BAB II
TINJAUAN PUSTAKA

2.1 Profil BAPPEDA Kota Bandung

2.1.1 Sejarah BAPPEDA

Badan Perencanaan Pembangunan Daerah (Bappeda) Kota Bandung adalah salah satu lembaga teknis di lingkungan Pemerintah Kota Bandung. Awal mula pembentukan Bappeda bermula ketika pada tahun 1972 Pemerintah Provinsi Jawa Barat melakukan penyempurnaan Badan Perancang Pembangunan Daerah (Bappemda) Provinsi Jawa Barat dengan membentuk Badan Perancang Pembangunan Kotamadya (Bappemko) dan Badan Perancang Pembangunan Kabupaten (Bappemka), yang merupakan badan perencanaan pertama di Indonesia yang bersifat regional dan lokal serta ditetapkan dengan SK Gubernur Provinsi Jawa Barat No. 43 Tahun 1972.

Setelah berjalan 2 tahun, kedudukan Badan Perencanaan Pembangunan Daerah Tingkat I dikukuhkan dan diakui dengan SK Presiden No. 15 Tahun 1974, sedangkan untuk Daerah Tingkat II masih berlaku SK Gubernur. Baru kemudian dengan SK Presiden No. 27 Tahun 1980, Badan Perencanaan Pembangunan Daerah Tingkat II diakui secara nasional. Dengan SK Presiden tersebut, lahirlah Badan Perencanaan Pembangunan Daerah Tingkat I atau Bappeda Tingkat I dan Badan Perencanaan Pembangunan Daerah Tingkat II atau Bappeda Tingkat II.

Dalam lingkup Kota Bandung sendiri, pembentukan Bappeda Kotamadya Daerah Tingkat II Bandung didasarkan pada Perda No. 21 Tahun 1981 dan Perda No. 24 Tahun 1981, sebagaimana telah mengalami penyesuaian sejalan dengan perubahan paradigma pembangunan. Seiring dengan diberlakukannya Undang-Undang Nomor 22 Tahun 1999, maka Pemerintah Kota Bandung menata kembali Struktur Organisasi Perangkat Daerahnya, termasuk merubah nama Bappeda Kotamadya Daerah Tingkat II Bandung menjadi Bappeda Kota Bandung. Perubahan ini ditetapkan dengan Perda Kota Bandung No. 06 Tahun 2001 Tentang Pembentukan dan Susunan Organisasi Lembaga Teknis Daerah Tingkat Kota Bandung, sedangkan uraian tugas dan fungsinya ditetapkan dengan Perda No. 17 Tahun 2001 tentang Rincian Tugas Pokok dan Fungsi Satuan Organisasi Lembaga Teknis Daerah Kota Bandung.

Kemudian dengan berlakunya Undang-undang No. 25 Tahun 2000 tentang Kewenangan Daerah, maka keberadaan lembaga Bappeda di masing-masing daerah disesuaikan dengan tuntutan reformasi dan kebutuhan daerahnya dalam rangka pemenuhan optimalisasi pelayanan kinerja. Terkait dengan hal tersebut, susunan organisasi Bappeda Kota Bandung kembali ditetapkan dengan Perda No. 12 tahun 2007 tentang Susunan Organisasi dan Tata Kerja Pemerintah Kota Bandung.

2.1.2 Logo Instansi

Logo LPSE Bappeda masih menggunakan logo pusat. Seperti yang terlihat pada gambar 2.1 di bawah ini.



