (file) komputer antar mesin-mesin dalam sebuah internetwork.

4. Dapat dikustom. Dreamweaver dapat disesuaikan dengan kebutuhan yang diperlukan. Menu, tab, perintah, font dan warna semua kode dapat disesuaikan dengan preferensi pribadi. Selain itu *dreamweaver* didukung banyak *plug-in* yang membantu dalam proses desain.

