BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Umum Instansi

SMA Negeri 2 subang merupakan salah satu sekolah lanjutan tingkat atas (SLTA) yang terletak di Jalan Dangdeur Km 5 Kelurahan dangdeur Kabupaten Subang. SMA Negeri 2 Subang pada awal berdirinya merupakan Sekolah Menengah Pembangunan Persiapan (SMPP) yang pembukaannya didasarkan pada Surat Keputusan Materi Pendidikan dan Kebudayaan Nomor 0236/O/1973 tanggal 18 Desember 1973.

SMA Negeri 2 Subang menempati tanah seluas 50000 m^2 dengan perincian 3741 m^2 digunakan untuk bangunan, 8270 m^2 dipakai sebagai sarana olahraga, 9012 m^2 berupa halaman dan taman dan sisanya 25537 m^2 berupa tanah kebun.

2.1.1 Visi dan Misi Organisasi

a. Visi sekolah

   Unggul dalam prestasi, beriman, trampil, dan berbudaya.

   Indikator :

   1) Unggul dalam prestasi akademik
   2) Unggul dalam aplikasi Ilmu Pengetahuan
   3) Unggul dalam disiplin

4) Unggul dalam kegiatan seni dan olahraga
5) Unggul dalam aktivitas keagamaan

b. Misi sekolah

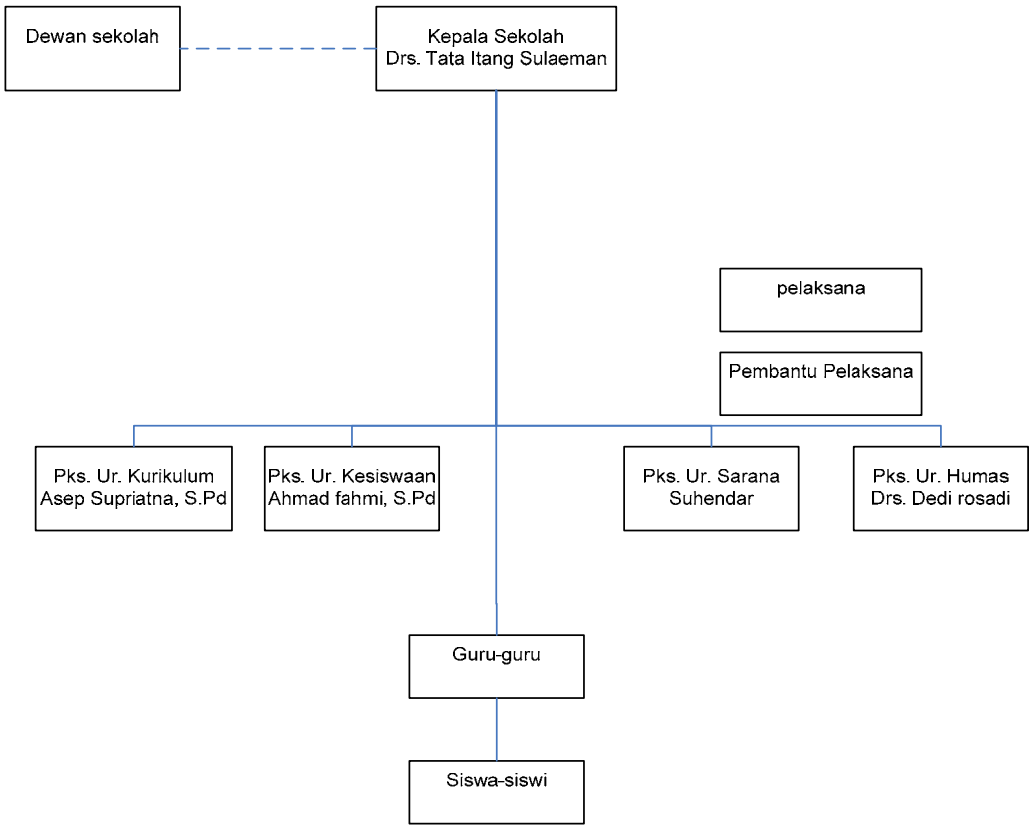
1) Melaksanakan kegiatan belajar mengajar secara efektif dengan mengarah kepada pemberdayaan siswa dalam mengembangkan potensi yang dimiliki,
2) Menumbuhkan motivasi/semangat unggul kepada seluruh warga sekolah,
3) Menumbuhkan kemampuan berkreativitas dalam kegiatan ilmiah remaja, seni dan olahraga dengan didasari disiplin yang tinggi,
4) Menumbuhkan penghayatan terhadap ajaran agama, serta budaya bangsa yang menjadikan manusia berbudi pekerti baik, sehingga menjadi sumber kearifan dalam pergaulan maupun dalam bertindak,
5) Menerapkan manajemen partisipatif yang melibatkan seluruh warga sekolah, masyarakat sekitad serta orang tua siswa.

Dilihat dari prospek masa depannya merupakan salah satu kebanggaan dan sekaligus tumpuan harapan masyarakat Subang dalam mempersiapkan generasi muda yang cerdas, tangguh, terampil, dalam menggapai masa depannya. Harapan masyarakat tersebut bukanlah harapan kosong melainkan sebagai realita SMA Negeri 2 Subang yang memiliki integritas yang kuat dan komitmen yang tinggi dalam membina dan mempersiapkan menjadi generasi yang potensial dalam sumber daya

manusia-nya. Maka semua pihak terkait terutama sekolah harus mampu mempersiapkan diri dengan program-program yang rasional, universal, dan realitis sebanding dengan kemauan dan kemampuan yang tersedia.

2.1.2 Struktur Organisasi

Untuk lebih jelasnya akan digambarkan mengenai struktur organisasi di SMA Negeri 2 Subang dapat dilihat pada gambar 2.1.



