

## BAB II

### TINJAUAN PUSTAKA

#### A. SDN Jenangan 1

SDN Jenangan 1 berdiri sejak tahun 1916 yang beralamat di Jalan Raya Jenangan No. 173 Desa Jenangan Kecamatan Jenangan Kabupaten Ponorogo. Kepala SDN Jenangan 1 adalah Hj. Tri Wahyuningsih, M.Pd dengan jumlah siswa 130. Disekolah ini memiliki 11 orang guru pengajar dan seorang tenaga kependidikan. SDN Jenangan 1 merupakan sekolah inti karena letaknya yang berda di pusat Kecamatan. Sekolah ini terakreditasi A pada tahun 2015.

1. Visi :

Terwujudnya peserta didik mandiri, berprestasi, berbudaya dan peduli terhadap lingkungan hidup berdasarkan Iman dan Taqwa

2. Misi

- a. Mengembangkan potensi peserta didik dalam rangka membentuk pribadi yang mandiri
- b. Melaksanakan pembelajaran yang kontekstual dan bernuansa paikem
- c. Menanamkan kepedulian sosial, lingkungan, cinta damai dan cinta tanah air
- d. Memberdayakan potensi semua pihak untuk peduli terhadap lingkungan
- e. Melaksanakan budaya hidup bersih dan sehat sebagai wujud pelestarian terhadap lingkungan.

- f. Melaksanakan kegiatan pencegahan terjadinya pencemaran dan kerusakan lingkungan hidup
- g. Menyelenggarakan pendidikan demi terwujudnya pada system nilai adat istiadat, agama dan tetap mengikuti perkembangan iptek

3. Tujuan Sekolah

- a. Meningkatkan kemandirian dan rasa tanggung jawab pada diri siswa melalui kegiatan kepramukaan, keolahragaan, dan ekstra kurikuler
- b. Melaksanakan PBM dengan pendekatan PAIKEM
- c. Membiasakan hidup sehat, kepekaan sosial sehat dalam setiap kegiatan baik di lingkungan sekolah maupun rumah
- d. Menciptakan sekolah yang peduli pada lingkungan hidup;
- e. Mengembangkan model pembelajaran terintegrasi pendidikan lingkungan hidup
- f. Melaksanakan pemilahan dan pengolahan sampah organik dan anorganik
- g. Menanamkan sikap peduli dan berbudaya lingkungan sehingga tercipta lingkungan sekolah yang bersih, sehat, indah, aman dan nyaman
- h. Menyiapkan generasi yang mantap dalam keimanan dan ketakwaan terhadap Tuhan Yang Maha Esa

## B. Pengertian Sistem Informasi

### 1. Pengertian Sistem

Sistem adalah sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan. Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, terkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk tujuan tertentu (Yakub, 2012 : 1). Menurut Jogiyanto (2005) sistem adalah kumpulan elemen-elemen yang berinteraksi untuk mencapai tujuan tertentu.

Dari beberapa pendapat tentang definisi sistem tersebut, definisi sistem yang lebih luas dibandingkan pendekatan sistem yang lebih menekankan pada prosedur dan lebih banyak diterima adalah pendekatan sistem yang merupakan kumpulan dari komponen atau elemen-elemen atau subsistem-subsistem, karena pada kenyataannya suatu sistem memang terdiri dari subsistem-subsistem.

#### 1. Karakteristik Sistem

Karakteristik sistem meliputi :

Komponen Sistem

- a. Batasan Sistem.
- b. Lingkungan Sistem.
- c. Penghubung Sistem.
- d. Masukan Sistem.
- e. Keluaran Sistem.
- f. Pengolahan Sistem.

g. Sasaran Sistem.

## 2. Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya adalah sebagai berikut :

### a. Sistem Abstrak dan Sistem Fisik.

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Sistem fisik merupakan sistem yang ada secara fisik.

### b. Sistem Alamiah dan Sistem Buatan Manusia.

Sistem alamiah adalah sistem yang terjadi karena proses alam tidak dibuat oleh manusia. Sistem buatan manusia adalah sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi manusia dengan mesin disebut dengan *human-machine system* atau ada yang menyebut dengan *man-machine system*.

### c. Sistem Tertentu (*Deterministic System*) dan Sistem Tak Tentu (*Probabilistic System*).

Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Sedangkan sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

### d. Sistem Tertutup dan Sistem Terbuka.

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sedangkan sistem

terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya.