Gambar 2.1 Logo Instansi

2.1.3 Lambang

Lambang kota Bandung ditetapkan dengan Peraturan Daerah Kota besar Bandung tahun 1953, tertanggal 8 Juni 1953, yang diijinkan dengan Keputusan Presiden tertanggal 28 april 1953 No. 104 dan diundangkan dalam Berita Propinsi Jawa Barat tertanggal 28 Agustus 1954 No. 4 lampiran No. 6 Lambang tersebut bertokoh PERISAI yang berbentuk JANTUNG. Perisai tersebut terbagi dalam dua bagian oleh sebuah BALOK- LINTANG mendatar bertajuk empat buah, yang berwarna HITAM dengan pelisir berwarna PUTIH(PERAK) pada pinggir sebelah atasnya:

1. Bagian atas latar KUNING (EMAS) dengan lukisan sebuah GUNUNG berwaarna HIJAU yang bertumpu pada blok-lintang daaan
2. Bagian bawah latar PUTIH(PERAK) dengan lukisan empat bidaang jalur mendatar berombak yang berwarna BIRU.

Di bawah perisai itu terlukis sehelai PITA berwarna KUNING (EMAS) yang melambai pada kedua ujungnya, Pada pita itu tertulis dengan huruf-huruf besar latin

berwarna HITAM amsal dalam bahasa KAWI, yang berbunyi GEMAH RIPAH WIBAWA MUKTI. Sebagai tokoh lambang itu diambil bentuk perisai atau tameng, yang dikenal kebudayaan dan peradaban sebagai senjata dalam perjuangan untuk mencapai sesuatu tujuandengan melindungi diri. Perkakas perjuangan yang demikian itu dijadikan lambang yang mempunyai arti menahan segala mara bahaya dan kesukaran.

KUNING (EMAS), berarti : kesejahteraan, keluhungan.

HITAM (SABEL), berarti : kokoh, tegak, kuat.

HIJAU (SINOPEL), berarti : kemakmuran sejuk.

PUTIH (PERAK), berarti : kesucian.

BIRU (AZUUR), berarti : kesetiaan.

Gemah ripah wibawa mukti, berarti : tanah subur rakyat makmur.

2.1.4 Badan Hukum Instansi

Fungsi dan peran BAPPEDA sebagai lembaga teknis daerah yang bertanggung jawab terhadap perencanaan pembangunan sebagaimana diamanatkan dalam pasal 14, ayat (1), Undang-Undang Nomor 32 Tahun 2004 tentang Pemerintahan Daerah, bahwa salah satu urusan wajib yang menjadi kewenangan pemerintah daerah adalah urusan perencanaan dan pengendalian pembangunan. Kewenangan perencanaan pengendalian tersebut kemudian dipertegas kembali dalam Peraturan Pemerintah Nomor 38 Tahun 2007 tentang Pembagian Urusan Pemerintahan antara Pemerintah, Pemerintah Daerah Provinsi dan Pemerintah Daerah Kabupaten/Kota, dari 26 (dua puluh enam) urusan sesuai dengan pasal 7, ayat (2), BAPPEDA sebagai salah satu lembaga teknis daerah yang merupakan unsur pendukung tugas kepala daerah, mengemban 3 (tiga) urusan wajib yang wajib dilaksanakan, yaitu urusan penataan ruang, perencanaan pembangunan dan urusan statistik. Selain itu dalam Undang-Undang Nomor 25 Tahun 2004 tentang Sistem Perencanaan Pembangunan Nasional.

2.1.5 Visi dan Misi Bappeda Kota Bandung

Visi Bappeda Kota Bandung yaitu Terwujudnya Bappeda Sebagai Lembaga Perencanaan Pembangunan Yang Kredibel Dalam Memantapkan Kota Bandung Sebagai Kota Jasa Bermartabat.

