

## BAB II

### TINJAUAN PUSTAKA

#### A. Tugas Dan Fungsi BPPKAD Kabupaten Ponorogo

- 1) Badan merupakan unsur penunjang urusan pemerintahan di bidang pendapatan, pengelolaan keuangan dan asset daerah yang dipimpin oleh Kepala Badan yang berkedudukan di bawah dan bertanggungjawab kepada Bupati melalui Sekretaris Daerah.
- 2) Badan mempunyai tugas membantu Bupati dalam melaksanakan fungsi penunjang Urusan pemerintahan yang menjadi kewenangan Kabupaten di bidang pengelolaan keuangan, pendapatan dan asset daerah.
- 3) Dalam melaksanakan tugas sebagaimana dimaksud pada ayat (2), Badan menyelenggarakan fungsi:
  - a. penyusunan kebijakanteknis dibidang pengelolaan keuangan, pendapatan dan asset daerah;
  - b. pelaksanaantugasdukunganteknisdibidang pengelolaan keuangan, pendapatan dan asset daerah;
  - c. pemantauan, evaluasi, dan pelaporan pelaksanaan tugas dukungan teknis dibidang pengelolaan keuangan, pendapatan dan asset daerah;
  - d. pembinaanteknispenyelenggaraanfungsi-fungsi penunjang Urusan Pemerintahan Daerah dibidang pengelolaan keuangan, pendapatan dan asset daerah; dan
  - e. pelaksanaan fungsi lain yang diberikan oleh bupati sesuai dengan tugas dan fungsinya.

2. Dalam melaksanakan tugas dan fungsi sebagaimana dimaksud pada ayat (2) dan ayat (3), Badan mempunyai kewenangan:
- a. Fasilitasi dan koordinasi perumusan perda tentang pokok-pokok pengelolaan keuangan daerah;
  - b. Perencanaan anggaran penanganan urusan pemerintahan kabupaten;
  - c. Fasilitasi dan koordinasi perumusan perda tentang APBD dan perubahan APBD;
  - d. Fasilitasi dan koordinasi pelaksanaan kebijakan pendanaan urusan pemerintahan yang menjadi tanggung jawab bersama (urusan concurrent) antara kabupaten dan desa;
  - e. Pelaksanaan pengelolaan pajak dan retribusi daerah;
  - f. Pembinaan dan pengawasan pajak dan retribusi daerah;
  - g. Pelaksanaan kebijakan pengelolaan investasi dan aset daerah;
  - h. Pengawasan pengelolaan investasi dan aset daerah;
  - i. Fasilitasi pengelolaan aset daerah pemekaran;
  - j. Fasilitasi dan koordinasi pelaksanaan pengelolaan Badan Usaha Milik Daerah (BUMD) dan lembaga keuangan mikro kabupaten;
  - k. Pelaksanaan kebijakan pengelolaan pinjaman dan obligasi daerah, serta Badan Layanan Umum (BLU) kabupaten;
  - l. Pengawasan pinjaman dan obligasi daerah, serta Badan Layanan Umum (BLU) kabupaten;
  - m. pengelolaan data dasar penghitungan alokasi Dana Alokasi Umum (DAU) kabupaten;

- n. Pengelolaan Dana Alokasi Umum (DAU) kabupaten;
- o. Pelaporan pengelolaan Dana Alokasi Umum (DAU) kabupaten;
- p. Fasilitasi dan koordinasi usulan program dan pengelolaan Dana Alokasi Khusus (DAK);
- q. Pengendalian dan pelaporan pengelolaan Dana Alokasi Khusus (DAK);
- r. Penyiapan data realisasi penerima Dana Bagi Hasil (DBH) kabupaten;
- s. Pengendalian dan pelaporan pengelolaan Dana Bagi Hasil (DBH);
- t. Fasilitasi dan koordinasi penetapan kebijakan teknis tentang sistem dan prosedur akuntansi pengelolaan keuangan daerah;
- u. Penyusunan laporan keuangan dan pertanggungjawaban pelaksanaan Anggaran Pendapatan dan Belanja Daerah;
- v. Perumusan laporan keuangan dan pertanggungjawaban pelaksanaan pendanaan urusan pemerintahan yang menjadi tanggung jawab bersama (urusan *concurrent*).

