BAB II

TINJAUAN PUSTAKA

2.1 Profil Perusahaan

Dinas Komunikas dan Informatika (DISKMINFO) provinsi Jawa Barat merupakan salah satu instansi dari pemerintah kota Bandung yang berlokasi di jalan Taman Sari No 55 Bandung. Berikut akan dijelaskan tugas pokok, fungsi, serta visi dan misi DISKOMINFO kota Bandung.

2.1.1 Sejarah Instansi

Kantor Pengolahan Data Elektronik (KPDE) Provinsi Daerah Tingkat I Jawa Barat adalah kelanjutan dari organisasi sejenis yang semula sudah ada di lingkungan Pemerintah Provinsi Daerah Tingkat I Jawa Barat dengan nama Pusat Pengolahan Data (PUSLAHTA) Provinsi Daerah Tingkat I Jawa Barat.

Keberadaan PUSLAHTA di Jawa Barat dimulai pada tahun 1977, yaitu dengan adanya Proyek Pembangunan Komputer Pemerintah Provinsi Daerah Tingkat I Jawa Barat. Proyek tersebut dimaksudkan untuk mempersiapkan sarana prasarana dalam rangka memasuki era komputer. Dalam perkembangan selanjutnya, pada tanggal 8 April 1978 dengan Surat Keputusan Gubernur Kepala Daerah Tingkat I Jawa Barat Nomor : 294/Ok.200-Oka/SK/78 diresmikan pembentukan/pendirian Kantor Pusat Pengolahan Data (PUSLAHTA) Provinsi Daerah Tingkat I Jawa Barat yang berkedudukan di jalan Tamansari No. 57 Bandung.

Sebagai tindak lanjut dari Surat Keputusan Gubernur Nomor : 294/Ok.200-Oka/SK/78, maka pada tanggal 29 Juni 1981 pendirian Kantor PUSLAHTA dikukuhkan dengan Peraturan Daerah Nomor : 2 Tahun 1981 tentang Pembentukan Pusat Pengolahan Data (PUSLAHTA) Provinsi Daerah Tingkat I Jawa Barat dan Peraturan Daerah Nomor : 3 Tahun 1981 tentang Susunan Organisasi dan Tata Kerja Pusat Pengolahan Data Provinsi Daerah

Tingkat I Jawa Barat. Dengan kedua Peraturan Daerah tersebut keberadaan PUSLAHTA di lingkungan Pemerintah Provinsi Daerah Tingkat I Jawa Barat semakin berperan, khususnya dalam melaksanakan kebijaksanaan Gubernur Kepala Daerah di bidang komputerisasi. Akan tetapi keberadaan kedua Peraturan Daerah tersebut tidak mendapat pengesahan dari pejabat yang berwenang dalam hal ini Menteri Dalam Negeri, sehingga keberadaan PUSLAHTA di lingkungan Pemerintah Daerah Tingkat I Jawa Barat kedudukan organisasi menjadi non structural. Akan tetapi dengan keberadaan Puslahta Provinsi Daerah Tingkat I Jawa Barat pada masa itu telah banyak dirasakan manfaatnya selain oleh lingkungan Pemerintah Provinsi Jawa Barat juga oleh instansi lain dalam bentuk kerja sama penggunaan mesin komputer IBM S-370/125 seperti : IPTN, PJKA, ITB Dan pihak Swasta lainnya.

2.1.2 Visi dan Misi

Dalam menjalankan perusahaan kearah yang lebih maju maka diperlukan sebuah Visi dan Misi yang jelas agar kemajuan suatu perusahaan dapat tercapai, adapun visi dan misi DISKOMINFO (Dinas Komunikasi dan informatika) sebagai berikut :

Visi **DISKOMINFO** adalah:

Terwujudnya masyarakat informasi Jawa Barat melalu penyelenggaraan komunkasi dan informatika yang efektif dan efisien .

Misi **DISKOMINFO** adalah:

1. Meningkatkan sarana dan prasana dan profesionalisme sumber daya aparatur bidang Komunikasi dan Informatika.
2. Mengoptimalkan pemanfaatan sarana Komunikasi dan Informasi pemerintah dan masyarakat, serta melaksanakan diseminasi informasi.
3. Mewujudkan layanan online dalam penyelenggaraan pemerintah berbasis Teknologi Informasi dan Komunikasi.
4. Mewujudkan pengelolaan data menuju satu data pembangunan untuk Jawa barat.

2.2 Logo Instansi

2.2.1 Arti Logo

Logo merupakan suatu bentuk gambar atau sekedar sketsa dengan arti tertentu, dan mewakili suatu arti dari perusahaan, daerah, perkumpulan, produk, negara dan hal-hal lainnya yang dianggap membutuhkan hal yang singkat dan mudah sebagai pengganti dari nama sebenarnya.



Gambar 2.1 Logo perusahaan

Keterangan dari Logo DISKOMINFO :

1. Lambang berbentuk bulat telor
   Bentuk ini berasal dari bentuk perisai yang banyak dipakai oleh laskar kerajaan-kerajaan dulu.
2. Kujang
   Adalah gambar alat serba guna yang sangat dikenal dihampir setiap rumah tangga sunda, lima lubang yang ada di dalam kujang melambangkan lima dasar pokok negara”PANCASILA”.

3. Padi

   Melambangkan bahan makanan pokok di Jawa Barat serta sekalian melambangkan pangan, jumlah padi(17 buah) menyatakan hari ke-17 dari tahun proklamasi.

4. Kapas

   Melambangkan sandang jumlah kapas (8 buah) menyatakan bulan ke 8 dari tahun proklamasi.

5. Gunung

   Melambangkan badian terbesar dari Jawa Barat terdiri dari daerah pegunungan.

6. Padi dan Kapas

   Pada dasar hijau melambangkan kesuburan dan kemakmuran tanah Jawa Barat.

7. Sungai dan terusan

   Melambangkan sungai, terusan dan saluran air yang banyak terdapat di daerah Jawa Barat.

8. Sawah, Perkebunan

   Melambangkan jumlah sawah tidak sedikit, tersebar diseluruh wilayah Jawa Barat, perkebunan di bagian tengah dan selatan.

9. Dam, Saluran Air dan Bendungan

   Usaha dan pekerjaan dibidang irigasi merupakan salah satu pekerjaan yang mendapat perhatian pokok, mengingat sifat agrasi didaerah Jawa Barat.