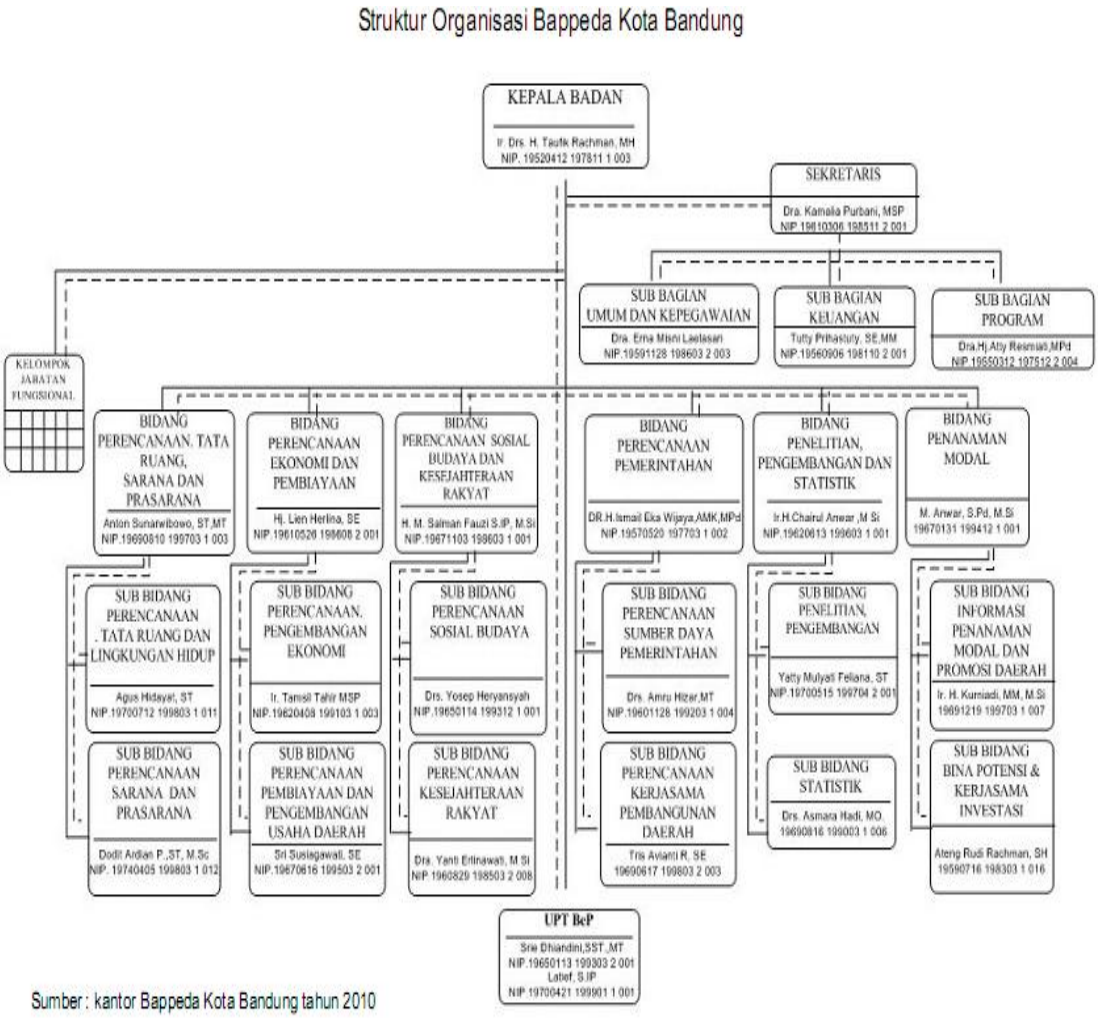
Sedangkan misinya diantaranya :

1. Meningkatkan kompetensi aparatur perencanaan pembangunan daerah Kota Bandung yang professional.
2. Meningkatkan kualitas dan kuantitas sarana prasarana perencanaan pembangunan yang memadai.
3. Memantapkan sistem pengelolaan perencanaan pembangunan daerah yang terintegrasi dan transparan.
4. Meningkatkan sinergitas penyelenggaraan perencanaan pembangunan internal daerah, antar Pemerintah Kabupaten/Kota Provinsi dan Pusat.
5. Meningkatkan kerjasama perencanaan pembangunan dengan dunia usaha dalam dan luar negeri.