Gambar 2.1 : Stuktur Organisasi Sekolah

a. Tugas Dewan Sekolah yaitu menilai kemajuan sekolah

b. Tugas Kepala Sekolah

    1) Sebagai penanggung jawab pelaksanaan pendidikan di sekolah termasuk dalam pelaksanaan administrasi sekolah.
    2) Mengatur kegiatan sekolah dengan mengeluarkan atau memberikan intruksi terhadap kemajuan sekolah
    3) Penyusunan program kegiatan sesuai dengan kalender pendidikan

c. Tugas Pelaksana yaitu melaksanakan program perencanaan proses belajar

d. Tugas Pembantu pelaksana yaitu sebagai pembantu pelaksanaan dalam program perencanaan membangun sekolah.

e. Tugas Pembantu Kepala sekolah (PKS)

    1) Membuat program tahunan tentang proses kegiatan belajar mengajar selama satu tahun pelajaran.
    2) Membantu kepala sekolah dalam mengatur kepanitiaan kegiatan, penerimaan siswa, kebutuhan sekolah, guru dan kegiatan belajar mengajar.
    3) Membantu mengembangkan bakat baik akademis maupun non akademis.

f. Tugas Guru yaitu bertanggung jawab kepada kepala sekolah dan mempunyai tugas melaksanakan proses belajar mengajar secara efektif dan efisien.

2.2 Landasan Teori

Pada landasan teori ini diuraikan sejumlah teori untuk membantu dan memecahkan permasalahan yang ada. Beberapa landasan teori tersebut meliputi konsep dasar dan definisi-definisi yang berkaitan dengan perangkat lunak yang digunakan serta faktor-faktor pendukung dalam pelaksanaan perancangan.

2.2.1 Pengolahan Data

Pengolahan data dengan mengunakan komputer dikenal dengan sebutan Pengolahan Data Elektronik (PDE) atau *Electronic Data Processing (EDP).*

Data adalah kumpulan kejadian yang diambil dari suatu fakta dan data dapat berupa angka-angka, huruf-huruf atau simbol-simbol khusus atau gabungan dari kesemuanya. Suatu data mentah adalah merupakan data yang belum dapat memberi informasi yang banyak dan lengkap, sehingga perlu diolah terlebih dahulu.

Pengolahan data adalah manipulasi dari kumpulan data-data kedalam bentuk yang lebih berguna dan lebih berarti, yaitu merupakan suatu informasi. Informasi yang dihasilkan ini harus bersifat komunikatif dalam interaktif atau dengan kata lain bahwa informasi harus dapat dimengerti secara jelas dan lengkap oleh penerima informasi.

2.2.2 Sistem Pengolahan Data

Data perlu diolah dengan baik agar dapat dihasilkan informasi yang *up to date*, akurat, tepat waktu, relevan dan ekonomis.

Menurut John Longkutoy, dalam suatu sistem pengolahan data harus ada kegiatan-kegiatan seperti dibawah ini :

1) Pengumpulan data
2) Verifikasi data (*checking*)
3) Penyimpanan data
4) Pengambilan atau pembacaan data tersimpan
5) Pengolahan data (kalkulasi)
6) Pembuatan laporan
7) Penyebaran laporan

Data yang baik mempunyai karakteristik seperti tersebut diatas (*current, accurate, timely, relevant, economically*). Data seperti itu hanya bisa dihasilkan oleh sistem manajemen *database* yang baik dan dipengaruhi oleh sistem manajemen data yang baik pula. Sistem manajemen *database* yang baik dipengaruhi oleh *database* yang baik pula.

2.2.3 Definisi Sistem dan Informasi

Suatu organisasi dalam setiap kegiatannya pasti mempunyai tujuan yang ingin dicapai. Untuk mencapai tujuan organisasi tersebut, setiap organisasi tentu saja memiliki suatu sistem tersendiri.

Suatu sistem terdiri dari elemen-elemen yang berhubungan melalui berbagai bentuk interaksi dan bekerja sama untuk mencapai tujuan tertentu. Elemen-elemen tersebut mempunyai sifat dan di dalam menjalankan fungsinya tidak dapat dipisahkan antara satu dengan yang lain akan saling berhubungan dan saling mempengaruhi proses serta secara keseluruhan.

Dalam memahami suatu sistem diperlukan pemahaman mengenai sistem itu sendiri, terdapat beberapa pendekatan dalam mendefinisikan sistem, antara lain yang menekankan pada prosedurnya dan yang menekankan pada komponen atau elemen.

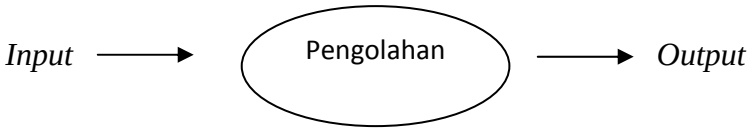
Pendekatan sistem yang lebih menekankan pada prosedur, mendefinisikan sistem sebagai berikut: "*Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu tujuan tertentu"*. (Jogiyanto, 2000 : 684).

Pendekatan sistem yang lebih menekankan pada elemen atau komponennya, mendefinisikan sistem sebagai berikut : *"Sistem adalah sekelompok elemen-elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan"*. (Mcleod Jr, 2001 : 9).

Pendekatan sistem yang menekankan pada komponen akan lebih mudah di dalam mempelajari suatu sistem untuk tujuan analisis dan perancangan suatu sistem.

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu mempunyai komponen-komponen *(components)*, batas sistem *(boundary)*, lingkungan luar sistem *(environment)*, penghubung *(Interprest)*, masukan *(input)*, keluaran *(output)*, pengolah *(Process)* dan sasaran *(objective)* dan tujuan *(goal)*.

Model dasar sebuah sistem yang paling sederhana terdiri dari masukan, pengolahan dan keluaran.



Gambar 2.2 Model Dasar Sebuah Sistem

Informasi sangat penting bagi suatu sistem. *"Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam mengambil keputusan saat ini atau mendatang"* (Davis, 1995 : 28). Sumber dari informasi adalah data.

Data merupakan bentuk yang masih mentah yang belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut. Data diolah melalui suatu model untuk menghasilkan informasi. Data yang diolah melalui suatu model menjadi informasi, akan diterima oleh penerima informasi tersebut, membuat keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sebuah data kembali.

Kualitas dari suatu informasi dari sistem informasi tergantung dari tiga hal, yaitu :

1. Relevansi

   Informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk setiap orang berbeda satu dengan yang lainnya.