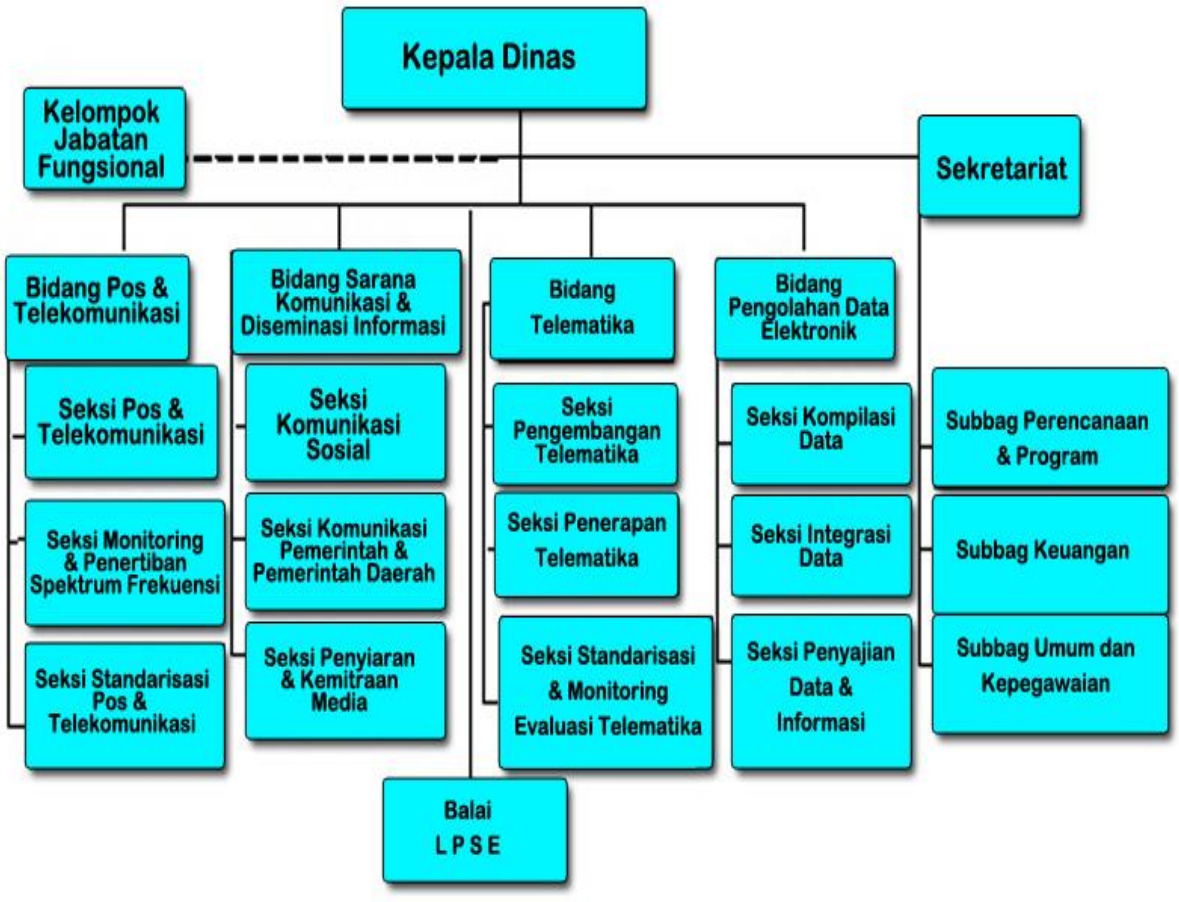
10. Gemah Ripah, Repeh Rapih

    Merupakan sebuah pepatah lama dikalangan masyarakat Sunda yang menyatakan bahwa daerah Jawa Barat yang kaya raya ini didiami oleh penduduk yang padat yang hidup rukun dan damai.

2.3 Struktur Organisasi

Dalam menjalankan kegiatan dinas, DISKOMINFO telah mengelompokan unit -unit bagian yang ada dalam organisasi ke dalam beberapa bentuk bidang. Dibawah ini akan dijelaskan struktur organisasi DISKOMINFO Jawa Barat sebagai berikut :

1. Bidang Pos dan Telekomunikasi.
2. Bidang Sarana Komunikasi dan Diseminasi.
3. Bidang Telematika.
4. Bidang Pengolahan Data Elektronik.



Gambar 2.2 Struktur Organisasi

Gambar dari struktur organisasi DISKOMINFO. Struktur Organisasi Dinas dan Informatika (DISKOMINFO) terdiri dari seorang Kepala Dinas (Eselon II), seorang Sekretaris Dinas (Eselon III), 4 (empat) orang Kepala Bidang (Eselon III), dan 15 Kepala Seksi (Eselon IV)

Bidang Telematika membawahi tiga kepala subbagian meliputi :

- Seksi keamanan sosial
- Seksi Penerapan Telematika
- Seksi standarisasi dan monitoring Evaluasi Telematika

2.4 Landasan Teori

Dalam pemecahan masalah yang dianggap relevan dengan pokok bahasan dalam laporan kerja praktek ini yaitu mengenai pengertian Konsep Basis Data(*Database*), ERD dan Data Flow Diagram (DFD).

2.4.1 Basis Data (*Database)*

2.4.1.1 Pengertian Basis Data

Basis data dibayangkan sebagai sebuah lemari arsip. Jika kita memiliki sebuah lemari arsip dan berweang untuk mengelolanya, maka kemungkinan besar kita akan melakukan hal – hal seperti : meberi map pada kumpulan – kumpulan arsip yang akan disimpan, menentukan kelompok atau jenis arsip, memberi penomoran dengan pola tertentu yang nilainya unik pada setiap map, lalu menempatkan arsip – arsip tersebut dengan urutan tertentu di dalam lemari.

Basis data terdiri dari dua kata yaitu Basis yang dapat diartikan sebagai markas atau gudang, sedangkan Data adalah 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 sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti :

a. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
c. Kumpulan file atau table atau arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

Yang sangat ditonjolkan dalam basis data adalah pengaturan atau pemilihan atau pengelompokkan atau pengorganisasian data yang akan kita simpan sesuai fungsi ataupun jenisnya. Hal ini dapat berbentuk sejumlah file atau table terpisah atau dalam bentuk pendefinisian kolom-kolom data dalam setiap file atau table.