2.1.6 Struktur Organisasi BAPPEDA

Struktur organisasi dalam suatu dinas maupun instansi-intansi pada suatu organisasi sangat diperlukan keberadaannya. Karena struktur organisasi ini dapat dijadikan pedoman dalam pembagian tugas, oleh setiap bagian sesuai dengan fungsinya masing-masing agar lebih mengarah pada pelaksanaan pedoman kerja yang telah disusun sebelumnya. Bappeda Kota Bandung mempunyai struktur organisasi garis dan staf, maksudnya adalah dalam setiap tingkat level organisasi dikepalai oleh seorang atasan yang membawahi beberapa orang aparatur.

Adapun bagan struktur organisasi Bappeda Kota Bandung dapat dilihat pada gambar 2.2 dibawah ini.



Gambar 2.2 Struktur Organisasi Bappeda Kota Bandung

2.2 LANDASAN TEORI

Tools yang digunakan dalam membangun Aplikasi Pengadaan Data Barang yang berbasis *Web*, antara lain:

2.2.1 Database

Menurut DAVID M. KROENKE, *Database* merupakan fungsi utama untuk aplikasi *client - server*, aplikasi organisasi, dan aplikasi *e-commerce* bisnis ke konsumen serta bisnis ke bisnis.

Kumpulan fakta yang saling berhubungan disimpan secara bersama sedemikian rupa dan tanpa pengulangan(*redudansi*) yang tidak perlu untuk memenuhi berbagai kebutuhan. Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data juga perlu diorganisasikan

sedemikian rupa supaya informasi yang dihasilkan mempunyai kualitas yang baik. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya.[2]

Basis data biisa juga diartikan sebagai sekumpulan informasi yang disusun sedemikian rupa untuk dapat diakses oleh sebuah software tertentu. Database tersusun atas *field* dan *record* yang terdapat dalam sebuah file. Sebuah *field* merupakan kesatuan terkecil dari informasi dalam sebuah database. Sekumpulan *field* yang saling berkaitan akan membentuk *record*.[3]

Dalam membuat aplikasi ini, kami membuat di *MySql (My Structure Query Language)* sebagai *database* kami. *MySql* merupakan suatu perangkat lunak *database* relasi *(Relational Database Management System* atau *RDBMS)* [4]. Sebuah program pembuat *database* yang bersifat *open source* dan berjalan di semua *platform* baik *Windows* maupun *Linux*. Selain itu, *MySQL* juga merupakan program pengakses *database* yang bersifat jaringan sehingga dapat digunakan untuk aplikasi *Multi User* (Banyak Pengguna). .Sangat cocok jika digabungkan dengan *PHP*

MySQL adalah sebuah perangkat lunak sistem manajemen basis data *SQL (database management system)* atau *DBMS* yang *multithread, multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. *MySQL AB* membuat *MySQL* tersedia sebagai perangkat lunak gratis dibawah lisensi *GNU General Public License (GPL)*, tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan *GPL*. Tidak sama dengan proyek-proyek seperti *Apache*, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, *MySQL* dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia *MySQL AB*, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan *MySQL AB* adalah David Axmark, Allan Larsson, dan Michael "Monty" Widenius.[4]

2.2.2 HTML

Hypertext Markup Language merupakan kepanjangan dari kata HTML. HTML adalah *script* dimana kita bisa menampilkan informasi dan daya kreasi kita lewat *intenet*. HTML juga merupakan *file* teks murni yang dapat dibuat dengan *editor* teks sembarang yaitu yang dikenal sebagai *web page* atau dokumen yang disajikan dalam *web browser*. Dokumen ini umumnya berisi informasi atau *interface*

aplikasi didalam *internet*.[8]

HTML sendiri adalah suatu dokumen teks biasa yang mudah dimengerti dibanding bahasa pemrograman lainnya, dan karena bentuknya itu maka HTML dapat dibaca oleh berbagai *platform* seperti : windows, Linux, Macintosh. Kata *Markup Language* pada HTML menunjukkan fasilitas yang berupa tanda tertentu dalam *script* HTML sehingga kita bisa mengatur judul, garis, tabel, gambar dan lain-lain dengan perintah yang telah ditentukan dalam elemen HTML.