2. Akurasi

   Informasi harus terbebas dari kesalahan-kesalahan dan tidak menyesatkan. Akurat dapat berarti informasi harus mencerminkan maksudnya. Informasi dari sumber informasi harus akurat sampai ke penerima informasi dikarenakan kemungkinan banyak terjadi gangguan yang dapat merubah atau merusak informasi tersebut.

3. Ketepatan Waktu

   Informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan didalam pengambilan keputusan. Bila pengambilan keputusan terlambat, maka dapat berakibat fatal untuk organisasi.

Nilai dari informasi ditentukan dari dua hal, yaitu manfaat dan biaya mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya.

2.2.4 Konsep Dasar Basis Data

Basis data merupakan sekumpulan informasi yang diatur dalam cara tertentu sehingga sebuah program komputer dapat dengan cepat memilih data yang diinginkan.

2.2.4.1 Pengertian Basis Data

Basis data terdiri atas dua kata, yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang dan berkumpul. Sedangkan data adalah fakta yang terungkap atau 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. Jadi basis data adalah sekumpulan data tentang suatu benda atau kejadian yang saling berhubungan satu sama lainnya.

2.2.4.2 Hirarki Data

Secara tradisional data diorganisasikan kedalam suatu hirarki yang terdiri atas elemen data, rekaman *(record)* dan berkas *(file).*

a. *Entity*

Adalah orang tempat kejadian atau konsep yang informasinya direkam. Pada bidang administrasi mahasiswa misalnya, *entity* adalah mahasiswa, buku pembayaran dan nilai ujian. Pada bidang kesehatan *entity* adalah pasien, dokter, obat, kamar.

b. *Atribute* atau elemen data

Setiap *entity* mempunyai atribut atau sebutan untuk mewakili suatu *entity*. Seorang mahasiswa dapat dilihat atributnya, misalnya nama, NIM, alamat, nama orang tua, agama dan jenis kelamin. Atribut juga disebut sebagai data elemen, data *field, item* data atau medan.

c. *Data value* atau isi

Adalah data aktual atau informasi yang disimpan pada tiap elemen data atau atribut. Atribut nama karyawan akan menunjukkan tempat dimana informasi nama karyawan disimpan, sedangkan *data value* adalah bejo, mitro yang merupakan isi dari data nama karyawan tersebut.

d. *Record* atau rekaman

Adalah gabungan dan kumpulan sejumlah elemen - elemen data yang saling berkaitan menginformasikan tentang *entity* secara lengkap. Satu *record* akan mewakili satu data atau informasi yang mewakili tentang seseorang atau objek lain misalnya NIS, Nama, Alamat, Nomor telepon, *email* dan sebagainya.

e. *File* atau berkas

Adalah kumpulan rekaman data yang berkaitan dengan subjek data.

2.2.4.3 RDBMS (*Relational Database Management System*)

Merupakan sekumpulan data yang disimpan sedemikian rupa sehingga mudah diambil informasinya bagi pengguna dan data tersebut saling berhubungan. RDBMS

merupakan suatu paket perangkat lunak yang kompleks yang digunakan untuk memanipulasi *database.* Ada tiga prinsip RDBMS :

1. Bahasa Definisi Data (DDL atau *data definition language*)

DDL adalah perintah-perintah yang biasa digunakan oleh administrator basis data (DBA) untuk mendefinisikan skema ke DBMS. Skema adalah deskripsi lengkap tentang struktur medan, rekaman, dan hubungan data pada basis data. Tugas utama skema adalah menjabarkan struktur basis data kepada DBMS.

2. Bahasa Manipulasi Data (DML atau *data manipulation language*)

DML adalah perintah-perintah yang digunakan untuk mengubah, memanipulasi, dan mengambil data pada basis data. DML pada dasarnya dibagi menjadi dua, yaitu *prosedural* dan *nonprosedural*. *Prosedural* menuntut pengguna menentukan data apa saja yang diperlukan dan bagaimana cara mendapatkannya. *Nonprocedural* menuntut pengguna data apa saja yang diperlukan, tetapi tidak perlu menyebutkan cara mendapatkannya.

3. *Data Control Language (DCL)*

Bagian ini berhubungan dengan cara mengendalikan data, seperti siapa saja yang bisa melihat isi data, bagaimana data bisa digunakan oleh banyak *user*, dan sebagainya. Dengan kata lain DLC digunakan untuk hal-hal yang berhubungan dengan sekuritas terhadap basis data. Semua operasi masukan dan keluaran yang berhubungan dengan basis data harus menggunakan DBMS. Bila pemakai akan

mengakses *database*, DBMS menyediakan penghubung *(interface)* antara pemakai dengan basis data.

2.2.5 Analisis Sistem

Penggambaran dan analsis model sistem secara logika dapat dibuat dalam bentuk Flowchart, Diagram E-R, dan Diagram Alir Data (DAD) atau *Data Flow Diagram* (DFD). Sedangkan arus data pada Diagram Alir Data atau *Data Flow Diagram* dapat dijabarkan dengan menggunakan kamus data (*Data Dictionary*).

2.2.5.1 Diagram Konteks

Diagram konteks merupakan arus data yang berfungsi untuk menggambarkan keterkaitan aliran-aliran data antar sistem dengan bagian luar (kesatuan luar). Kesatuan luar ini merupakan sumber arus data atau tujuan data yang berhubungan dengan sistem informasi tersebut.

2.2.5.2 Diagram Aliran Data

Diagram Alir Data atau *Data Flow Diagram* (DFD) adalah suatu model yang menjelaskan arus data mulai dari pemasukan sampai dengan keluaran data. Tingkatan DFD dimulai dari diagram konteks yang menjelaskan secara umum suatu sistem atau batasan sistem aplikasi yang akan dikembangkan. Kemudian DFD dikembangkan menjadi DFD tingkat 0 atau level 0 dan kemudian DFD level 0 dikembangkan lagi

menjadi level 1 dan selanjutnya sampai sistem tersebut tergambarkan secara rinci menjadi tingkatan-tingkatan lebih rendah lagi.