2.4.1.2 Sistem Basis Data

Sistem adalah sebuah tatanan (keterpaduan) yang terdiri atas sejumlah komponen fungsional (dengan satuan fungsi / tugas khusus) yang saling berhubungan dan secara bersama – sama bertujuan untuk memenuhi suatu proses atau pekerjaan tertentu. Basis data hanyalah sebuah objek yang pasif atau mati. Ia ada karena ada pembuatnya. Ia tidak akan pernah berguna jika tidak ada pengelola atau penggeraknya. Yang menjadi penggeraknya secara langsung adalah program atau aplikasi (*software*). Gabungan keduanya menghasilakn sebuah sistem. Karena itu, secara umum sebuah sistem basis data merupakan sistem yang terdiri atas kumpulan file (table) yang saling berhubungan (dalam sebuah basis data di sebuah sistem komputer) dan sekumpulan program (DBMS) yang memungkinkan beberapa pemakai dan atau program lain untuk mengakses dan memanipulasi file – file (table - tabel) tersebut.

Lebih jauh lagi, dalam sebuah sistem basis data, secara lengkap akan terdapat komponen – komponen utama sebagai berikut :

a. Perangkat keras (*Hardware*)

Perangkat keras yang biasanya terdapat dalam sebuah sistem basis data adalah :

1. Komputer (satu untuk sistem yang *stand – alone* atau lebih dari satu untuk sistem jaringan).
2. Memori sekunder yang *on – line* (*harddisk*).
3. Memori sekunder yang *off – line* (*Tape* atau *Removable Disk*) untuk keperluan *backup* data.
4. Media atau perangkat komunikasi (untuk sistem jaringan).

b. Sistem Operasi (*Operating System*)

Secara sederhana, sistem operasi merupakan program yang mengaktifkan atau memfungsikan sistem komputer, mengendalikan seluruh sumber daya (*resource*) dalam komputer dan melakukan operasi – operasi dasar dalam computer (operasi I/O, pengelolaan file, dll). Program pengelola basis data hanya dapat aktif (*running*) jika sistem operasi yang dikehendakinya atau sesuai telah aktif.

c. Basis Data (*Database*)

Sebuah sistem basis data dapat memiliki beberapa basis data. Setiap basis data dapat berisi sejumlah objek basis data seperti *file / table*, indeks, dll. Di samping berisi atau menyimpan data, setiap basis data juga mengandung definisi struktur (baik untuk basis data maupun objek – objeknya secara deta

d. Sistem (Aplikasi / perangkat lunak) Pengelola Basis Data (DBMS)

Pengelolaan basis data secara fisik tidak dilakukan oleh pemakai secara langsung, tetapi ditangani oleh sebuah perangkat lunak yang khusus. Perangkat lunak ini disebut DBMS (*Database Management Sistem*) yang yang akan menentukan bagaimana data diorganisasi, disimpan, diubah dan diambil kembali. Ia juga menerapkan mekanisme pengamanan data, pemakaian data secara bersama, pemaksaan keakuratan atau konsistensi data, dan sebagainya.

e. Pemakai (*User*)

Ada beberapa jenis pemakai terhadap suatu sistem basis data yang dibedakan berdasarkan cara mereka berinteraksi terhadap sistem :

1. Programmer Aplikasi

   Pemakai yang berinteraksi dengan basis data melalui *Data Manipulation Language* (DML), yang disertakan (*embedded*) dalam program yang ditulis dalam bahasa pemrograman induk (seperti C, Pascal, Cobol, dan lain lain).

2. User Mahir (*Casual User*)

   Pemakai yang berinteraksi dengan sistem tanpa menulis modul program. Mereka menyatakan *query* (untuk akses data) dengan bahasa *query* yang disediakan oleh suatu DBMS.

3. User Umum (*End User / Naïve User*)

   Pemakai yang berinteraksi dengan sistem basis data melalui pemanggilan satu program aplikasi permanen (*executable program*) yang telah ditulis atau disediakan sebelumnya.

4. User Khusus (*Specialized User*)

   Pemakai yang menulis aplikasi basis data non-konvensional, tetapi untuk keperluan – keperluan khusus, seperti untuk aplikasi AI, Sistem Pakar, Pengolahan Citra, dll, yang bisa saja mengakses basis data dengan / tanpa DBMS yang bersangkutan.

f. Aplikasi (*perangkat lunak*) lain

Aplikasi (perangkat lunak) lain ini bersifat opsional. Artinya, ada atau tidaknya tergantung pada kebutuhan kita. DBMS yang kita gunakan lebih berperan salam pengorganisasian data dalam basis data, sementara bagi pemakai basis data (khususnya yang menjadi *end – user* / *naive - user*) dapat dibutuhkan / disediakan program khusus / lain untuk melakukan pengisian, pengubahan dan pengambilan data. Pemrogram ini ada yang sudah disediakan bersama dengan DBMS – nya, ada juga yang harus dibuat sendiri dengan menggunakan aplikasi lain yang khusus untuk itu (*delevopment tools*).

2.4.1.3 Model data

Model data sendiri dapat didefinisikan sebagai kumpulan perangkat konseptual untuk menggambarkan data, hubungan data, semantic (makna) data dan batasan data. Oleh karena yang ingin ditunjukkan adalah makna dari data dan keterhubungannya dengan data lain, maka Model Data ini lebih tepat jika disebut Model Data Logik. Penyingkatannya menjadi Model Data dilakukan demi penyederhanaan penulisan.

Ada sejumlah secara dalam meprestasikan Model Data dalam perancangan basis data, yang secara umum dapat dibagi dalam 2 (dua) kelompok yaitu, yaitu :

a. Model Logik Data Berdasarkan Objek (*Objek – Based Logical Models*) yang terdiri dari :
   1. *Entity – Relationship Model*
   2. *Object – Oriented Model*
   3. *Semantic Data model*
   4. *Functional Data model*
b. Model Lojik Data Berdasarkan *Record* (*Record – Based Locical* Models) yang terdiri dari :
   1. *Relational Model*
   2. *Hierarchical Model*
   3. *Network Model*

2.4.1.4 Model *Entity–Relationship* (Model Keterhubungan-Entitas)

Pada model Entity – Relationship, semesta data yang ada di ‘Dunia Nyata’ diterjemahkan dengan memanfaatkan sejumlah perangkat konseptual menjadi sebuah diagram data, yang umum disebut sebagai Diagram *Entity – Relationship* (Diagram E – R). sebelum kita mebahas lebih jauh tentang bagaimana Diagram E - R tersebut dapat kita gambarkan, maka yang harus lebih dulu diketahui adalah komponen – komponen pembentukan Model Entity – relationship. Sesuai namanya, ada dua komponen utama pembentuk Model Entity –Relationship, yaitu Entitas (*Entity*) dan Relasi (*Relation*). Kedua komponen ini di deskripsikan lebih jauh melalui sejumlah *Atribut* atau *Properti*.