Simbol penandaan yang digunakan dalam HTML ditandai dengan tanda lebih kecil (<) dan tanda lebih besar (>) yang disebut dengan *tag*. Misalnya hendak menampilkan teks yang tercetak miring, maka *mark up* yang digunakan adalah sebagai berikut :

** Teks ini akan dicetak miring **

HTML terdiri dari beberapa bagian yang fungsinya sebagai penanda suatu kelompok perintah tertentu, misalnya kelompok perintah *form* yang ditandai dengan kode <form>, judul dengan <title> dan sebagainya. Untuk lebih lanjut mengenai bagian-bagian HTML perhatikan skema dibawah ini :

```
<html>
<head>
<title>Judul Halaman</title>
</head>
<body>
…isi dari halaman web…
</body>
</html>
```

Keterangan :

1. Dokumen HTML selalu diawali dengan tand *tag* pembuka <html> dan diakhiri dengan tanda *tag* penutup </html>.
2. Pada elemen head <head></head>, dapat kita sisipkan kode untuk menuliskan keterangan tentang dokumen HTML. Atau dapat juga kita sisipkan *scripts* pemrograman *web* seperti Javascript, VBscript atau CSS untuk menambah daya tarik pada situs yang kita buat agar lebih menarik dan dinamis.
3. Elemen-elemen <body></body> berisi tag-tag untuk isi atau layout tampilan situs kita, seperti : , <table></table>, <form></form>.

Jadi jelas bahwa elemen adalah suatu bagian yang besar yang terdiri dari kode-kode *tag* tersebut. Sedangkan *tag* hanyalah merupakan bagian dari elemen. *Tag* adalah kode-kode yang digunakan untuk mengatur dokumen HTML. Secara garis besar bentuk umum *tag* adalah sebagai berikut : <tag-awal>TEKS<tag-akhir>. Namun ada juga *tag* yang tidak perlu ada *tag* penutup seperti <br>, <hr>, <img> dan lain-lain.

2.2.3 Web Programming (PHP)

PHP Hypertext Preprocessor adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam *HTML*. *PHP* banyak dipakai untuk memrogram situs web dinamis. *PHP* dapat digunakan untuk membangun sebuah *CMS.* [1]

Pada awalnya *PHP* merupakan kependekan dari *Personal Home Page (Situs personal). PHP* pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu *PHP* masih bernama *Form Interpreted (FI),* yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya *PHP/FI.* Dengan perilisan kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan *PHP*.

Pada November 1997, dirilis *PHP/FI 2.0.* Pada rilis ini, interpreter *PHP* sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan *PHP/FI* secara signifikan. Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter *PHP* menjadi lebih

bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk *PHP* dan meresmikan rilis tersebut sebagai *PHP 3.0* dan singkatan *PHP* diubah menjadi akronim berulang PHP *(Hypertext Preprocessing).*

Pada pertengahan tahun 1999, Zend merilis interpreter *PHP* baru dan rilis tersebut dikenal dengan *PHP 4.0*. *PHP 4.0* adalah versi *PHP* yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi. Pada Juni 2004, Zend merilis *PHP 5.0*. Dalam versi ini, inti dari interpreter *PHP* mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam *PHP* untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek.

PHP (Hypertext Preprocessor) merupakan bahasa pemrograman yang berjalan di sisi server atau biasa disebut dengan *Server Side.* Secara garis besar bahasa pemrograman web terdiri dari dua macam. Yang pertama disebut dengan *Server Side* atau script yang dijalankan pada computer. *Server* dan yang kedua disebut dengan *Client Side* atau *script* yang dijalankan pada komputer *Client*. Untuk dapat menjalankan *PHP* membutuhkan sebuah *web server*, disini kami menggunakan *wamp server* sebagai *web server* nya.