DFD merupakan penurunan atau penjabaran dari diagram konteks. Dalam pembuatan DFD harus mengacu pada ketentuan sebagai berikut :

1) Setiap penurunan *level* yang lebih rendah harus mempresentasikan proses tersebut dalam spesifikasi proses yang jelas.
2) Penurunan dilakukan apabila memang diperlukan.
3) Tidak semua bagian dari sistem harus ditunjukkan dengan jumlah *level* yang sama.

Simbol-simbol yang digunakan pada diagram alir data atau *data flow diagram* antara lain :

a. Entitas luar (*external entity*)

   Sesuatu yang berada di luar sistem, tetapi memberikan data ke dalam sistem atau memberikan data dari sistem, disimbolkan dengan suatu kotak notasi. *External entity* tidak termasuk bagian dari sistem bila sistem informasi dirancang untuk satu bagian maka bagian lain yang masih terkait menjadi *external entity*.

b. Arus data (*data flow*)

   Arus data merupakan tempat mengalirnya informasi dan digambarkan dengan garis yang menghubungkan komponen dari sistem. Arus data ditunjukan dengan arah panah dan garis diberi nama atau arus data yang

mengalir. Arus data ini mengalir diantara proses, data *store* dan menunjukan arus data dari data yang berupa masukan untuk sistem atau hasil proses sistem.

c. Proses (*proccess*)

Proses merupakan apa yang dikerjakan oleh sistem. Proses dapat mengolah data atau aliran data masuk menjadi aliran data keluar. Proses berfungsi mentransformasikan satu atau beberapa data keluaran sesuai dengan spesifikasi yang diinginkan. Setiap proses memiliki satu atau beberapa masukan serta menghasilkan satu atau beberapa data keluaran. Proses sering juga disebut *bubble*.

d. Simpanan data (*data store*)

Simpanan data merupakan tempat penyimpanan data yang ada dalam sistem. *Data store* dapat disimbolkan dengan dua garis sejajar atau dua garis dengan salah satu sisi samping terbuka. Proses dapat mengambli data dari atau memberikan data ke *database*.

2.2.5.3 Kamus Data

Kamus data (KD) atau *data dictionary* (DD) atau disebut juga dengan istilah *systems data dictionary* adalah katalog fakta tentang data dan kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan KD, analisis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap. KD dibuat pada tahap

analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem. Pada tahap analisis, KD dapat digunakan sebagai alat komunikasi antara analisis sistem dengan pemakai sistem tentang data yang mengalir di sistem, yaitu tentang data yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh pemakai sistem. Pada tahap perancangan sistem, KD digunakan untuk merancang input, merancang laporau-laporan dan *database*. KD dibuat berdasarkan arus data yang ada di DFD. Arus data di DFD sifatnya adalah global, hanya ditunjukkan nama arus datanya saja.

Biasanya untuk menunjukkan informasi-informasi tambahan ini di KD dipergunakan notasi-notasi sebagai berikut ini :

= : terdiri dari m{ }n : iterasi min. m kali, max. n kali

\+ : dan

( ) : optional

[ ] : salah satu dari

@ : identifikasi atribut kunci

| : sama dengan [ ]

* : komentar/catatan

Dengan menggunakan kamus data, analisis sistem dapat memberikan informasi mengenai definisi struktur pemakaian masing-masing elemen, dapat mendefinisikan data yang mengalir di sistem dengan lengkap, dapat menghindari duplikasi elemen-elemen dan menghindari konflik antara elemen-elemen.

2.2.5.4 *Entity Relationship Diagram* (ERD)

Model entitas didasarkan pada persepsi dunia nyata yang terdiri dari himpunan objek dasar disebut dengan entitas dan relasi antar entitas. Entitas adalah objek yang dapat diindentifikasikan secara unik.

Diagram E-R yang berisi komponen-komponen entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang merepresentasikan seluruh fakta dari yang kita tinjau, dapat digambarkan dengan lebih sistematis dengan menggunakan diagram E-R

Model data diagram E-R dibentuk dari empat komponen dasar, yaitu :

1) Entitas (*Entity*)

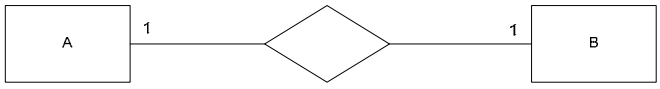
Entitas adalah sesuatu yang ada dan dapat berupa orang, tempat, benda, peristiwa atau konsep yang bisa memberi atau mengandung informasi

2). Relasi (*Relationship*)

Jenis hubungan untuk entitas yang ada di dalam diagram adalah sebagai berikut :

a. *One to One Relationship* (Satu ke Satu)

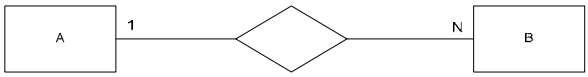
Setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B, dan begitu pula sebaliknya.



Gambar 2.3 *One to One Relationship*

b. *One to Many Relationship* (Satu ke Banyak)

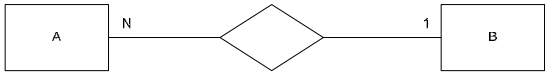
Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya dimana setiap entitas pada himpunan B berhubungan paling banyak dengan satu entitas pada entitas A.



Gambar 2.4 *One to Many Relationship*

c. *Many to One Relationship* (Banyak ke Satu)

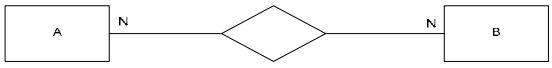
Setiap entitas pada himpunan A berhubungan paling banyak satu entitas pada himpunan entitas B, tetapi tidak sebaliknya dimana setiap entitas pada himpunan entitas B berhubungan dengan banyak entitas pada entitas himpunan B.



Gambar 2.5 *Many to One Relationship*

d. *Many to Many Relationship* (Banyak ke Banyak)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, demikian pula sebaliknya.



Gambar 2.6 *Many to Many Relationship*

3). Atribut

Setiap entitas memiliki atribut-atribut atau elemen data yang mencirikan entitas tersebut.