2.4.1.5 Entitas (*Entity)* dan Himpunan Entitas (*Entitas Sets*)

Entitas merupakan invidu yang mewakili sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain. Contoh – contoh himpunan entitas adalah:

a. Semua pelanggan atau pelanggan saja. Dengan entitas Budiman, Aminah dan seterusnya.
b. Semua mobil atau mobil saja. Dengan emtitas mobil Suzuki, mobil Toyota, mobil Honda dll.
c. Semua mahasiswa atau mahasiswa saja. Dengan entitas Ali, Budi, Iman dan seterusnya.

2.4.1.6 Atribut (Atributes / Properties)

Setiap Entitas pasti memiliki Atribut yang mendeskripsikan karakteristik (*property*) dari Entitas tersebut. Sebagaimana telah disebutkan sebelumnya, penentuan / pemilihan atribut – atribut yang relevan bagi sebuah entitas meru[akan hal penting lainnya dalam pembentukan model data. Penetapan atribut bagi sebuah entitas umunya memang didasarkan pada fakta yang ada.

Yang relevan untuk lebih diperhatikan dalam pembuatan model E-R adalah kedudukan atribut dalam entitas. Harus dapat kita bedakan / ketahui, mana atribut yang berfungsi sebagai *Primary Key* dan mana yang bukan (atribut deskriptif).

2.4.1.7 Hubungan / Relasi

Hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Relasi dapat digambarkan sebagai berikut :

Relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dalam satu basis data yaitu:

1. Satu ke satu *(One to one)*

   Hubungan relasi satu ke satu yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B.

2. Satu ke banyak *(One to many)*

   Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi setiap entitas pada entitas B dapat berhubungan dengan satu entitas pada himpunan entitas A.

3. Banyak ke Satu (*many to one*)

   Entitas pada himpunan A berhubungan dengan paling banyak dengan satu jenis entitas pada himpunan entitas B, tetapi tidak sebaliknya, dimana setiap entitas pada himpunan entitas A berhubungan dengan paling banyak satu entitas pada himpunan entitas B.

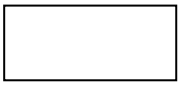
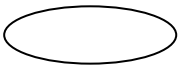
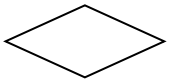
4. Banyak ke banyak *(Many to many)*

   Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B.

2.4.2 Diagram Entity – Relationship (ERD)

Model *Entity – Relational* yang berisi komponen-komponen Himpunan Entitas dan Himpunan Relasi yang masing – masing dilengkapi dengan atribut – atribut yang mempresentasikan seluruh fakta dari '*dunia nyata'* yang di tinjau, dapat digambarkan dengan lebih sistematis dengan menggunakan ERD (*Entity Relationship Diagram*). Notasi – notasi simbolik di dalam ERD yang dapat digunakan adalah :

Tabel 2. 1 Notasi – Notasi Simbolik ERD (Entity Relationship Diagram)

	Persegi panjang, ,menyatakan Himpunan Entitas.
	Elips menyatakan Atribut (Atribut yang berfungsi sebagai *key* digarisbawahi).
	Belah ketupat menyatakan Himpun Relasi.
	Garis sebagai penghubung antara Himpunan Relasi dengan Himpunana Entitas dan Himpunan Entitas dengan atributnya.
a. 1 dan 1 untuk relasi satu-ke-satu. b. 1 ke N untuk relasi satu-ke-banyak. c. N ke N untuk relasi banyak-ke-banyak.	Kardinalitas Relasi dapat dinyatakan dengan banyaknya garis cabang atau dengan pemakaian angka.

2.4.2.1 Varian Entitas dan Relasi

Idealnya himpunan entitas yang terlibat dalam sebuah ERD adalah himpunan entitas kuat atau bebas. Himpunan entitas demikian tidak memiliki ketergantungan dengan himpunan entitas lainnya. Namun demikian, dalam pembentukan ERD kita tidak selalu dapat melibatkan himpunan entitas seperti itu. Ada kalanya kita juga melibatkan himpunan entitas yang lemah atau merupakan bagian dari himpunan entitas lainnya.

2.5 DFD (Data Flow Diagram)

DFD adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan dan kemana tujuan data yang akan keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimapan dan proses yang dikenakan pada data tersebut. DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau dimana data tersebut akan disimpan.

DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur. Diagram yang menunjukkan bagaimana data berpindah/berjalan didalam sistem Informasi yang akan dikembangkan. Data flow diagram dapat menunjukkan ringkasan dari sistem yang luas/besar dari sistem input, proses dan output.

2.5.1 Type DFD (Data Flow Diagram)

Idealnya himpunan entitas yang terlibat dalam sebuah ERD adalah himpunan entitas kuat / bebas. Himpunan entitas demikian tidak memiliki ketergantungan dengan himpunan entitas lainnya. Namun demikian, dalam pembentukan ERD kita tidak selalu dapat melibatkan himpunan entitas seperti itu. Ada kalanya kita juga melibatkan himpunan entitas yang lemah atau merupakan bagian dari himpunan entitas lainnya.

2.5.1.1 Keuntungan DFD : Data Flow Diagram

a. Kebebasan dari menjalankan implementasi teknis sistem.

b. Pemahaman lebih jauh mengenai keterkaitan satu sama lain dalam sistem dan subsistem.

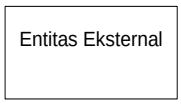
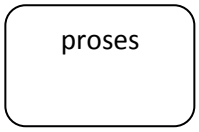
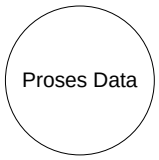
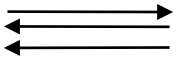
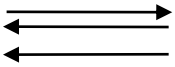
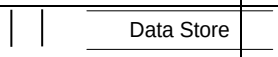
c. Mengkomunikasikan pengetahuan sistem yang ada dengan pengguna melalui diagram aliran data.

d. Menganalisis sistem yang diajukan untuk menentukan apakah data – data dan proses yang diperlukan sudah ditetapkan.