## **B. Pengertian Website**

*Website* adalah kumpulan dari beberapa halaman web dimana informasi dalam bentuk teks, gambar, suara, dan lain-lain dipersentasikan dalam bentuk hypertext dan dapat diakses oleh perangkat lunak yang disebut dengan browser. Informasi pada sebuah website pada umumnya di tulis dalam format *HTML*. Informasi lainnya disajikan dalam bentuk grafis (dalam format *GIF, JPG, PNG, dll*), suara (dalam format *AU, WAV, dll*), dan objek multimedia lainnya (seperti *MIDI, Shockwave Quicktime Movie, 3D World, dll*).

*Website* merupakan fasilitas *internet* yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen pada *website* disebut dengan web page dan link dalam *website* memungkinkan pengguna bisa berpindah dari satu *page* ke *page* lain (*hyper text*), baik diantara *page* yang disimpan dalam *server* yang sama maupun *server* diseluruh dunia. *Pages* diakses dan dibaca melalui *browser* seperti *NetscapeNavigator* atau *InternetExploler* berbagai aplikasi *browser* lainnya. (Hakim Lukmanul, 2004)

### C. Pengertian Hirarki Website

Hirarki *website* adalah stuktur *link* dan *page* pada suatu *website*, yang digambarkan dalam bentuk *tree*, hirarki *website* biasanya juga memperhatikan sistematika dan konsistensi sebuah *website*. Penelusuran dan navigasi mudah dilakukan jika *website* memiliki stuktur hirarki yang baik. (Supriansyah, Haris, 2006).

### D. Pengertian Sistem Informasi

Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan, Tata Sutabri (2005).

Menurut Jogiyanto (2005) Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi

dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. Komponen sistem informasi terdiri dari beberapa hal sebagai berikut:

1. Blok Masukan

Input mewakili data yang masuk ke dalam sistem informasi. Input disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukan, yang dapat berupa dokumen-dokumendasar.

2. Blok Model

Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi

Teknologi merupakan “kotak alat” (*tool-box*) dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan.

## 5. Blok Basis Data

Basis Data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

## 6. Blok Kendali

Banyak hal yang dapat merusak sistem informasi, seperti misalnya bencana alam, api, temperature, air, debu, kecurangan-kecurangan, kegagalan-kegagalan sistem itu sendiri, kesalahan-kesalahan, ketidak efisienan, sabotase dan lain sebagainya.

## E. Pengertian *Internet*

*Internet* yang merupakan kependekan dari *Interconnection Networking* adalah kumpulan komputer yang saling terhubung dan membentuk sistem jaringan melalui jalur telekomunikasi seperti telepon, radio link, satelit dan lainnya. (Slamin, Dkk, 2007)

Menurut Akhmad Fauzi (2008) *Internet* memiliki beberapa kelebihan dibanding telepon atau media komunikasi lainnya, dari segi biaya *internet* tidak memperhitungkan jarak seperti percakapan di telepon. fasilitas-fasilitas *internet* memiliki beberapa tujuan, yaitu:

1. Sebagai media melakukan transfer *file*
2. Sebagai sarana mengirim surat (*e-mail*)
3. Melakukan *mailing list*, *newgroup* dan konferensi
4. *Chatting*
5. Mesin pencari (*Search Engine*)

6. Untuk mengirim *SMS* ke telepon seluler
7. Sarana entertainmen dan permainan

## F. *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, 2011:43). 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:

- 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**

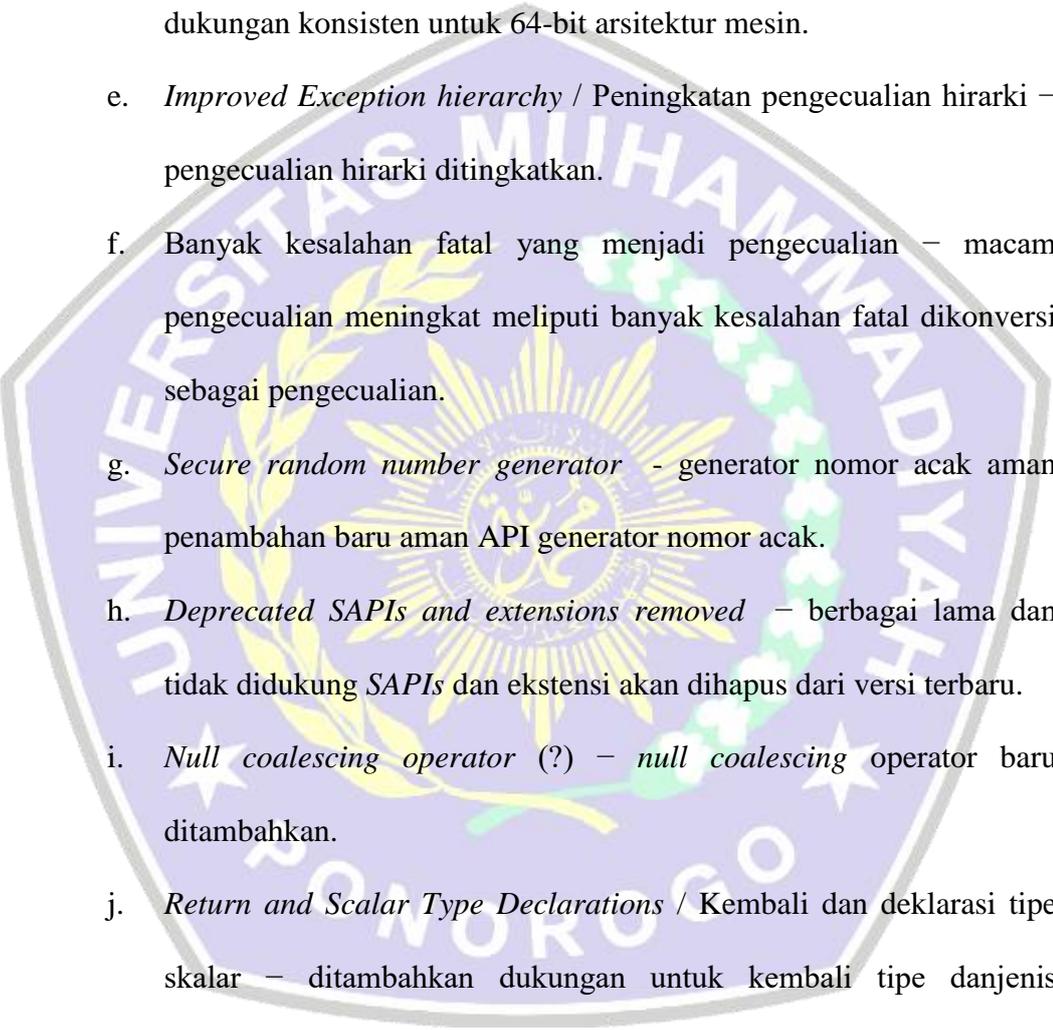
Sintaks *PHP* ditulis dalam apitan tanda khusus *PHP*. Ada empat macam pasangan tag *PHP* yang digunakan:

- a. `<?PHP....?>`
- b. `<script language="PHP">...</script>`
- c. `<?....?>`
- d. `<%....%>`

*Pointa* dan *b* merupakan cara yang paling umum yang digunakan sekalipun cara *c* lebih praktis karena cara *c* tidak selalu diaktifkan pada konfigurasi *filePHP* 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*.

## 3. Fitur baru pada PHP 7

- a. *Improved Performance* – *PHPNG* memiliki kode bergabung di *PHP7*, sudah dua kali secepat *PHP 5*.