## 2. Pengertian Informasi

Menurut McLoad, 2004 (dalam Yakub, 2012 : 8) mendefinisikan informasi adalah data yang di olah menjadi bentuk lebih berguna dan lebih berarti bagi yang menerimanya. Menurut Sutabri, 2002 (dalam Hamim Tohari, 2007 : 7) informasi merupakan data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan. Menurut Davis, 1999 (dalam Abdul Kadir, 2014 : 45) informasi merupakan data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang.

Kegunaan informasi adalah untuk mengurangi ketidak pastian didalam proses pengambilan keputusan tentang suatu keadaan. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya untuk mendapatkan informasi tersebut.

### 1. Kualitas Informasi

Kualitas informasi (*quality of information*) sangat dipengaruhi atau ditentukan oleh 6 hal berikut :

- a. Relevan (*relevancy*)
- b. Akurat (*accuracy*)
- c. Tepat Waktu (*timeliness*)
- d. Ekonomis (*economy*)
- e. Efisien (*efficiency*)

- f. Dapat dipercaya (*reliability*)
2. Informasi dan Tingkat Manajemen

Berdasarkan tingkatan manajemen, informasi dapat dikelompokkan berdasarkan penggunaannya, yakni sebagai berikut :

a. Informasi Strategis

Digunakan untuk mengambil keputusan jangka panjang, mencakup informasi eksternal (tindakan pesaing, langganan), rencana perluasan perusahaan dan sebagainya.

b. Informasi Taktis

Digunakan untuk mengambil keputusan jangka menengah, mencakup informasi trend penjualan yang dapat dipakai untuk menyusun rencana-rencana penjualan.

c. Informasi Operasional

Digunakan untuk keperluan operasional sehari-hari, informasi persediaan *stock*, retur penjualan dan laporan kas harian.

3. **Pengertian Sistem Informasi**

Sistem informasi adalah suatu cara tertentu untuk menyediakan informasi yang dibutuhkan oleh organisasi untuk beroperasi dengan cara yang sukses dan untuk organisasi bisnis dengan cara yang menguntungkan (Wahyono, 2004:17). Menurut O'Brian, 2005 (dalam Yakub, 2012 : 17) sistem informasi adalah kombinasi teratur dari orang-orang, perangkat keras (*hardware*), perangkat lunak (*software*), jaringan komunikasi, dan sumber daya data yang mengumpulkan, mengubah, dan

menyebarkan informasi dalam sebuah organisasi. Tujuan dari sistem informasi sendiri adalah menghasilkan informasi.

#### 1. Manfaat Sistem Informasi

- a. Organisasi menggunakan sistem informasi untuk mengolah transaksi-transaksi, mengurangi biaya dan menghasilkan pendapatan sebagai salah satu produk atau pelayanan mereka.
- b. Bank menggunakan sistem informasi untuk mengolah cek-cek nasabah dan membuat berbagai rekening koran dan transaksi yang terjadi
- c. Perusahaan menggunakan sistem informasi untuk mempertahankan persediaan pada tingkat paling rendah agar konsisten dengan jenis barang yang tersedia.

### C. Perpustakaan

#### 1. Pengertian, Tujuan, dan Tugas Pokok Perpustakaan

Perpustakaan diartikan sebuah ruangan atau gedung, yang digunakan untuk menyimpan buku dan terbitan lainnya, juga sebagai pusat sumber informasi yang diatur menurut sistem dan aturan yang baku, dan didayagunakan untuk keperluan pendidikan dan penelitian (Basuki, 1991). Perpustakaan secara umum bertujuan untuk menciptakan masyarakat terpelajar dan terdidik, terbiasa membaca, berbudaya tinggi serta mendorong terciptanya pendidikan sepanjang hayat, dan juga melakukan layanan informasi literal kepada masyarakat. Karena



tujuannya memberi layanan informasi literal kepada masyarakat maka tugas pokok adalah:

- a. Menghimpun bahan pustaka yang meliputi buku dan nonbuku sebagai sumber informasi.
- b. Mengolah dan merawat pustaka.
- c. Memberikan layanan bahan pustaka.

## 2. Koleksi Perpustakaan

Beberapa jenis koleksi yang dimiliki oleh perpustakaan adalah:

### a. Buku

Beberapa jenis buku yang dimiliki oleh sebuah perpustakaan sekolah adalah sebagai berikut:

- 1) Buku Teks.
- 2) Buku Penunjang.
- 3) Makalah dan Kliping.

### b. Koleksi Referensi

Isi buku referensi tidak mendalam dan kadang-kadang hanya memuat informasi tertentu saja.

### c. Jenis Serial (Terbitan Berkala)

Pada umumnya terbitan berkala berupa majalah dan koran. Jika dilihat dari isinya majalah dibedakan majalah populer, semi populer dan ilmiah.

d. Brosur yaitu buku atau lembaran-lembaran lepas yang memuat masalah-masalah aktual yang bersifat sementara.