2.5.1.2 Simbol – simbol DFD

DFD terdiri dari empat simbol yaitu :

Tabel 2. 2 ***Simbol – symbol DFD***

Gane/Sarson	**Yourdon/De Marco**	**Keterangan**
		Entitas eksternal, dapat berupa orang/unit terkait yang berinteraksi dengan sistem tetapi diluar sistem
		Orang, unit yang mempergunakan atau melakukan transformasi data. Komponen fisik tidak didefinisikan.
		Aliran data dengan arah khusus dari sumber ke tujuan.
 		Penyimpanan data atau tempat data direfer oleh proses.

2.5.1.3 Jenis DFD

DFD terdiri dari *context diagram* dan *diagram rinci (DFD Levelled).* Context diagram berfungsi memetakan model lingkungan (menggambarkan hubungan antara entitas luar, masukan dan keluaran sistem), yang dipresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem.

DFD levelled menggambarkan sistem sebagai jaringan kerja antara fungsi yang berhubungan satu sama lain dengan aliran dan penyimpanan data, model ini hanya memodelkan sistem dari sudut pandang fungsi.

2.5.1.4 DFD : *Context* Diagram

Diagram yang menunjukan ikhtisar dari sistem informasi dan lingkup didalamya. Konteks Diagram hanya mempunyai satu proses. Diagram Konteks menunjukkan hanya 3 lambang :

a. Lambang proses
b. Lambang Kesatuan
c. Lambang Arus Data

Membuat DFD *context* diagram diantaranya ada beberapa point :

a. Menunjukkan context dalam proses bisnis yang dilakukan.
b. Menunjukkan global dari proses bisnis yang ditunjukkan hanya satu proses.
c. Menunjukkan semua external entities yang dibutuhkan informasinya atau yang berkontribusi ke dalam sistem.
d. Menunjukkan semua proses utama yang termasuk dalam sistem secara global komponen internal dalam *context* diagram.
e. Menunjukkan bagaimana proses utama dikaitkan dengan alur datanya (data flows).
f. Menunjukkan external entities dan proses utama dan interaksi antar keduanya Penambahan data proses.

2.5.1.5 DFD Levelled

Dalam DFD levelled akan terjadi penurunan level dimana dalam penurunan level yang lebih rendah harus mampu merepresentasikan proses tersebut ke dalam spesifikasi proses yang jelas. DFD levelled bisa dimulai dari DFD level 0, kemudian turun ke DFD level 1 dan seterusnya. Setiap penurunan hanya dilakukan bila perlu. Aliran data yang masuk dan keluar pada suatu proses di level x harus berhubungan dengan aliran data yang masuk dan keluar pada level x+1 yang mendefinisikan proses pada level x tersebut. Prosses yang tidak dapat

diturunkan/dirinci lagi dikatakan primitif secara fungsional dan disebut sebagai proses primitif.

Secara umum, Diagram level 1 dibuat dari setiap proses utama pada diagram level 0.

Menunjukkan semua internal proses termasuk proses tunggal pada diagram level 0. Menunjukkan semua internal proses termasuk proses tunggal pada diagram level 0. Menunjukkan bagaimana informasi mengalir dari dan ke setiap proses tersebut. Jika sebuah parent proses didekomposisikan menjadi 3 proses child misalnya, maka 3 proses child tersebut harus lengkap dan dibangun berdasarkan parent prosesnya.

2.5.1.5.1 Level 1 Diagrams

Menunjukkan semua proses yang termasuk pada single proses pada diagram level 1. Menunjukkan bagaimana sistem informasi mengalir dari dan ke setiap proses tersebut. Diagram level 2 boleh tidak dilakukan/dibuat pada setiap proses pada level 1. Penomoran secara benar pada setiap proses membantu user untuk memahami dimana proses tersebut terjadi pada keseluruhan sistem.

2.5.1.5.2 Level 2 Diagrams

Membuat context diagram. Membuat fragmentasi DFD untuk setiap case. Mengorganisasikan fragmentasi DFD ke dalam level 0 diagram. Dekomposisi proses pada level 0 ke dalam diagram level 1 jika diperlukan. Dekomposisi proses level 1 ke dalam diagram level 2 jika diperlukan. dst. Validasi DFD dengan uses untuk kelengkapan dan kebenaran proses dalam sistem.

2.5.1.6 Peraturan Pembuatan DFD

1. Antar entitas tidak diijinkan terjadi hubungan atau relasi.
2. Tidak boleh ada aliran data antara entitas eksternal dengan data store.
3. Untuk alasan kerapian (menghindari aliran data yang bersilangan), entitas eksternal atau data store boleh digambar beberapa kali dengan tanda khusus, misalnya diberi nomor.

4. Satu aliran data boleh mengalirkan beberapa paket data.

5. Bentuk anak panah aliran data boleh bervariasi.

6. Semua objek harus mempunyai nama.

7. Aliran data selalu diawali atau diakhir dengan proses.

8. Semua aliran data harus mempunyai tanda arah.

9. Jumlah proses tidak lebih dari sembilan proses dalam sistem, jika melebihi maka sebaiknya dikelompokkan beberapa proses yang bekerja bersama – sama didalam suatu subsistem.

2.5.1.7 Langkah – Langkah Pembuatan DFD

1. Membuat context diagram.

2. Membuat fragmentasi DFD untuk setiap case.

3. Mengorganisasikan fragmentasi DFD ke dalam level 0 diagram

4. Dekomposisi proses pada level 0 ke dalam diagram level 1 jika diperlukan, Dekomposisi proses level 1 ke dalam diagram level 2 jika diperlukan, dst.

5. Validasi DFD dengan uses untuk kelengkapan dan kebenaran proses dalam sistem.

6. Gambarkan satu proses untuk representasikan seluruh sistem (proses 0).

7. Inventarisasi semua input dan output dari sistem besarannya yang digambarkan sebagai data flow.

8. Gambarkan dalam eksternal entities untuk source atau destinasi dari data flow.

2.6 Software Pendukung

2.6.1 PHP (Personal Home Page)

PHP adalah bahasa pemorgraman yang memungkinkan para web developer untuk membuat aplikasi web yang dinamis dengan cepat. PHP merupakan singkatan dari "PHP: *Hypertext Preprocessor*". PHP ditulis dan diperkenalkan pertama kali sekitar tahun 1994 oleh Rasmus Lerdorf melalui situsnya untuk mengetahui siapa saja yang telah mengakses ringkasan *online*-nya. PHP merupakan salah satu bahasa script yang terbilang baru dan tersedia secara bebas dan masih memungkinkan untuk dikembangkan lebih lanjut. PHP dapat diintegrasikan *(embedde)* ke dalam web *server*, atau dapat berperan sebagai program CGI yang terpisah. Karakteristik yang paling unggul dan paling kuat dalam PHP adalah lapisan integrasi *database* (*database integration layer*). *Database* yang didukung PHP adalah: *Oracle, Adabas-D, Sybase, FilePro, mSQL, Velocis, MySQL, Informix, Solid, dBase, ODBC, Unix dbm, dan PostgreSQL*.
Tujuan utama penggunaan bahasa ini adalah untuk memungkinkan perancang web menulis halaman web dinamik dengan cepat.