- 
- b. *Lower Memory Consumption* - Pemakaian memori rendah dioptimalkan PHP 7 memanfaatkan sumber daya yang lebih rendah.
  - c. *Scalar type declarations* / Tipe skalar Deklarasi – sekarang parameter dan kembali jenis dapat dilaksanakan.
  - d. *Consistent 64-bit support* / Konsisten pada dukungan 64-bit – dukungan konsisten untuk 64-bit arsitektur mesin.
  - e. *Improved Exception hierarchy* / Peningkatan pengecualian hirarki – pengecualian hirarki ditingkatkan.
  - f. Banyak kesalahan fatal yang menjadi pengecualian – macam pengecualian meningkat meliputi banyak kesalahan fatal dikonversi sebagai pengecualian.
  - g. *Secure random number generator* - generator nomor acak aman penambahan baru aman API generator nomor acak.
  - h. *Deprecated SAPIs and extensions removed* – berbagai lama dan tidak didukung *SAPIs* dan ekstensi akan dihapus dari versi terbaru.
  - i. *Null coalescing operator (?)* – *null coalescing* operator baru ditambahkan.
  - j. *Return and Scalar Type Declarations* / Kembali dan deklarasi tipe skalar – ditambahkan dukungan untuk kembali tipe dan jenis parameter.
  - k. *Anonymous Classes* / Anonim kelas – dukungan untuk anonim ditambahkan.
  - l. *Zero cost asserts* / Biaya nol menegaskan – dukungan untuk menyatakan biaya nol ditambahkan.

## **G. *HyperText Markup Language (HTML)***

*HyperText Markup Language (HTML)* adalah sebuah bahasa yang digunakan untuk membuat sebuah halaman *web*, menampilkan berbagai informasi di dalam sebuah penjelajah *web Internet* dan pemformatan hiperteks sederhana yang ditulis dalam berkas format *ASCIL* agar dapat menghasilkan tampilan wujud yang terintegrasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan dalam format *ASCIL* normal sehingga menjadi halaman *web* dengan perintah – perintah *HTML*. Bermula dari sebuah bahasa sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan *SGML (Standart Generalized Markup Language)*, *HTML* adalah sebuah standart yang digunakan secara luas untuk menampilkan halaman *web*. *HTML* saat ini merupakan standar internet yang didefinisikan dan dikendalikan penggunaannya oleh *Word Wide web Consortium (W3C)*. *HTML* dibuat oleh kolaborasi *Caillau TIM* dengan *Berners-lee Robert* ketika mereka bekerja di *CERN* pada tahun 1989 (*CERN* adalah lembaga penelitian fisika energi tinggi di Jenawa).

## **H. Basis Data**

### **1. Pengertian Basis Data**

Basis Data adalah kumpulan data yang saling berhubungan secara logikal serta deskripsidari data tersebut, yang dirancang untuk memenuhi kebutuhan informasi suatu organisasi. Basis Data adalah sebuah

penyimpanan data yang besar yang bisa digunakan oleh banyak pengguna dan departemen. Semua data terintegrasi dengan jumlah duplikasi yang minimum. Basis Data tidak lagi dipegang oleh satu departemen, tetapi dibagikan ke seluruh departemen pada perusahaan. Basis Data itu sendiri tidak hanya memegang data operasional organisasi tetapi juga penggambaran dari data tersebut (Connolly & Begg, 2010).

Basis data adalah kumpulan data store yang terintegrasi yang diatur dan di kontrol secara sentral. Sebuah basis data biasanya menyimpan ribuan *class*. Informasi yang disimpan termasuk *class* attribute dan relasi antar *class*. Basis data juga menyimpan informasi yang deksriptif seperti nama atribut, pemberian batasan suatu nilai, dan kontrol akses untuk data-data yang sensitif (Satzinger, Robert, & Stephen, 2005: 398). Basis data juga diartikan sebagai sekumpulan *file* dikomputer yang saling terhubung. *File-file* ini diatur sesuai kesamaan elemennya, sehingga data yang diinginkan dapat dicari secara mudah (Williams & Sawyer, 2007: 181).

Basis data adalah dua atau lebih simpanan data dengan elemen-elemen data penghubung, yang dapat diakses lebih dari satu cara. Basis data dinyatakan dengan tehnik-tehnik formal dan manajemen basis data. Dari definisi diatas, maka dapat dikatakan bahwa basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya (Iskandar & Rangkuti, 2008). Basis data adalah dua atau lebih simpanan data dengan elemen-elemen data penghubung, yang dapat

diakses lebih dari satu cara. Basis data dinyatakan dengan tehnik-tehnik formal dan manajemen basis data (Abdillah, 2012).