### e. Bahan Pandang Dengar (*Audio Visual*)



Bahan pandang dengan memuat informasi yang dapat ditangkap secara bersama-sama oleh indra mata dan telinga.

### 3. **Klasifikasi Bahan Pustaka dan Penempatan Koleksi**

Koleksi perpustakaan harus diolah dan diatur secara sistematis, dengan tujuan untuk memudahkan penemuan kembali koleksi yang dibutuhkan. Kegiatan pengaturan atau pengelompokan bahan pustaka berdasarkan aturan tertentu disebut dengan klasifikasi.

Tujuan klasifikasi dapat dirinci sebagai berikut:

- a. Menghasilkan urutan yang berguna.
- b. Penempatan yang tepat

Bila bahan pustaka diperlukan pemakai, pustaka yang diinginkan mudah diketemukan serta mudah dikembalikan.

- c. Penyusunan mekanis

Bahan pustaka baru mudah disisipkan diantara bahan pustaka yang sudah dimiliki.

### 4. **Jenis Layanan Perpustakaan**

Beberapa Jenis Layanan Perpustakaan secara umum adalah sebagai berikut:

- a. Layanan peminjaman bahan pustaka (layanan sirkulasi).

Layanan sirkulasi atau layanan peminjaman dan pengembalian bahan pustaka adalah satu kegiatan di perpustakaan yang melayani peminjaman dan pengembalian buku. Kegiatan sirkulasi dapat dilaksanakan sesudah buku-buku selesai diproses dengan lengkap

dengan label-labelnya seperti kartu buku, kartu tanggal kembali, dan *call number* pada punggung buku.

b. Layanan referensi

Koleksi ini tidak boleh dibawa pulang oleh pengunjung perpustakaan dan hanya untuk dibaca ditempat.

c. Layanan ruang baca

Layanan ruang baca adalah layanan yang diberikan oleh perpustakaan berupa tempat layanan untuk melakukan kegiatan membaca di perpustakaan.

## **D. Konsep Analisis dan Perancangan Sistem**

### **1. Pengertian Analisis Sistem**

Menurut McLeod, 1998 (dalam Abdul Kadir, 2014 : 345) analisis sistem adalah suatu proses analisa terhadap masalah yang akan diselesaikan oleh sebuah organisasi dengan menggunakan sistem informasi, yang terdiri dari pendefinisian masalah dan indentifikasi spesifikasi kebutuhan yang harus dipenuhi oleh solusi dari sistem. Sedangkan menurut Alter (1999 : 413) analisa sistem adalah proses umum yang mencakup pendefinisian masalah, pengumpulan informasi yang berkaitan, pengembangan solusi alternatif dan pemilihan diantara solusi tersebut.

## 2. *Flowchart* (Diagram Alur)

### a. 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 :21).

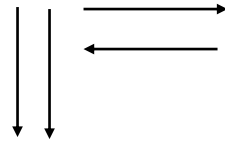
### b. Simbol-simbol *flowchart*.

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 : 21-22)

**Tabel 2.1.** Simbol *Flowchart*

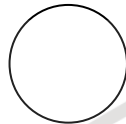
Simbol <i>Flowchart</i>	Fungsi
<i>TERMINAL</i>	
	Simbol ini digunakan untuk mengawali atau mengakhiri suatu proses/kegiatan.
<i>PREPARATION</i>	
	Simbol ini digunakan untuk mempersiapkan harga awal/nilai awal suatu variabel yang akan diproses.
<i>DECISION</i>	
	Simbol ini digunakan untuk pengujian suatu kondisi yang sedang diproses.
<i>PROSES</i>	
	Simbol ini digunakan untuk menggambarkan suatu proses yang sedang dieksekusi.
<i>INPUT/OUTPUT</i>	
	Simbol ini digunakan untuk menggambarkan proses input ( <i>read</i> ) maupun proses output ( <i>print</i> ).
<i>SUBROUTINE</i>	
	Simbol ini digunakan untuk menggambarkan proses pemanggilan subprogram dari 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*



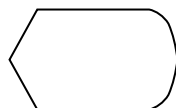
Simbol ini digunakan untuk menggambarkan suatu kegiatan atau proses bersifat manualisasi.

*PRINTER*



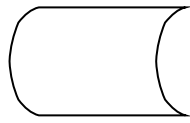
Digunakan untuk menggambarkan suatu kegiatan mencetak suatu informasi dengan mesin printer.