2.6.1.1 Hubungan PHP dengan HTML

Halaman web biasanya disusun dari kode-kode html yang disimpan dalam sebuah file berekstensi .html. File html ini dikirimkan oleh *server* (atau file) ke *browser*, kemudian *browser* menerjemahkan kode-kode tersebut sehingga menghasilkan suatu tampilan yang indah. Lain halnya dengan program PHP, program ini harus diterjemahkan oleh *web-server* sehingga menghasilkan kode html yang dikirim ke *browser* agar dapat ditampilkan. Program ini dapat berdiri sendiri ataupun disisipkan di antara kode-kode html sehingga dapat langsung ditampilkan bersama dengan kode-kode html tersebut. Program PHP dapat ditambahkan dengan mengapit program tersebut di antara tanda . Tanda-tanda tersebut biasanya disebut tanda untuk *escaping* (kabur) dari kode html. File html yang telah dibubuhi program PHP harus diganti ekstensi-nya menjadi .PHP3 atau .PHP. PHP merupakan bahasa pemograman web yang bersifat *server-side* HTML = *embedded scripting*, di mana script-nya menyatu dengan HTML dan berada si

server. Artinya adalah sintaks dan perintah-perintah yang kita berikan akan sepenuhnya dijalankan di *server* tetapi disertakan HTML biasa. PHP dikenal sebgai bahasa scripting yang menyatu dengan tag HTML, dieksekusi di *server* dan digunakan untuk membuat halaman web yang dinamis seperti ASP (*Active Server Pages*) dan JSP (*Java Server Pages*).

2.6.1.2 Sejarah PHP

PHP pertama kali dibuat oleh Rasmus Lerdroft, seorang *programmer* C. Semula PHP digunakannya untuk menghitung jumlah pengunjung di dalam webnya. Kemudian ia mengeluarkan *Personal Home Page Tools* versi 1.0 secara gratis. Versi ini pertama kali keluar pada tahun 1995. Isinya adalah sekumpulan script PERL yang dibuatnya untuk membuat halaman webnya menjadi dinamis. Kemudian pada tahun 1996 ia mengeluarkan PHP versi 2.0 yang kemampuannya telah dapat mengakses *database* dan dapat terintegrasi dengan HTML. Pada tahun 1998 tepatnya pada tanggal 6 Juni 1998 keluarlah PHP versi 3.0 yang dikeluarkan oleh Rasmus sendiri bersama kelompok pengembang *softwarenya.*

Versi terbaru, yaitu PHP 4.0 keluar pada tanggal 22 Mei 2000 merupakan versi yang lebih lengkap lagi dibandingkan dengan versi sebelumnya. Perubahan yang paling mendasar pada PHP 4.0 adalah terintegrasinya Zend Engine yang dibuat oleh Zend Suraski dan Andi Gutmans yang merupakan penyempurnaan dari PHP *scripting engine*. Yang lainnya adalah *build in* HTTP *session*, tidak lagi menggunakan *library* tambahan seperti pada PHP. Tujuan dari bahasa *scripting* ini adalah untuk membuat aplikasi-aplikasi yang dijalankan di atas teknologi web. Dalam hal ini, aplikasi pada umumnya akan memberikan hasil pada web *browser*, tetapi prosesnya secara keseluruhan dijalankan web *server*.

2.6.1.3 Kelebihan PHP

Ketika *e-commerce* semakin berkembang, situs-situs yang statispun semakin ditinggalkan, karena dianggap sudah tidak memenuhi keinginan pasar, padahal situs tersebut harus tetap dinamis. Pada saat ini bahasa PERL dan CGI sudah jauh ketinggalan jaman sehingga sebagian besar designer web banyak beralih ke bahasa *server-side scripting* yang lebih dinamis seperti PHP. Seluruh

aplikasi berbasis web dapat dibuat dengan PHP. Namun kekuatan yang paling utama dari PHP adalah pada konektivitasnya dengan sistem *database* di dalam web. Sistem *database* yang dapat didukung oleh PHP adalah :

1. Oracle

2. SQL

3. Sybase

4. PostgreSQL

5. dan lainnya

2.6.1.4 *Script* PHP

Setiap program PHP disebut dengan script. Script berupa file text yang dapat dibuat dengan menggunakan program editor file text biasa seperti notepad, edit, dan lainya. Script PHP diawali dengan tag <? Dan diakhiri dengan tag ?>. Setiap baris atau statement harus diakhiri dengan menggunakan tanda titik koma (;) dan umumnya setiap statement dituliskan dalam satu baris. Script PHP merupakan script yang digunakan untuk menghasilkan halaman-halaman web. Cara penulisan script juga dibedakan menjadi 2 yaitu:

1. Embedded Script

Adalah script PHP yang disisipkan diantara tag-tag HTML. Script PHP digunakan apabila isi dari suatu dokumen HTML diinginkan dari hasil eksekusi suatu script PHP

```
<html>
<head>
<title>contoh</title>
</head>
<body>
<?php
echo “Website Mandiri Motor”;
?>
</body>
</html>
```

2. Non Embedded Script

Script PHP dalam *non embedded script* ini digunakan sebagai murni pembuatan program dengan PHP, tag HTML yang dihasilkan untuk membuat dokumen merupakan bagian dari script PHP.

2.6.1.5 Tipe Data PHP

PHP mengenal lima tipe data, yaitu : *integer, floating point, string, arrays* dan *Objects*. Tipe data dari sebuah variabel akan ditentukan otomatis oleh PHP bergantung pada operasi yang sedang dilakukan mengunakan variable tersebut.

1. Tipe Data *Integer*

   Tipe data *integer* meliputi semua bilangan bulat, range data *integer* pada PHP yaitu antara –2,147,483,648 samapai +2,147,483,647 pada platform 32 bit. PHP akan secara otomatis mengkonversi data *integer* menjadi *floating point* jika berada diluar range diatas.