Adapun fungsi *PHP* untuk mengakses *MySQL* yang biasa digunakan diantaranya adalah:

a. mysql_connect()

Fungsi mysql_connect adalah untuk menghubungkan *PHP* dengan database *MySQL*.

Format fungsinya adalah:

```
mysql_connect(string hostname, string username, string password);
```

b. mysql_select_db

Setelah terhubung ke database *MySQL* dengan menggunakan mysql_connect, langkah selanjutnya adalah memilihi database yang akan digunakan. Fungsi mysql_connect_db digunakan untuk memilih database. Format fungsinya adalah:

```
mysql_select_db(string database, koneksi);
```

c. mysql_query

Dalam *database MySQL*, perintah untuk melakukan trnasaksi ialah perintah *SQL*. Sebutan untuk mengirim perintah *SQL* dinamakan *query*. *Query* memberi perintah kepada database untuk melakukan apa yang dikehendaki. Format fungsinya adalah:

```
int mysql_query(string query, int[link_identifier]);
```

d. mysql_num_rows

Kegunaan dari fungsi ini adalah untuk menghitung jumlah baris yang dikenai oleh proses *SQL*. Format penulisannya adalah:

```
int mysql_num_rows(int result);
```

e. mysql_fetch-array

Fungsi ini digunakan untuk menampilkan data. Dalam fungsi ini, hasil query ditampung dalam bentuk array. Format fungsinya adalah:

```
Array mysql_fetch_array(int result, int [result_type]);
```

2.2.4 Web Server

Untuk menjalankan *PHP* kami menggunakan *Wamp server*. Aplikasi ini dirancang untuk menjadi *windows web* lingkungan pengembangan. Wamp server juga dilengkapi dengan *php myadmin* dan *sqlitemanager* untuk dengan mudah mengelola *database*. *Wamp* server diinstal secara otomatis dan penggunaannya sangat intuitif. *User* dapat menyetel *server* anda tanpa menyentuh file pengaturan. *Wamp server* merupakan salah satu solusi yang akan memungkinkan untuk mereproduksi *server* produksi. *Wamp* server setelah terinstal akan terintegrasi dengan *apache*, *mysql*, dan *php*. *Wamp* server juga memiliki *trayicon* untuk mengelola *server* anda dan pengaturannya.

Fungsi utama sebuah server web adalah untuk mentransfer berkas atas permintaan pengguna melalui protokol komunikasi yang telah ditentukan. Disebabkan sebuah halaman *web* dapat terdiri atas berkas teks, gambar, video, dan lainnya pemanfaatan server web berfungsi pula untuk mentransfer seluruh aspek pemberkasan dalam sebuah halaman web yang terkait; termasuk di dalamnya teks, gambar, video, atau lainnya. Pengguna, biasanya melalui aplikasi pengguna seperti peramban *web*, meminta layanan atas berkas ataupun halaman *web* yang terdapat pada sebuah server *web*, kemudian server sebagai manajer layanan tersebut akan merespon balik dengan

mengirimkan halaman dan berkas-berkas pendukung yang dibutuhkan, atau menolak permintaan tersebut jika halaman yang diminta tidak tersedia.[6]

Saat ini umumnya *server web* telah dilengkapi pula dengan mesin penerjemah bahasa skrip yang memungkinkan *server web* menyediakan layanan situs web dinamis dengan memanfaatkan pustaka tambahan *seperti PHP, ASP*. Pemanfaatan *server web* saat ini tidak terbatas hanya untuk publikasi situs *web* dalam *World Wide Web*, pada prakteknya *server web* banyak pula digunakan dalam perangkat-perangkat keras lain seperti *printer, router*, kamera *web* yang menyediakan akses layanan *http* dalam jaringan lokal yang ditujukan untuk menyediakan perangkat manajemen serta mempermudah peninjauan atas perangkat keras tersebut.