*CONSOLE*



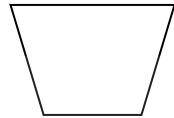
Simbol ini digunakan untuk menggambarkan suatu kegiatan menampilkan data atau informasi melalui monitor atau CRT (*Cathode Ray Tube*).

*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*.

---

#### c. Jenis *flowchart*.

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

##### 1) Program *flowchart*.

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

##### 2) Sistem *Flowchart*.

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

d. 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:

1) *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*).

2) *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 :24).

3. ***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*

Notasi <i>Yourdon</i> <i>DeMarco</i>	Notasi <i>Gane</i> <i>Sarson</i>	Fungsi
		Simbol Entitas eksternal 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

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

a. *Diagram Context.*

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:

- 1) Terminologi sistem :
  - a) Batas Sistem adalah batas antara “daerah kepentingan sistem”.
  - b) Lingkungan Sistem adalah segala sesuatu yang berhubungan atau mempengaruhi sistem tersebut.
  - c) *Interface* adalah aliran yang menghubungkan sebuah sistem dengan lingkungan sistem tersebut.
- 2) 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*).
- 3) Nama/keterangan di simbol proses tersebut sesuai dengan fungsi sistem tersebut.
- 4) Antara *Entitas Eksternal/Terminator* tidak diperbolehkan komunikasi langsung.
- 5) 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 ( # ).

6) Jika Terminator mewakili individu (personil) sebaiknya diwakili oleh peran yang dipermainkan *personil* tersebut.

7) Aliran data ke proses dan keluar sebagai *output* keterangan aliran data berbeda.

b. *DFD* fisik.

*DFD* fisik adalah *representasi* grafik dari sebuah sistem yang menunjukkan *entitas-entitasinternal* dan *eksternal* dari sistem tersebut, dan aliran-aliran data ke dalam dan keluar dari *entitas-entitas* tersebut. *Entitas-entitasinternal* 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.

c. *DFD* logis.

*DFD* logis adalah *representasi* grafik dari sebuah sistem yang menunjukkan proses-proses dalam sistem tersebut dan aliran-aliran

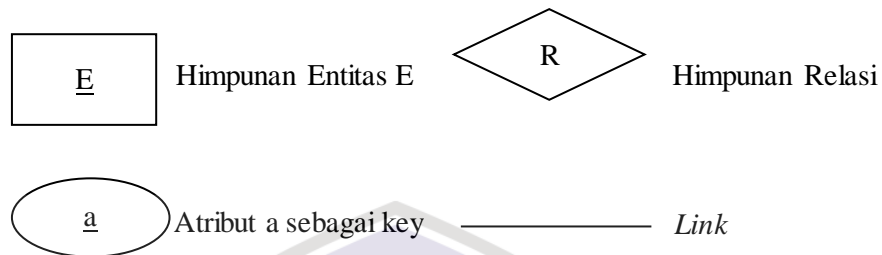
data ke dalam dan ke luar dari proses-proses tersebut. Kita menggunakan *DFD logis* untuk membuat dokumentasi sebuah sistem informasi karena *DFD logis* dapat mewakili logika tersebut, yaitu apa yang dilakukan oleh sistem tersebut, tanpa perlu menspesifikasi dimana, bagaimana, dan oleh siapa proses-proses dalam sistem tersebut dilakukan. Keuntungan dari *DFD logis* dibandingkan dengan *DFD* fisik adalah dapat memusatkan perhatian pada fungsi-fungsi yang dilakukan sistem.

#### 4. *ERD (Entity Relationship Diagram)*

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 *Entity Relationship Diagram (ERD)*. Notasi-notasi simbolik didalam *ERD* yang dapat kita gunakan adalah :

- a. Persegi panjang, menyatakan Himpunan Entitas.
- b. Lingkaran / *Ellips*, menyatakan Atribut (*Atribut* yang berfungsi sebagai *key* digaris bawah).
- c. Belah ketupat, menyatakan Himpunan Relasi.
- d. Garis, sebagai penghubung antara Himpunan Relasi dengan Himpunan Entitas dengan Atributnya.
- e. Kardinalitas relasi dapat dinyatakan dengan banyaknya garis cabang atau dengan pemakaian angka (1 dan 1 untuk relasi satu-ke-satu, 1

dan N untuk relasi satu-ke-banyak atau N dan N untuk relasi banyak-ke-banyak).