4). *Key* (Kunci)

Sebuah atribut atau set atribut yang nilainya mengidentifikasikan entitas secara unik dalam set entitas. Key ini terdiri dari *Primary Key* dan *Foreign Key*.

a. *Primary Key*

Primary Key bertugas untuk menentukan nilai yang berbeda pada setiap kolom, dan terdapat hanya disatu kolom dari banyak kolom yang ada. Maka tidak akan ada dua nilai yang sama di dua kolom yang berbeda, tetapi ini tergantung dari desain biasanya dalam satu kumpulan kolom bisa terdapat beberapa *Primary Key*, tetapi biasanya hanya terdapat satu saja.

b. *Foreign key*

Sedangkan yang dimaksud dengan *Foreign key* adalah *Primary Key* yang ada pada tabel lain yang akan dihubungkan dengan tabel yang lainnya, dikarenakan datanya saling berhubungan.

2.2.6 Teori Dasar *Internet*

Internet adalah suatu jaringan global yang terbentuk dari jaringan-jaringan komputer lokal dan regional, memungkinkan komunikasi antar komputer-komputer yang terhubung ke jaringan tersebut.

2.2.6.1 *Web*

Web merupakan suatu satu layanan yang di dapat oleh pemakai komputer yang terhubung ke *internet*. *Web* pada awalnya adalah ruang informasi dalam *internet* menggunakan teknologi *Hypertext*. Pemakai dituntun untuk menemukan informasi dengan mengikuti link yang disediakan dalam dokumen *web* yang ditampilkan dalam browser *web*.

Web memudahkan pengguna komputer untuk berinteraksi dengan pelaku internet lainnya dan menelusuri (informasi) di *internet*. Selain itu juga memiliki kemudahan-kemudahan lainnya seperti :

1. Akses informasi mudah
2. *Setup server* lebih mudah
3. Informasi mudah didistribusikan, dan
4. Bebas *platform*, informasi dapat disajikan oleh *browser web* pada sistem operasi mana saja, karena adanya standar dokumen berbagai tipe data dapat disajikan.

Protokol yang digunakan adalah protokol TCP/IP yang merupakan protokol yang dipakai dalam komunikasi *internet*.

2..2.6.2 *Universal Resource Locator* (URL)

URL adalah konsep nama *file* standar yang diperluas dengan jaringannya. Nama *file* ini tidak hanya menunjukan direktori dan nama *file*-nya saja, tetapi juga

nama mesinnya dalam jaringan. URL dapat disediakan (ada atau diakses) dengan berbagai metode dan bisa jadi bukan sekedar *file*. Karena URL dapat menunjukan *query-query* dokumen dapat disimpan dalam *database*, hasil dari suatu perintah *finger* atau *achie* atau apapun yang berkaitan dengan data hasil proses.

2.2.6.3 *Domain Name System* (DNS)

DNS digunakan untuk mempermudah penggunaan *internet*, dengan memetakan IP *address* ke nama *host.* Agar data nama *host* dapat didistribusikan di banyak *server*, format data yang digunakan harus mencerminkan terdistribusinya data tersebut. Untuk itu, digunakan format *tree* dengan masing-masing nodenya disebut domain. Penulisan nama *host* secara lengkap disebut sebagai *Full Qualifield Domain Name (FQDN).*

DNS bekerja secara hirarki dan berbentuk seperti pohon (*tree*). Bagian atas adalah *Top Level Domain* (TLD) seperti COM, ORG, EDU, MIL dsb. Seperti pohon DNS mempunyai cabang-cabang yang dicari dari pangkal sampai ke ujung. Pada waktu kita mencari alamat misalnya linux.or.id pertama-tama DNS bertanya pada TLD *server* tentang DNS *Server* yang melayani domain .id misalnya dijawab ns1.id, setelah itu dia bertanya pada ns1.id tentang DNS *Server* yang bertanggung jawab atas .or.id misalnya ns.or.id kemudian dia bertanya pada ns.or.id tentang linux.or.id dan dijawab 64.29.24.175.

2.2.6.4 *World Wide Web*

Awalnya *internet* dimulai pada masa di mana orang masih menggunakan alat-alat akses yang tidak *user-friendly* yaitu terminal berbasis teks serta perintah-perintah *command line* yang panjang-panjang serta sukar di ingat, sangat berbeda dengan komputer dewasa ini. Kemudian orang mulai berfikir sesuatu yang lebih baik. *Internet* menggunakan standar baru yaitu HTTP dan HTML diperkenalkan pada masyarakat. Pemunculan HTTP dan HTML kemudian membuat orang mengenal istilah baru dalam *internet* yang sekarang menjadi populer bahkan sedemikian populernya sering di anggap identik dengan *internet* itu sendiri yaitu *Worl Wide Web* (WWW).

2.2.6.5 HTTP (*Hypertext Transfer Protokol*)

Server dan *browser web* berkomunikasi satu sama lain dengan protokol yang memang dibuat khusus untuk ini yaitu HTTP. HTTP bertugas menangani permintaan-permintaan (*request*) dan *browser* untuk mengambil dokumen-dokumen *web*. HTTP bisa dianggap sebagai sistem yang bermodel *client-server*.

2.2.7 *Hypertext Markup Language* (HTML)

Homepage dibangun oleh sebuah *file* yang didalamnya terdapat kode-kode dan informasi dan dibaca oleh sebuah aplikasi yang bernama *browser*. Oleh *browser*, kode dan informasi akan disajikan sebagai sebuah halaman *web* yang kita kenal

sebagai *homepage*. Kode tersebut haruslah ditulis menurut suatu aturan tertentu yang dapat dimengerti *browser*. Aturan tersebut dikenal dengan “bahasa pengkodean”.

Untuk membuat *homepage website*, bahasa pengkodean yang digunakan adalah HTML. Sekalipun banyak orang menyebutnya sebagai salah satau bahasa pemrograman, HTML sebenarnya sama sekali bukan bahasa pemrograman karena HTML merupakan bahasa Mark Up (pengkodean) yang digunakan untuk menentukan format atau *style* dan teks yang ditandai. Simbol *Mark Up* yang digunakan oleh HTML ditandai dengan tanda lebih kecil (<) dan lebih besar (>), keduanya dinamai tag. Sebagai contoh akan ditampilkan teks yang tercetak tebal, maka mark up yang digunakan adalah sebagai berikut:

<B> Teks ini bercetak tebal </B>

Untuk menandai bahwa sebuah *file* teks merupakan file HTML, maka ciri yang terlihat jelas adalah ekstensi *file*-nya yaitu .htm atau .html. Namun labih jauh dari pada itu didalam *file* tersebut harus mengandung struktur sebagai berikut:

<HTML>

…….