Dapat disimpulkan basis data adalah penyimpanan data yang terstruktur, terintegrasi dan saling berkaitan dengan elemen-elemen penghubungnya dan dapat di akses dengan berbagai cara, oleh karena itu basis data juga bisa didefinisikan sebagai kumpulan yang menggambarkan sendiri dari catatan yang terintegrasi dan penggambaran dari data dikenal sebagai sistem katalog (atau kamus data atau metadata). Definisi data disini dibedakan dari program aplikasi, yang umumnya sama dengan pendekatan pengembangan modern perangkat lunak, dimana definisi internal dan eksternal dari sebuah objek dipisahkan. Salah satu keuntungan dari pendekatan tersebut adalah abstraksi data dimana kita dapat mengubah definisi internal dari sebuah objek tanpa mempengaruhi pengguna dari objek jika definisi eksternal objek tersebut tidak berubah.

## **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 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
- b. Pengaksesan Data
- c. Data Terisolasi untuk Standarisasi
- d. Masalah Keamanan atau *Security*
- e. *MultipleUser*

## I. *Database Management System (DBMS)*

### 1. **Pengertian DBMS**

*Database Management System* adalah sistem perangkat lunak yang memungkinkan pengguna dapat mendefinisikan, membuat, merawat, dan mengatur akses ke Basis Data. Biasanya *DBMS* memungkinkan pengguna untuk mendefinisikan *database* melalui sebuah *Data Definition Language (DDL)*, menspesifikasikan tipe data, struktur dan batasan pada data yang disimpan pada *database*. Kemudian juga memungkinkan insert, update, delete, dan mengambil data dari *database* melalui *Data Manipulation Language (DML)*, mempunyai pusat penyimpanan untuk semua data dan deskripsi data memungkinkan DML untuk menyediakan fasilitas umum untuk data tersebut yang umumnya disebut bahasa *query* (Connolly & Begg, 2010).

### 2. **Keuntungan DBMS**

Berikut ini adalah beberapa keuntungan dari Database Management System, yakni:

#### a. Kontrol terhadap redundansi data

Sistem berbasis file tradisional membuang tempat penyimpanan dengan menyimpan informasi yang sama lebih dari satu file.

#### b. Konsistensi data

Dengan menghilangkan atau mengendalikan redundansi, kita bisa mengurangi resiko dari inkonsistensi yang akan terjadi. Apabila sebuah item data disimpan hanya sekali dalam Database, jika terjadi pembaruan pada nilainya yang harus dilakukan hanya sekali maka

nilai yang baru tersebut akan langsung bisa digunakan untuk semua pengguna.

- c. Lebih banyak informasi dari sumber yang sama.

Dengan integrasi dari data operasional, memungkinkan bagi organisasi untuk mengambil data tambahan dari informasi yang sama.

- d. Pembagian Data

Biasanya file dimiliki oleh departemen atau yang menggunakannya. Dilain hal, Database seharusnya berada diseluruh organisasi dan bisa di-share pada seluruh pengguna yang diizinkan.

- e. Meningkatkanintegritas data

Integritas Database mengacu pada validitas dan konsistensi data yang tersimpan.

- f. Meningkatkan keamanan

Keamanan Database adalah perlindungan dari Database dari pengguna yang tidak sah. Tanpa langkah-langkah keamanan yang sesuai, integrasi membuat data lebih rentan daripada sistem Basis Data.

- g. Penegakan Standar

Integrasi memungkinkan DBA untuk mendefinisikan dan menegakan standar yang diperlukan. Termasuk departemen, organisasi, standar nasional, ataustandar internasional untuk hal-hal seperti format data untuk memfasilitasi pertukaran data antara

sistem, konvensi penamaan, standar dokumentasi, prosedur update, dan aturan akses.

h. Skala Ekonomi

Menggabungkan semua data operasional organisasi ke dalam satu Database, dan menciptakan kumpulan aplikasi yang bekerja pada salah satu sumber data, yang berdampak pada penghematan biaya.