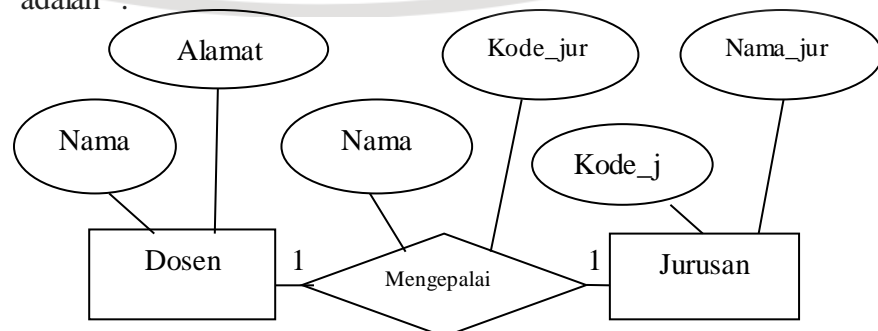
**Gambar 2.1.** Kardinalitas Relasi

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

a. Relasi satu-ke-satu (*one-to-one*)

Contoh :

Adanya relasi antara himpunan entitas Dosen dengan himpunan entitas jurusan. Himpunan relasinya kita beri nama 'Mengepalai' para relasi ini, setiap dosen paling banyak mengepalai satu jurusan (walaupun memang tidak semua dosen yang menjadi ketua jurusan). Dan setiap jurusan pasti dikepalai oleh paling banyak satu orang dosen. Maka penggambarannya adalah :



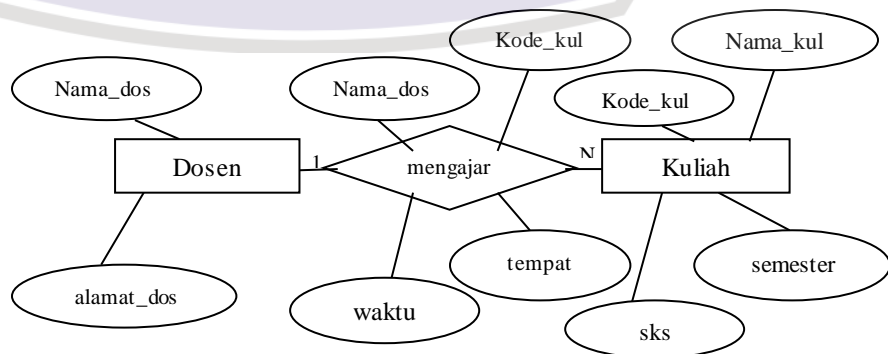
**Gambar 2.2.** ERD untuk Relasi Satu ke Satu

Pada *ERD* diatas dapat kita lihat bahwa himpunan entitas Dosen dan himpunan entitas Jurusan masing-masing memiliki 2 buah atribut saja. Sementara itu, pada himpunan relasi Mengepalai juga terdapat 2 buah atribut yang secara bersama-sama berfungsi juga sebagai *key* pada himpunan relasi tersebut. Karena kedua atribut/*key* tersebut sebetulnya berasal dari atribut *key* dari masing-masing himpunan entitas yang dihubungkannya, maka keduanya digolongkan sebagai *Key Asing (Foreign-Key)*. Adanya kedua atribut *key* tersebut selanjutnya akan dapat menunjukkan dosen mana yang mengepalai suatu jurusan (menjadi ketua jurusan), atau sebaliknya jurusan mana yang dikepalai seorang dosen.

b. Relasi satu-kebanyak (*one-to-many*).

Contoh :

Adanya relasi antara himpunan entitas Dosen dengan himpunan entitas Kuliah. Himpunan relasinya kita beri nama 'Mengajar'. Pada relasi ini, setiap dosen dapat mengajar lebih dari satu mata kuliah, sedang setiap mata kuliah diajar hanya oleh paling banyak satu orang dosen. Maka penggambaran adalah :