2. Tipe Data *Floating Point*

   Floating point merepresentasikan bilangan pecahan atau bilangan desimal, range data *floating point* pada PHP yaitu antara 1.7E-308 sampai 1.7E+308.

3. Tipe Data *String*

   Sebuah data dengan tipe *string* dinyatakan dengan mengapitnya mengunakan tanda petik tunggal (‘ ‘) maupun ganda (“ “). Perbedaannya adalah, jika kita tanda petik tunggal, maka pada *string* itu tidak dapat dimasukkan variabel dan escape sequence handling.

4. Tipe Data *Array*

   Array adalah sebuah data yang mengandung satu atau lebih data, dan dapat diindeks berdasarkan numerik maupun *string*. Data yang dikandung pada sebuah *array* dapat dari tipe data yang lain bahkan dapat juga bertipe *array*. PHP membolehkan elemen dari *array* terdiri atas data dengan tipe yang berbeda-beda.

5. Tipe Data *Object*

Object adalah sebuah tipe data yang dapat berupa sebuah bilangan, variabel atau bahkan sebuah fungsi. *Object* dibuat dengan tujuan untuk membantu programmer yang terbiasa dengan *Object Oriented Programming*.

2.6.1.6 Struktur Kendali PHP

Struktur Kendali atau statement adalah bagian penting dalam suatu bahasa pemrograman, karena bagian ini mengatur jalurnya eksekusi suatu program. Dalam bahasa pemrograman PHP dikenal 4 jenis pernyataan utama, yaitu if, switch, while, dan for.

2.7 XAMPP

XAMPP Kepanjangan dari *XAMPP* yaiu *APACHE*, PHP, *MySQL* dan PHPMyAdmin. *XAMPP* merupakan tool yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dengan menginstall *XAMPP* maka tidak perlu lagi melakukan instalasi dan konfigurasi web *server APACHE*, PHP dan *MySQL* secara manual. *XAMPP* akan menginstalasi dan mengkonfigurasikannya secara otomatis untuk anda atau auto konfigurasi.

2.7.1 MySQL

Perkembangannya disebut SQL yang merupakan kepanjangan dari *Structured Query language*, SQL merupakan bahasa terstruktur yang khusus digunakan untuk mengolah *database*, SQL pertama kali didefinisikan oleh *American National Institute* (ANSI) pada tahun 1986. *MySQL* adalah sebuah sistem manajemen database yang bersifat open source, MySQL pasangan serasi dari PHP, MySQLdibuat dan dikembangkan oleh MySQL AB yang berada di swedia.

MySQL dapat digunakan untuk mengelola database mulai dari yang kecil sampai dengan yang besar, MySQLjuga dapat menjalankan perintah-perintah *Structured Query Language* (SQL) untuk mengelola database-database yang ada didalamnya, hingga kini MySQL sudah berkembang hingga versi 5, MySQL versi 5 sudah mendukung *trigger* untuk memudahkan pengelolaan tabel dalam *database*.

Sebuah website yang interaktif dan dinamis tentunya akan membutuhkan penyimpanan dara yang fleksibel dan cepat untuk diakses. Salah satu *database* untuk *sever* adalah MySQL, MySQL merupakan aplikasi RDBMS *(Relational Data Base Management System)* yang sangat cepat dan dapat diandalkan dalam menangani basis data. Jenis database ini sangat populer dan dapat digunakan pada banyak *website*, MySQL menggunakan bahasa SQL *(Structured Query Language)* yang bersifat *free* (gratis atau tidak perlu membayar untuk menggunakannya), selain itu MySQL dapat berjalan di berbagai platform diantaranya, Linux, Windows dan lain sebagainya.

Melakukan Koneksi kepada *database* MySQL dengan menggunakan PHP menggunakan dua tahap :

1. Koneksi ke *server* MySQL

   Dilakukan untuk melakukan otentikasi, digunakan untuk login ke dalam server menggunakan MySQL_connect().

2. Koneksi ke *database*

   Dilakukan untuk memilih database mana yang ada didalam server yang akan dimanipulasi, karena bisa saja database yang dimaksud sudah tidak ada. Fungsi yang digunakan adalah MySQL_select_db().

2.7.1.1 Pengertian MySQL

MySQL adalah perangkat lunak database server atau sebut saja *database smart,database* ini semakin lama semakin populer dengan menggunakan database ini, data semakin aman dan berdaya guna. *Database* ini juga banyal digunakan pada *web database* sehingga data semakin terintegrasi antara *database* desktop dengan *database web*, untuk menggunakan *database* MySQL kita harus menginstalasinya terlebih dahulu ke komputer.

2.7.1.2 keistimewaan MySQL

Sebagai server database dengan konsep *database modern*, MySQL memiliki keistimewaan. Beberapa keinstimewaanyang dimiliki MySQLsebagai berikut :

1. *Portability*

   Database MySQL berfungsi dengan stabil tanpa kendala, berarti berlaku pada berbagai sistem operasi seperti *Windows, Linux, FreeBSD, Mac OS X Server, Solaris, Amiga, HP-Unix* dan lain-lain.

2. *Open Source*

   MySQL merupakan *open source* (gratis), dibawah ini lisensi *GPL* sehingga dapat memperoleh dan menggunakannya secara Cuma-Cuma tanpa membayar sepersen pun.

3. *Multiuser*

   MySQL merupakan database yang dapat digunakan untuk menangani beberapa user dalam waktu bersamaan tanpa mengalami masalah dan memungkinkan sebuah database server MySQL dapat diakses *client* secara bersamaan pula.

4. *Performance Tunning*

   MySQL mempunyai kecepatan yang cukup baik dalam menangani query-query sederhana serta mampu memproses lebih banyak SQLper satuan waktu.

5. *Column Type*

   Database MySQL didukung dengan tipe data yang sangat kompleks, seperti signed atau unsiged integer, float, double, char, varchar, text, bolb, data, time, datetime, year, set serta enum.

6. *Command and Function*

   MySQL server memiliki operator dan berfungsi secara penuh yang mendukung perintah *SELECT* dan *WHERE* dalam *query*.

7. *Security*

   Sistem security pada *MySQL* mempunyai bebrapa lapisan sekuritas seperti tingkatan *subnetmask*, *hostname*, dan izin akses user dengan sistem perizinan yang mendetil serta *password* terenkripsi.

8. *Scalability and Limits*

   MySQL mempunyai kemampuan menangani database dalam skala yang cukup besar dengan jumlah *record* lebih dari 50 juta dan 60 ribu tabel serta

5 miliar baris, selain itu dapat menampung indeks sampau 32 indeks pada tiap tabelnya.