…….

</HTML>

Tag <HTML> harus diletakan pada bagian awal dan tag </HTML> harus diletakan pada bagian terakhir dari suatu *file* HTML. Tag-tag HTML tidak bersifat *case sensitive* artinya penggunaan huruf besar maupun huruf kecil tidak menjadi masalah. Jadi tag <HTML> akan sama dengan tag <html>.

2.2.8 *Hypertext Pro Pecessor* (PHP)

PHP adalah sebuah bahasa *scripting* yang dibundel dengan HTML, yang berjalan disisi *server*. Sebagian besar perintahnya berasal dari bahasa *C, Java* dan *Perl* dengan beberapa tambahan fungsi khusus PHP.

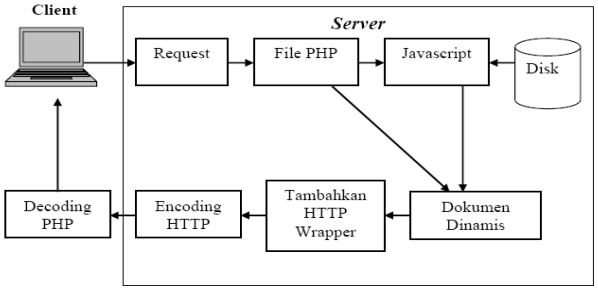
Bahasa ini memungkinkan para pembuat aplikasi *web* yang menyajikan HTML yang dinamis dan interaktif dengan cepat dan mudah, yang di hasilkan *server*. PHP biasa berinteraksi dengan hampir semua teknologi *web* yang sudah ada. *Developer* biasa menulis sebuah program PHP yang mengeksekusi suatu program CGI di *server* lain. Fleksibilitas ini amat bermanfaat bagi pemilik situs-situs *web* yang besar dan sibuk, karena pemilik masih menggunakan aplikasi-aplikasi yang sudah terlanjur dibuat dimasa lalu dengan menggunakan CGI, ISAP atau dengan *script* seperti *Perl, awk* atau *Phyton* selama proses migrasi ke aplikasi baru yang di buat dengan menggunakan PHP.

Kode program PHP menyatu dengan tag-tag HTML dalam satu *file*. Kode PHP diawali dengan tag <? Dan ditutup dengan tag ?>. File yang berisi tag HTML dank ode PHP ini diberi ekstensi .php atau ekstensi lainnya yang ditetapkan pada Apache/*web server*. Berdasarkan ektensi ini, pada saat *file* diakses, *server* akan tahu bahwa *file* ini mengandung kode PHP. *Server* akan menerjemahkan kode ini dan menghasilkan *output* dalam bentuk tag HTML yang akan dikirim ke *browser client* yang mengakses *file* tersebut. Contohnya dapat dituliskan seperti ini :

```
<?
echo (“<html>
```

```
<head>
<title>Pemrograman PHP</title>
</head>
<body>
<center>Selamat menggunakan PHP</center>
</body>
</html>”);
?>
```

Dengan PHP, *developer* tidak perlu lagi berurusan dengan dua buah *file* yang terpisah seperti pada CGI. *Browser web* mengacu secara langsung ke *file* yang dituju, yang lalu dibaca oleh *server* sebagaimana file HTML statis biasa. Bedanya, sebelum dikirim balik ke *browser web*, *server web* memeriksa isi *file* dan menentukan apakah ada kode didalam *file* tersebut yang harus dieksekusi. Bila ada, kode-kode tersebut akan dieksekusi. Hasilnya akan dimasukkan kedalam dokumen yang sama. *Server web* bekerja secara langsung terhadap *file* yang bersangkutan tidak memanggil *script* terpisah seperti pada metode CGI. Seluruh kode di eksekusi di *server* (oleh karena itu disebut *server-side script*).



Gambar 2.7 Prinsip kerja PHP dan *Apache*

PHP membuat proses pengembangan aplikasi menjadi mudah karena kelebihan-kelebihannya, yaitu :

1) Script (kode program) terintegrasi dengan file HTML, sehingga *developer* biasa berkonsentrasi langsung pada penampilan *web*-nya.

2) Tidak ada proses *compiling* dan *linking.*

3) Berorientasi objek *(object Oriented).*

4) Sintaksis pemrogramannya udah dipelajari ,sangat menyerupai *C* dan *Perl.*

5) Integrasi yang sangat luas ke berbagai *server database*. Menulis *web* yang terhubung ke *database* menjadi sangat sederhana. *Database* yang didukung oleh PHP : *Oralce, Sybase, mSQL, MySQL, Solid, ODBC, PostgreSQL, Adabas D, FilePRo, Velocis, Informic, dBase, UNIX dbm.*

2.2.9 *Java Script*

JavaScript adalah bahasa *script* yang digunakan pada halaman HTML. J*avaScript* adalah varian dari java yang sangat terkenal dalam lingkungan pemrograman *web*. J*avaScript* mempunyai keunggulan pada sisi client, maupun diakses lebih cepat dari *script server*. Tetapi kelemahannya pada *script* ini belum tentu mampu diproses oleh *browser client* tergantung kompetibilitas browser tersebut. Cara untuk menjalankan *Javascript* hanya dengan *javascript-enabled* yaitu browser yang mampu menjalankan *javascript* seperti Netscape Navigator (versi 2.0 keatas) atau internet ekplorer (MSIE-versi 3.0 ke atas).