**Gambar 2.3.** *ERD* untuk Relasi Satu ke Banyak



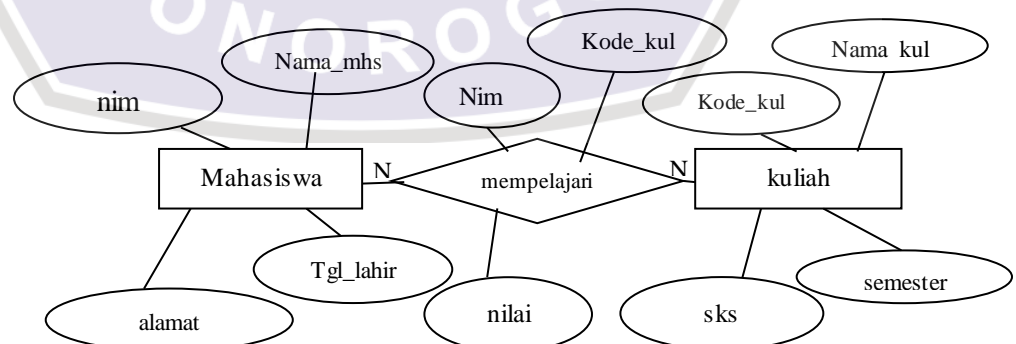
*Key Asing (Foreign-Key)* dari himpunan relasi Mengajar diatas adalah nama\_dos dan kode\_kul, yang masing-masing berasal dari himpunan entitas Dosen dan himpunan entitas kuliah. Tetapi disamping kedua atribut *key* tersebut, ada 2 atribut tambahan yang tidak berasal dari salah satu himpunan entitas yang dihubungkannya. Hal ini memang memungkinkan dan bahkan umum terjadi. Dengan adanya atribut tersebut pada himpunan relasi Mengajar, maka dapat kita ketahui jadwal pelaksanaan setiap mata kuliah beserta dosen yang mengajarkannya.

c. Relasi banyak-ke-banyak (*many-to-many*).

Contoh :

Adanya relasi himpunan entitas Mahasiswa dengan himpunan entitas Kuliah. Himpunan relasinya kita beri nama 'Mempelajari'.

Pada relasi ini, setiap mahasiswa dapat mempelajari lebih dari satu mata kuliah. Demikian juga sebaliknya, setiap mata kuliah dapat dipelajari oleh lebih dari 1 orang mahasiswa. Maka penggambarannya adalah :



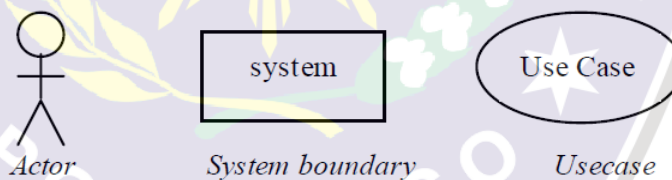
**Gambar 2.4.** ERD untuk Relasi Banyak ke Banyak



Keberadaan himpunan relasi Mempelajari diatas akan memiliki 2 fungsi, yaitu menunjukkan mata kuliah mana saja yang diambil oleh seorang mahasiswa (atau mahasiswa mana saja yang mengambil mata kuliah tertentu) dan indeks nilai yang di peroleh seseorang mahasiswa untuk mata kuliah tertentu (tentu saja setelah data indeks nilai tersebut disimpan). (Fathansyah,1999 :71-72).

### 5. Use Case Diagram

Menurut Schmuller,1999 (dalam Sugiarti, 2013 : 4)Use case adalah sebuah gambaran dari fungsi sistem yang dipandang dari sudut pandang pemakai. *Actor* adalah segala sesuatu yang perlu berinteraksi dengan sistem untuk pertukaran informasi. *System boundary* menunjukkan cakupan dari sistem yang dibuat dan fungsi dari sistem tersebut. Berikut ini merupakan gambar dari tiga komponen sistem dalam *use case diagram*:



**Gambar 2.5** Komponen-Komponen *Use Case*.

Jenis-jenis *use case relationships* antara lain :

a. **Association**

Garis yang menghubungkan antara *actor* dengan *use case*.