9. *Connectivity*

   Adanya kemampuan MySQL melakukan koneksi dengan *client* menggunakan protokol TCP/IP, Unix socket (Unix) atau Named Pipes (NT).

10. *Localization*

    Adanya kemampuan dalam mendeteksi kesalahan *(error code)* pada *client* menggunakan lebih dari dua puluh bahasa.

11. *Interface*

    MySQL mempunyai *interface* terhadap berbagai aplikasi dan bahasa pemrograman menggunakan fungsi API *(Application Programming Interface)*.

12. *Client dan Tools*

    Database MySQL dilengkapi dengan berbagai tools yang dapat digunakan untuk administrasi database.

13. *Struktur Table*

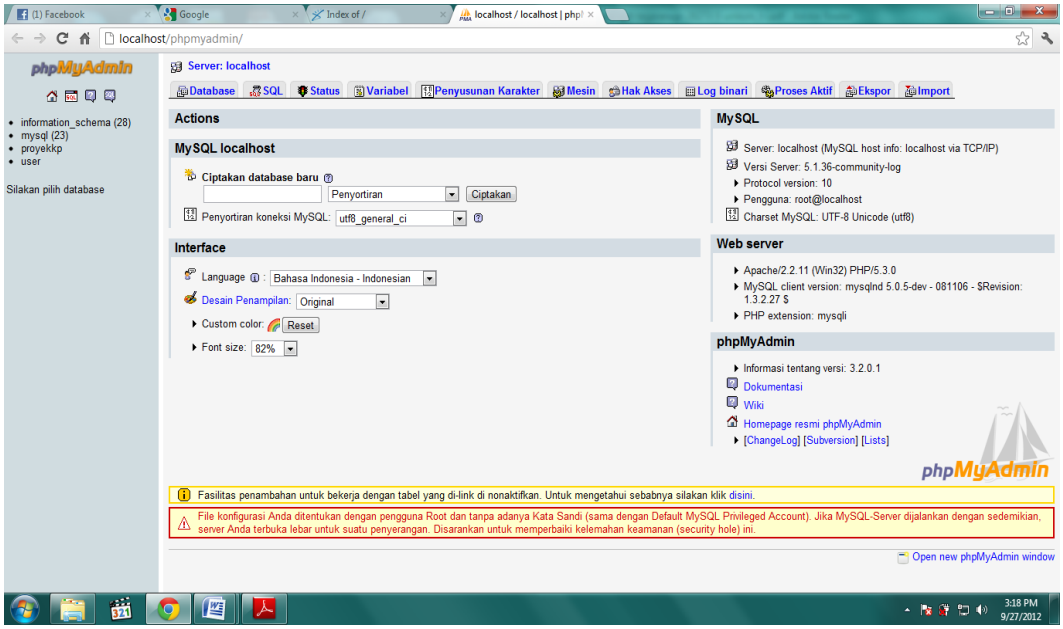
    MySQL mempunyai struktur tabel yang lebih flaksibel dalam menangani *ALTER TABLE* dibandingkan dengan database lainnya.

2.8 PHPMyAdmin

Pengolahan database dengan *MySQL* harus dilakukan dengan mengetikkan baris-baris perintah yang sesuai *(Commmand Line)* untuk setiap maksud tertentu. jika ingin membuat database pengguna harus mengetikkan baris perintah yang sesuai untuk database. Tetapi hal tersebut tentu cukup menyulitkan karena pengguna harus hafal dan mengetikkan perintahnya satu persatu.

Banyak sekali perangkat lunak yang dapat dimanfaatkan untuk mengelola *database* dalma *MySQL*, salah satunya adalah *PHPMyAdmin*. Dengan *PHPMyAdmin* pengguna dapat membuat table, mengisi data dan lain-lain dengan mudah tanpa harus hafal perintahnya. Untuk mengaktifkan PHPMyAdmin langkah-langkahnya adalah yang pertama setelah meng-*install Xampp*, pengguna harus mengaktifkan *web server Apache dan MySQL* dari control panel *Xampp*,

yang kedua jalankan di browser yang ada *(IE, Mozilla Firefox atau Google Chrome)* lalu aktifkan alamat web berikut : *http://localhost/phpmyadmin/* pada *adress* bar kemudian tekan *Enter*. Langkah selanjutnya apablia telah nampak *interface*(tampilan antara muka) phpmyadmin pengguna bisa memulainya dengan mengetikkan nama databasenya, nama tablenya dan seterusnya.



Gambar 2.3 Tampilan PhpMyAdmin

2.9 Adobe Macromedia Dreamveaver CS5

Adobe Macromedia Dreamweaver CS5 merupakan salah satu software dari kelompok macromedia yang banyak digunakan untuk mendesain website. Adapun *Adobe Macromedia Dreamweaver CS5* itu sendiri adalah sebuah HTML editor professional untuk mendesain secara visual dalam mengolah halaman *website*. *Dreamweaver CS5* memiliki performa yang lebih baik dan memiliki tampilan yang memudahkan anda untuk membuat website, baik dalam jendela desain maupun dalam jendela kode rumus, *Dreamwaver CS5* didukung dengan pemakaian yang praktis dan standar dan juga didukung untuk pengembangan pengguna *CSS, XML dan RSS*, dan kemudahan-kemudahan lain yang diperlukan.

Adobe Macromedia Dreamweaver CS5 merupakan software yang digunakan oleh desainer maupun web programmer dalam mengembangkan websitenya, hal ini disebabkan ruang kerja, fasilitas dan kemampuan *Macromedia*

Dreamweaver CS5 yang mampu mengingkatkan produktivitas dan efektivitas dalam mendesain maupun membangun sebuah *website*



Gambar 2.4 tampilan Adobe Macromedia Dreamweaver

Menu Panel Media adalah sekumpulan ikon yang berfungsi untuk melakukan penggabungan file dari luar program Macromedia Dreamweaver.

1. Menu Panel Head

   Menu Panel Head digunakan untuk operasi penambahan kompenen dalam head HTML.

2. Menu Panel Script

   Menu Panel Script adalah sekumpulan ikon yang digunakan untuk melakukan penulisan perintah script.

3. Menu Panel Application

   Menu Panel Application digunakan untuk operasi database.

4. Menu Panel Properties

   Menu Panel Properties adalah sekumpulan menu yang berfungsi sebagai tool dalam pengaturan objek yang ada pada halaman web.