Beberapa hal tentang *JavaScript* :

1) *JavaScript* didesai untuk menambah interaktif suatu web
2) *JavaScript* merupakan sebuah bahasa *scripting* dengan bahasa pemrograman yang ringan.
3) *JavaScript* biasanya disisipkan (*embedded*) dalam halaman HTML.
4) *JavaScript* adalah bahasa *interpreter* (yang berarti skrip dieksekusi tanpa proses kompilasi

Ada beberapa cara untuk menuliskan *JavaScript* yaitu :

a) Skrip pada bagian *head*

Skrip ini dieksekusi ketika dipanggil (biasanya berbentuk *function*) atau dipanggil berdasarkan *event* tertentu. Peletakan skrip di *head* akan menjamin skrip di-*load* terlebih dahulu sebelum digunakan.

```
<html>
<head>
<script type=”text/javascript”>
Function message ()
{
Alert (“Alert ini dipanggil saat even dijalankan”);
}
<script>
</head>
</html>
```

b) Skrip pada *Body*

Skrip akan dieksekusi ketika halaman di-*load* sampai dibagian <body> dan ketika ditempatkan berarti antara isi dan *javacript* dijadikan satu bagian.

```
<html>
<head>
</head>
<body>
<script type=”text/javascript”>
Document.write(“Pesan ini akan tampil”);
</script>
</body>
</html>
```

c) External *JavaScript*

Untuk memudahkan menulis script yang berulang-ulang di setiap halaman maka *JavaScript* dapat ditulis di *file* secara *ekternal*. Jadi, antara HTML dan *JavaScript* dipisah, kemudian *file* tersebut dipanggil dari dokumen HTML. *File JavaScript* tersebut disimpan dengan ektensi .js.

```
<html>
<head>
</head>
<body>
<script src=”xxx.js”>
</script>
</body>
</html>
```

2.2.10 Database MySQL

MySQL adalah sebuah sistem manajemen *database* relasi (*relational database management system)* yang bersifat “terbuka” *(open source).* Terbuka maksudnya adalah MySQL boleh di *download* oleh siapa saja. Baik versi kode program aslinya (*source code program*) maupun versi binernya (*executable program*) dan biasa

dugunakan secara (relatif) gratis baik untuk dimodifikasi sesuai dengan kebutuhan seseorang maupun sebagai suatu program aplikasi komputer.

MySQL menggunakan bahasa standar SQL (*Structure Query Language*) sebagai bahasa interaktif dalam mengolah data. Perintah SQL juga sering disebut *Query*. Karena menggunakan bahasa standar yang sama maka tidak akan menjadi kendala besar bila suatu saat nanti berhubungan dengan *database* selain MySQL. Tentu saja tetap ada perbedaan sintaks bahasa paket program SQL tersebut. Adapun beberapa kelebihan MySQL adalah sebagai berikut :

1) **Kecepatan**. Dengan adanya fitur tambahan *Query Chacing* pada MySQL rilis 4.0, kinerja query secara umum akan naik rata-rata 200% dari kinerja biasanya.
2) **Open source.** Dengan konsep ini siapapun dapat berpartisipasi mengembangkan MySQL dan hasil pengembangan itu diserahkan kepada umum atau kepada *Komunitas Open Source*.
3) **Kapabilitas**. MySQL telah digunakan untuk mengelola *database* dengan jumlah 50 juta *record*. Bahkan sanggup untuk mengelola 60.000 tabel dengan jumlah baris 5.000.000.000. Mendukung penggunaan indeks hingga 32 indeks per tabelnya, sedangkan setiap indexnya terdiri dari 1 hingga 16 kolom kriteria. Informasi ini dapat di peroleh dari manual MySQL yang didapat di *download* dari situs MySQL.
4) **Konektifitas dan keamanan**. MySQL mendukung dan menerapkan sistem keamanan dan ijin akses tingkat lanjut, termasuk dukungan keamanan dengan

pengacakan lapisan data (*SSL transport layer encryption*). Bahkan pada MySQL rilis 4.0 dapat diterapkan pembatasan penggunaan sumberdaya *server* berdasarkan penggunaan per *user* . Adanya tingkatan user dan jenis akses yang beragam dapat memberikan batasan akses berdasarkan lokasi pengakses, dan terdapatnya jaminan keamanan pada MySQL.

5) ***Fleksibilitas dan Portabilitas***. MySQL mendukung perintah-perintah ANSI SQL 99 dan beberapa perintah database alternative lainnya sehingga mempermudah untuk beralih dari dan ke MySQL.
6) **Lintas platform sistem operasi**. MySQL dapat dijalankan pada beberapa sistem operasi yang berbeda, seperti *Linux, Microsoft Windows, FreeBSD, Sun Solaris, Mac OS, HP-UX, AIX, QNX, Novell NetWare, SCO OpenUnix, SGI Iric, dan Dec OSF*.

Beberapa perintah dasar sql yang sering dipergunakan pada MySQL adalah sebagai berikut :

a) Create Database

Yaitu perintah yang digunakan untuk membuat database baru.

Sintaks : Create database database_nama database

b) Drop Database

Yaitu perintah yang digunakan untuk menghapus database.

Sintaks : Drop Tabel Tabel_name

c) *Create Tabel*

Yaitu perintah yang digunakan untuk membuat tabel baru.

Sintaks Create Tabel tabel_name (create_definition)

d) *Describe*

Yaitu perintah yang digunakan untuk mendeskripsikan tabel atau logam

Sintaks Describe (Desc) tabel [colum]

e) *Alter Tabel*

Yaitu perintah yang digunakan untuk memodifikasi tabel

Sintaks Alter [Ignor] Tabel table_name

f) *Drop Tabel*

Yaitu perintah yang digunakan untuk menghapus tabel

Sintaks Drop Tabel tabel_name [tabel_name..]

g) *Delete*

Yaitu perintah yang digunakan untuk menghapus record dri tabel

Sintaks Delete From tabel_name Where Where_definiition

h) *Select*

Yaitu perintah yang digunakan untuk query ke database

Sintaks select*from tabel_name

Select Field from tabel_name.

2.2.11 *Cascading Style Sheet (CSS)*

Cascading Style Sheet Disingkat dengan CSS. CSS adalah kumpulan perintah yang dibentuk dari berbagai sumber yang disusun menurut urutan sehingga mampu mengantisipasi konflik gaya yang ada.