b. **Extend**

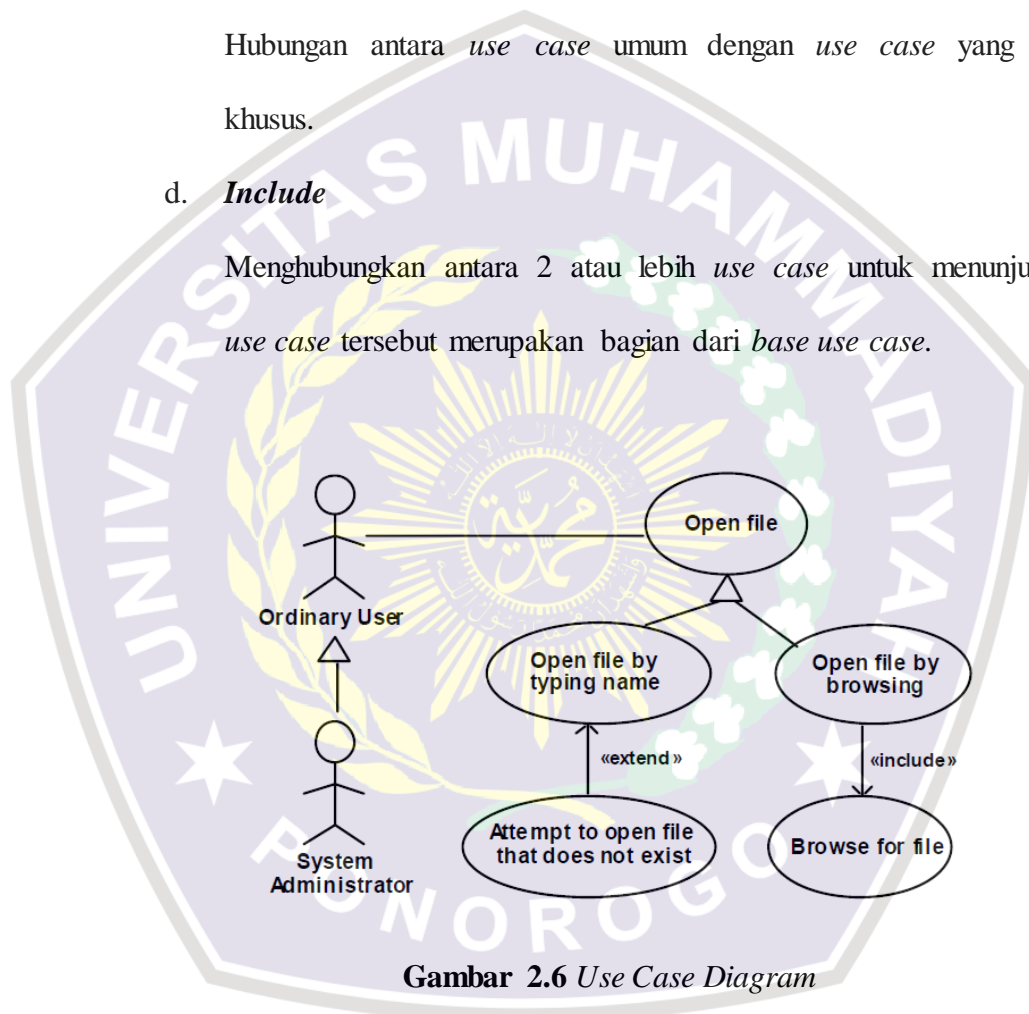
Menghubungkan antara dua atau lebih *use case* yang merupakan tambahan dari *base use case* yang biasanya untuk mengatasi kasus pengecualian.

c. **Generalization**

Hubungan antara *use case* umum dengan *use case* yang lebih khusus.

d. **Include**

Menghubungkan antara 2 atau lebih *use case* untuk menunjukkan *use case* tersebut merupakan bagian dari *base use case*.



Gambar 2.6 Use Case Diagram

## 6. Basis Data

a. 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.

e. Manfaat Basis Data

- 1) Kecepatan dan kemudahan (*Speed*), pemanfaatan basis data memungkinkan untuk dapat, menyimpan, merubah, dan menampilkan kembali data tersebut dengan lebih cepat dan mudah.
- 2) Efisiensi ruang penyimpanan (*space*), dengan basis data efisiensi atau optimalisasi penggunaan ruang penyimpanan dapat dilakukan, karena penekanan jumlah redundansi data, baik dengan sejumlah pengkodean atau dengan membuat tabel-tabel yang saling berhubungan.
- 3) Keakuratan (*accuracy*), pembentukan relasi antar data bersama dengan penerapan aturan atau batasan (*constraint*) tipe, domain dan keunikan data dapat diterapkan dalam sebuah basis data.
- 4) 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.

5) 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.

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)

f. 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 :

- 1) Pembuatan basis data baru (*CREATE DATABASE*)
- 2) Penghapusan basis data (*DROP DATABASE*)
- 3) Pembuatan *file* atau tabel baru ke suatu basis data (*CREATE TABLE*)
- 4) Penghapusan *file* atau tabel dari suatu basis data (*DROP TABLE*)
- 5) Penambahan atau pengisian data baru di sebuah basis data (*INSERT*)

- 6) Pengambilan data dari sebuah *file* atau tabel (*RETRIEVE* atau *SEARCH*)
- 7) Pengubahan data dalam sebuah *file* atau tabel (*UPDATE*)
- 8) 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)

#### d. 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*.

##### 1) *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.

##### 2) 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).

3) 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.

4) 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.

5) *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).

e. Sistem Basis Data

1) 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)

2) 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*).

3) *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.

#### 4) 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.

##### a) *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).

##### b) *User Mahir (Casual User)*

*Casualuser* 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.