i. Keseimbangan pada persyaratan yang bertentangan

Setiap pengguna atau departemen memiliki kebutuhan yang mungkin bertentangan dengan kebutuhan pengguna lain. Karena Database berada di bawah kendali DBA, DBA dapat membuat keputusan mengenai penggunaan desain dan operasional dari Database yang menyediakan penggunaan terbaik dari sumber daya bagi organisasi secara keseluruhan.

j. Meningkatkan aksesibilitas data dan data responsif

Sebagai akibat dari integrasi, data yang melintasi batas-batas departemen menjadi dapat diakses secara langsung ke pengguna akhir. Dengan demikian menyediakan suatu sistem dengan potensi yang lebih banyak mengenai fungsionalitas, misalnya, dapat digunakan untuk memberikan layanan yang lebih baik kepada pengguna akhir atau klien organisasi.

k. Peningkatan Produktifitas

DBMS menyediakan banyak fungsi standar yang biasanya seorang programmer harus menulis dalam aplikasi berbasis file. Pada tingkat dasar, DBMS menyediakan semua rutinitas penanganan file tingkat

rendah yang khas dalam program aplikasi. Penyediaan dari fungsi tersebut memungkinkan programmer untuk berkonsentrasi pada fungsi khusus yang diperlukan oleh pengguna tanpa harus khawatir tentang detail implementasi tingkat rendah.

l. Peningkatan pemeliharaan melalui independensi data

Dalam sistem berbasis file, deskripsi data dan logika untuk mengakses data dibangun ke dalam setiap program aplikasi, menjadikan program berketergantungan pada data. Perubahan struktur data, misalnya membuat alamat 41 karakter bukan 40 karakter, atau perubahan dengan bagaimana cara data disimpan pada disk, memerlukan perubahan besar untuk program yang terpengaruh oleh perubahan.

m. Peningkatan Konkurensi

Dalam beberapa sistem berbasis file, jika dua atau lebih pengguna diizinkan untuk mengakses file yang sama secara bersamaan, sangat mungkin akses tersebut saling mengganggu satu sama lain, sehingga dapat menyebabkan hilangnya informasi atau bahkan hilangnya integritas.

n. Peningkatan backup dan Jasa Pemulihan

Banyak system berbasis file menempatkan tanggung jawab pada pengguna untuk memberikan langkah-langkah untuk melindungi data dari kerusakan pada sistem komputer atau program aplikasi. Ini mungkin melibatkan backup dari data semalaman. Jika terjadi kerusakan di keesokan harinya, backup dipulihkan dan pekerjaan

yang telah dikerjakan sebelum backup ini hilang dan harus kembali dikerjakan. (Connolly & Begg, 2010: 77-80)

## J. *MySQL*

*MySQL* adalah sebuah perangkat lunak sistem manajemen basis data *SQL* atau *DBMS* yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi diseluruh dunia. *MySQL* merupakan *server* basis data dimana pemrosesan data terjadi di *server*, dan *client* hanya mengirimkan data serta meminta data. Oleh karena pemrosesan terjadi di *server* sehingga pengaksesan data tidak terbatas. (Solihin, 2010)

*MySQL* adalah *multiuser database* yang menggunakan bahasa *Structured Query Language* (*SQL*). *MySQL* dalam operasi *client-server* melibatkan *server daemon MySQL* disisi *server* dan berbagai macam program serta *library* yang berjalan disisi *client*. *MySQL* mampu menangani data yang cukup besar. Dengan menggunakan *SQL*, proses pengaksesan database lebih *user-friendly* dibandingkan dengan yang lain, misalnya *Base* atau *Clipper* karena mereka masih menggunakan perintah-perintah pemrograman murni, perintah yang digunakan biasanya adalah: *INSERT*, *DELETE*, *UPDATE*, dan *SELECT*.