2.2.5 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah suatu diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data sistem, yang penggunaannya sangat membantu untuk memahami sistem secara logika, tersruktur dan jelas. *DFD* merupakan alat bantu dalam menggambarkan atau menjelaskan sistem yang sedang berjalan logis. Suatu yang lazim bahwa ketika menggambarkan sebuah sistem kontekstual *data flow diagram* yang akan pertama kali muncul adalah interaksi antara sistem dan entitas luar. *DFD* didesain untuk menunjukkan sebuah sistem yang terbagi-bagi menjadi suatu bagian sub-sistem yang lebih kecil adan untuk menggarisbawahi arus data antara kedua hal yang tersebut diatas. Diagram ini lalu "dikembangkan" untuk melihat lebih rinci sehingga dapat terlihat model-model yang terdapat di dalamnya [7]

DFD ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.

Komponen *data flow diagram* menurut Yourdan dan DeMarco :

a. Terminator / Entitas Luar

Terminator mewakili entitas Eksternal yang berkomunikasi dengan sistem yang sedang dikembangkan. Biasanya terminator dikenal dengan nama entitas luar (*external entity*). Terminator mempunyai dua jenis:

1. Terminator sumber (*source*)
2. Terminator Tujuan (*sink*)

Terminator dapat berupa orang, sekelompok orang, organisasi, departemen di dalam organisasi, atau perusahaan yang sama tetapi di luar kendali sistem yang sedang dibuat modelnya. Terminator dapat juga berupa departemen, divisi atau sistem di luar sistem yang berkomunikasi dengan sistem yang sedang dikembangkan.

Komponen terminator ini perlu diberi nama sesuai dengan dunia luar yang berkomunikasi dengan sistem yang sedang dibuat modelnya, dan biasanya menggunakan kata benda, misalnya Bagian Penjualan, Dosen, Mahasiswa.

Ada tiga hal penting yang harus diingat tentang terminator :

1. Terminator merupakan bagian/lingkungan luar sistem. Alur data yang menghubungkan terminator dengan berbagai proses sistem, menunjukkan hubungan sistem dengan dunia luar.
2. Profesional Sistem Tidak berhak mengubah isi atau cara kerja organisasi atau prosedur yang berkaitan dengan terminator.
3. Hubungan yang ada antar terminator yang satu dengan yang lain tidak digambarkan pada DFD.

b. Proses

Komponen proses menggambarkan bagian dari sistem yang mentransformasikan input menjadi output.

Proses diberi nama untuk menjelaskan proses/kegiatan apa yang sedang/akan dilaksanakan. Pemberian nama proses dilakukan dengan menggunakan kata kerja transitif (kata kerja yang membutuhkan obyek), seperti Menghitung Gaji, Mencetak KRS, Menghitung Jumlah SKS.

Ada beberapa hal yang perlu diperhatikan tentang proses:

1. Proses harus memiliki input dan output.
2. Proses dapat dihubungkan dengan komponen terminator, data store atau proses melalui alur data.
3. Sistem/bagian/divisi/departemen yang sedang dianalisis oleh profesional sistem digambarkan dengan komponen proses.

c. Data Store

Komponen ini digunakan untuk membuat model sekumpulan paket data dan diberi nama dengan kata benda jamak, misalnya Mahasiswa.

Data store ini biasanya berkaitan dengan penyimpanan-penyimpanan, seperti file atau database yang berkaitan dengan penyimpanan secara komputerisasi, misalnya file disket, file harddisk, file pita magnetik. Data store juga berkaitan dengan penyimpanan secara manual seperti buku alamat, file folder, dan agenda.