##### c) *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.

d) *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.

f. *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 bekerja sama 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)

## 7. *MySQL*

Menurut Nugroho (2005), *MySQL* adalah sebuah program *data base server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, *multi user* serta menggunakan perintah standar *SQL (Structured Query Language)*. *MySQL* merupakan *software* yang tergolong sebagai *DBMS (Database Managemen Sistem)* yang bersifat *Open Source*. *Open Source* menyatakan bahwa software ini dilengkapi dengan *source code* (kode yang dipakai untuk membuat *MySQL*), selain tentu saja bentuk *executable*-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi, dan bisa dengan cara *men-download* (mengunduh) di internet secara gratis.

*MySQL* awalnya dibuat oleh perusahaan konsultan bernama *TcX* yang berlokasi di Swedia. Saat ini pengembangan *MySQL* berada dibawah naungan perusahaan *MySQL AB*.

Sebagai software *DBMS*, *MySQL* memiliki sejumlah fitur seperti yang dijelaskan dibawah ini.

a. *Multiplatform*.

*MySQL* tersedia pada beberapa *platform* (*Windows*, *Linuk*, *Unix*, dan lain-lain).

b. Handal, cepat, dan mudah digunakan.

*MySQL* tergolong sebagai *database server* (*server* yang melayani permintaan terhadap *database*) yang handal, dapat menangani *database* yang besar dengan kecepatan tinggi, mendukung banyak sekali fungsi untuk mengakses *database*, dan sekaligus mudah untuk digunakan. Berbagai *tool* pendukung juga tersedia (walaupun dibuatoleh pihak lain). Perlu diketahui, *MySQL* dapat menangani sebuah table yang berukuran dalam *Terabyte* (1 *Terabyte* = 1024 *Gigabyte*). Namun, ukuran yang sesungguhnya sangat bergantung pada batasan sistem operasi. Sebagai contoh pada system *solaris 9/10*, batasan ukuran *file* sebesar 16 *Terabyte*.

c. Jaminan keamanan akses.

*MySQL* mendukung pengamanan *database* dengan berbagai kriteria pengaksesan. Sebagai gambaran, dimungkinkan untuk mengatur *user* tertentu agar bisa mengakses data yang bersifat rahasia (misalnya gaji pegawai), sedangkang *user* lain tidak boleh.

*MySQL* juga mendukung konektivitas ke berbagai *software*. Sebagai contoh, dengan menggunakan *ODBC (Open Database Connectivity)*, *database* yang ditangani *MySQL* dapat diakses melalui program yang dibuat dengan *Visual Basic*. *MySQL* juga mendukung program *klien* melalui *JDBC (Java Database Connectivity)*. *MySQL* juga bisa diakses melalui aplikasi berbasis *Web* : misalnya dengan menggunakan *PHP*.

d. Dukungan *SQL*.

Seperti tersirat dalam namanya, *MySQL* mendukung perintah *SQL (Structured Query Language)*. Sebagai mana diketahui, *SQL* merupakan standart dalam pengaksesan *database relasional*. Pengetahuan akan *SQL* akan memudahkan siapa pun untuk menggunakan *MySQL*. (Kadir,2008 : 2-3).

## 8. *PHP (Hypertext Preprocessor)*

### a. Pengertian *PHP (Hypertext Preprocessor)*

*PHP (Hypertext Preprocessor)* adalah bahasa *server side scripting* yang menyatu dengan *HTML* untuk membuat halaman *web* yang dinamis (Arief, 2011:43). 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 :

- 1) *PHP* memiliki tingkat akses yang lebih cepat
- 2) *PHP* mampu berjalan di beberapa *server web*, seperti *Apache, Microsoft IIS*, dan *Xitami*
- 3) *PHP* mendukung akses ke beberapa *database* baik yang bersifat gratis atau komersial, seperti *mSQL, MySQL*, dan *MicrosoftSQL Server*
- 4) *PHP* bersifat *Open Source*

#### **b. Sintaks *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*.

#### 9. *Adobe Dreamweaver CS6*

*Adobe Dreamweaver CS6* merupakan sebuah editor profesional yang menggunakan *HTML* untuk mendesain *web* secara visual dan mengelola situs atau halaman *web* (Elcom, 2013 : 1). Salah satu kelebihan *Adobe Dreamweaver CS6* yaitu ruang kerja *Adobe Dreamweaver CS6* beserta *tools* yang tersedia dapat digunakan dengan sangat mudah dan cepat sehingga anda bisa membangun suatu *website* dengan cepat dan tanpa harus melakukan *coding*. Selain itu, *Adobe Dreamweaver CS6* juga mempunyai integrasi dengan produk *macromedia* lainnya, seperti *flash* dan *firework*, *flash* sudah sangat terkenal sebagai program untuk membuat animasi yang berbasis *web* dengan perkembangan kebutuhan dan teknologi, *flash* akhir-akhir ini juga digunakan untuk membuat animasi dan video.