### 1. *INSERT*

Menambah baris pada tabel. *Syntax* yang paling sering digunakan:  
*INSERT INTO* namatabel *VALUES* (nilai1, nilai2, nilai-n);

### 2. *DELETE*

Menghapus baris pada tabel. *Syntax*: *DELETE FROM* namatabel [*where* kondisi];

3. *UPDATE*

Mengubah isi beberapa kolom pada tabel. *Syntax: UPDATE* namatabel

*SET* kolom1=nilai1, kolom2=nilai2 [where kondisi];

4. *SELECT*

Menampilkan isi dari suatu tabel yang bisa dihubungkan dengan tabel yang lainnya;

Keunggulan *MySQL*

1. Program yang *multi-threaded*, sehingga dapat dipasang pada *server* yang memiliki *multi – CPU*.
2. Didukung bahasa pemrograman umum seperti *C, C++, Java, Perl, PHP, Python, TCL, APls* dls.
3. Bekerja pada berbagai *platform*
4. Memiliki jenis kolom yang cukup banyak sehingga memudahkan konfigurasi *system database*
5. Memiliki *system* sekuriti yang cukup baik dengan verifikasi *host*
6. Mendukung *ODBC* untuk *OS Microsoft Windows*
7. Mendukung *record* yang memiliki kolom dengan panjang tetap
8. *Software* yang *free*
9. Saling terintegrasi dengan *PHP*.

Dilihat dari kelebihan-kelebihan yang dimiliki oleh program *databaseMySQL*.

## K. *Flowchart* (Diagram Alur)

### 1. Pengertian *Flowchart* (Diagram Alur).

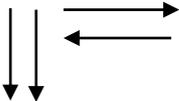
Karena komputer membutuhkan hal-hal yang rinci, maka bahasa pemrograman bukanlah alat baik untuk merancang sebuah *algoritma* awal. Alat yang banyak dipakai untuk membuat *algoritma* adalah diagram alur (*flowchart*).

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*.

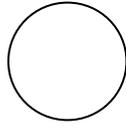
Sudah dikemukakan di atas bahwa diagram alur atau *flowchart* memiliki beberapa simbol yang biasa digunakan untuk menggambarkan rangkaian proses yang harus dilaksanakan. Simbol-simbol tersebut dijelaskan di bawah ini: (Sutabri, 2004)

**Tabel 2.1.** Simbol *Flowchart*

Simbol	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.
<i>FLOW LINE</i>	
	Simbol ini digunakan untuk menggambarkan arus proses dari suatu kegiatan ke kegiatan lain.

---

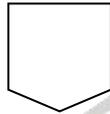
*CONNECTOR*



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*

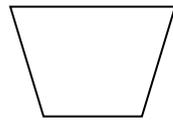


Simbol ini digunakan untuk menggambarkan suatu kegiatan membaca atau menulis data menggunakan media *magnetic disk*.

---

---

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*.

---

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 detail antara intruksi yang satu dengan intruksi yang lainnya dalam suatu program komputer yang bersifat *logik*.

b. Sistem *Flowchart*.

Simbol-simbol yang menggambarkan urutan prosedur secara detail 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 :24).

**L. *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	Notasi <i>Gane</i>	Fungsi
<i>Yourdon</i>	<i>Sarson</i>	
<i>DeMarco</i>		
		Simbol Entitas eksternal atau terminator 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 :

1. *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).

## 2. *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 menunjukkandimana, 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* 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 *DFDlogis* untuk membuat dokumentasi sebuah sistem informasi karena *DFDlogis* 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 *DFDlogis* dibandingkan dengan *DFD* fisik

adalah dapat memusatkan perhatian pada fungsi-fungsi yang dilakukan sistem.

#### **M. 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:

1. Persegi panjang, menyatakan Himpunan Entitas.
2. Lingkaran/*Ellips*, menyatakan Atribut (*Atribut* yang berfungsi sebagai *key* digarisbawahi).
3. Belah ketupat, menyatakan Himpunan Relasi.
4. Garis, sebagai penghubung antara Himpunan Relasi dengan Himpunan Entitas dengan Atributnya.
5. 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).

#### **N. *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.

#### O. XAMPP

*XAMPP* merupakan salah satu aplikasi *web server cross platform* yang bisa dipakai di komputer *Windows*, *Linux*, maupun *Mac*. Di modul ini sudah tersedia modul *Apache*, *MySQL*, dan *FileZilla*. *Web server* adalah suatu *server internet* yang menggunakan *protocol HTTP* untuk melayani semua proses penransferan data. *Web server* melihat hubungan dengan *internet* dan semua menunggu perintah atau permintaan dari *web browser* akan *HTML* atau dokumen.