Suatu data store dihubungkan dengan alur data hanya pada komponen proses,tidak dengan komponen DFD lainnya. Alur data yang menghubungkan data store dengan suatu proses mempunyai pengertian sebagai berikut:

1. Alur data dari data store yang berarti sebagai pembacaan atau pengaksesan satu paket tunggal data, lebih dari satu paket data, sebagian dari satu paket tunggal data, atau sebagian dari lebih dari satu paket data untuk suatu proses.
2. Alur data ke data store yang berarti sebagai pengupdatean data, seperti menambah satu paket data baru atau lebih, menghapus satu paket atau lebih, atau mengubah/memodifikasi satu paket data atau lebih.

Pada pengertian pertama jelaslah bahwa data store tidak berubah, jika suatu paket data/informasi berpindah dari data store ke suatu proses. Sebaliknya pada pengertian kedua data store berubah sebagai hasil alur yang memasuki data store. Dengan kata lain, proses alur data bertanggung jawab terhadap perubahan yang terjadi pada data store.

d. Alur Data

Suatu data flow / alur data digambarkan dengan anak panah, yang menunjukkan arah menuju ke dan keluar dari suatu proses. Alur data ini digunakan

untuk menerangkan perpindahan data atau paket data/informasi dari satu bagian sistem ke bagian lainnya.

Selain menunjukkan arah, alur data pada model yang dibuat oleh profesional sistem dapat merepresentasikan bit, karakter, pesan, formulir, bilangan real, dan macam-macam informasi yang berkaitan dengan komputer.

Alur data juga dapat merepresentasikan data/informasi yang tidak berkaitan dengan komputer. Alur data perlu diberi nama sesuai dengan data/informasi yang dimaksud, biasanya pemberian nama pada alur data dilakukan dengan menggunakan kata benda, contohnya Laporan Penjualan.

Ada empat konsep yang perlu diperhatikan dalam penggambaran alur data, yaitu :

1. Konsep Paket Data (*Packets of Data*)
2. Konsep Alur Data Menyebar (*Diverging Data Flow*)
3. Konsep Alur Data Mengumpul (*Converging Data Flow*)
4. Konsep Sumber atau Tujuan Alur Data

2.2.6 Entity Relationship Diagram (ERD)

Entity Relationship Diagram merupakan model data berupa notasi grafis dalam pemodelan data konseptual yang menggambarkan hubungan antara penyimpan. Model data sendiri merupakan sekumpulan cara, peralatan untuk mendeskripsikan data-data yang hubungannya satu sama lain, semantiknya, serta batasan konsistensi. Model data terdiri dari model hubungan entitas dan model relasional. Diagram hubungan entitas ditemukan oleh Peter Chen dalam buku Entity Relational Model-Toward a Unified of Data. Chen mencoba merumuskan dasar-dasar model dan setelah itu dikembangkan dan dimodifikai oleh Chen dan banyak pakar lainnya. Pada saat itu diagram hubungan entitas dibuat sebagai bagian dari perangkat lunak yang juga merupakan modifikasi khusus, karena tidak ada bentuk tunggal dan standar dari diagram hubungan entitas.[9]

Komponen-komponen dari ERD adalah sebagai berikut:

1. Tipe Objek (*Object Type*)

   Tipe objek digambarkan dengan sebuah kotak yang berarti kumpulan dari objek-objek nyata.

2. Hubungan (*Relationship*)

   Objek-objek yang telah diidentifikasi dihubungkan satu sama lain dengan menggunakan suatu penghubung yang digambarkan dengan suatu tanda panah.

   Beberapa jenis hubungan:

   - *One to One Relationship*

     Hubungan antara entity dan entity kedua adalah satu berbading satu.

   - *One to Many Relationship*

     Hubungan antara entity petama dan kedua adalah satu berbanding banyak atau dapat dibalik menjadi banyak lawan satu.

   - *Many to Many Relationship*

     Hubungan antara entity pertama dan entity kedua adalah banyak berbanding banyak.