Cascading Style Sheet, tahap 1 (CSS1) yang dipopulerkan oleh *World Wide Web Consortium* (W3C) telah menetapkan versi terbaru peselancar Web Netscape dan Microsoft untuk menentukan jenis gaya yang mungkin atau kenyataan yang menterjemahkan ciri-ciri elemen gaya dalam sebuah halaman *web*. CSS disediakan untuk memberikan kebebasan bagi para desainer halaman *web* sesuai dengan yang dikehendakinya.

Definisi elemen gaya halaman *web* yang disepakati adalah seperti berikut:

1. Elemen STYLE yang menentukan object gaya menggunakan senarai deklarasi gaya dan kemudian diletakkan di antara kenyataan <TITLE> dan <BODY>.
2. Elemen LINK yang menyambung kepada dokumen lain juga mempunyai elemen STYLE.
3. Sebuah objek gaya ini dimungkinkan ditempatkan pada dokumen yang berbeda, sehingga diakses oleh beberapa halaman yang memerlukan objek tersebut.

4. Ciri-ciri STYLE adalah elemen tag yang dibentuk khusus oleh para desainer halaman *web*.
5. Ciri gaya yang ditentukan oleh pengguna untuk sebuah halaman *web*.
6. Bentuk objek gaya piawai ditentukan oleh halaman *web* itu sendiri.

2.2.12 Aplikasi Pendukung

2.2.12.1 AppServ

AppServ merupakan tool yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dengan menginstal AppServ maka tidak perlu lagi melakukan instalasi dan konfigurasi *web server,* Apache, PHP dan MySQL secara manual. Appserv akan menginstalasi dan mengkonfigurasikannya secara otomatis atau *auto* konfigurasi.

Pada awalnya AppServ tidak di dukung oleh Pemerintahan Thailand ataupun perusahaan Negara yang tergabung. Berkat inspirasi dari *Phanupong Panyadee* (Yayasan AppServ) yang memberikan konsep bahwa AppServ adalah cara mudah untuk install Apache, PHP, MySQL dalam 1 menit maka distribusi AppServ pertama kali di sediakan pada tanggal 9 Oktober 2001. Awal distribusi, AppServ tersedia dalam bahasa Thailand dan Inggris. Kedepannya AppServ akan menyediakan bahasa-bahasa dunia lainnya. AppServ adalah fitur penuh dari Apache, MySQL, PHP, phpMyAdmin yang Semuanya dapat di *download* dari Official Binary Release. Selain

itu, AppServ mempunyai konfigurasi khusus saat install dalam hal mengkonfigurasi httpd.conf, konfigurasi MySQL my.ini, dan konfigurasi PHP php.ini.

2.2.12.2 Photoshop

Kakak-beradik Thomas dan John Knoll memulai pengembangan Photoshop di tahun 1987. Versi awal Photoshop yang diberi nama "Knoll Software" dirilis sebelum kerjasama dengan Adobe resmi dibuat.

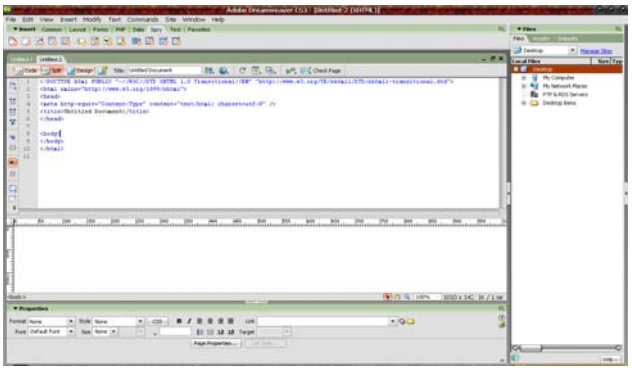
Meskipun pada awalnya Photoshop dirancang untuk menyunting gambar untuk cetakan berbasis-kertas, Photoshop yang ada saat ini juga dapat digunakan untuk memproduksi gambar untuk *World Wide Web*. Photoshop juga memiliki hubungan erat dengan beberapa perangkat lunak penyunting media, animasi, dan "*authoring*" buatan-Adobe lainnya.

Secara Photoshop adalah sebuah program penyunting gambar standar industri yang ditujukan untuk para profesional raster grafik, harga yang ditawarkan pun cukup tinggi; kira-kira US$600. Keadaan ini memancing beberapa programer untuk merancang peralatan grafik ("*graphics tools*") dengan harga yang lebih terjangkau.

Photoshop memiliki kemampuan untuk membaca dan menulis gambar berformat raster dan vektor seperti .png, .gif, .jpeg, dan lain-lain. Photoshop juga memiliki beberapa format *file* khas:*"'.PSD"' ("Photoshop Document") format yang menyimpan gambar dalam bentuk "*layer*", termasuk teks, "*mask*", "*opacity*", *"blend mode"*, *channel* warna, *channel alpha*, *"clipping paths"*, dan *setting duotone*.

2.2.12.3 Adobe Dreamweaver

Dreamweaver merupakan salah satu *software* editor HTML profesional untuk mendesain secara visual dan mengelola *website* dan halaman *web*. Konsep WYSYWIG (*What You See IS What You Get*) HTML (*HyperText Mark-up Language*) akan memberikan anda membuat membuat halaman *web* dan *websites* secara mudah dan cepat. Tetapi dibalik semuanya itu, dreamweaver juga memberikan anda kemudahan mempelajari DHTML (*Dynamic HTML*), yang bisa membuat *website* lebih menarik dan lebih interaktif. Tidak hanya berhenti disini, dreamweaver juga mempunyai fasilitas FTP (*File Transfer Protocol*) untuk *upload* situs anda ke *internet* ketika proses desain sudah selesai dibuat. Fasilitas Dreamweaver termasuk juga *coding tools* dan features: HTML, CSS, dan *JavaScript reference, a JavaScript Debugger*, dan *code editors* (*the Code view and Code inspector*) yang bisa untuk mengedit JavaScript, XML, dan dokumen *text* lainnya secara langsung dalam Dreamweaver. Teknologi baru tentang HTML mengimport dokumen HTML tanpa memformat ulang kode-nya sehingga bisa dengan mudah mengeset Dreamweaver untuk membersihkan dan memformat ulang HTML kapan pun yang inginkan.



Gambar 2.8 Tampilan Dreamweaver