BAB II

LANDASAN TEORI

2.1 Sistem *Database*

Secara umum *database* dapat didefinisikan sebagai kumpulan data yang saling berhubungan satu dengan yang lainnya secara sistematik. *Database* bermula dari ilmu komputer, akan tetapi seiring berkembangnya ilmu pengetahuan, makna *database* kemudian meluas. Dengan adanya *database* banyak sekali hal yang dapat diperoleh, antara lain ketepatan, kecepatan, dan kemudahan dalam pengambilan informasi, selain itu juga dapat menghemat tempat penyimpanan.

Sistem *database* merupakan sistem yang bertugas memanajemen *record* menggunakan komputer dan untuk menyimpan maupun mengambil kembali informasi yang diperlukan oleh pemakai. Selain itu sistem *database* juga bisa diartikan sebagai gabungan antara dua unsur, yaitu *databse* dan sistem manajemen *database*. Berikut adalah komponen dalam sistem *database* (Arifds, 2010):

1. Perangkat keras.
2. Sistem operasi.
3. *Database*.
4. Sistem manajemen *database*.
5. *User*.
6. Perangkat lunak

2.2 Tipe *Database* Relasional

Database Relasional bekerja dengan menghubungkan data pada file-file yang berbeda dengan menggunakan sebuah kunci atau elemen data yang umum. Cara kerja *database* relasional dideskripsikan sebagai elemen-elemen data disimpan dalam tabel lain yang membentuk baris dan kolom. Dalam model *database* ini data diatur secara logis, yakni berdasarkan isi. Masing-masing *record* dalam tabel diidentifikasi oleh sebuah *field* kunci primer yang berisi sebuah nilai unik. Karena itulah data dalam *database* relasional dapat muncul dengan cara yang berbeda dari cara *database* disimpan secara fisik pada komputer. Pengguna tidak boleh mengetahui lokasi fisik sebuah *record* untuk mendapatkan kembali datanya.

2.3 *Entity Relationship Diagram* (ERD)

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam *database* berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD 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 dimodifikasi oleh Chen dan banyak pakar lainnya. Pada saat itu ERD dibuat sebagai bagian dari perangkat lunak yang juga merupakan modifikasi khusus, karena tidak ada bentuk tunggal dan standar dari ERD.

Kegunaan ERD adalah untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol. Pada dasarnya ada tiga simbol yang digunakan, yaitu :

a. Entitas

   Entitas merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain (Fathansyah, 1999: 30). Simbol dari entitas ini biasanya digambarkan dengan persegi panjang. Keberadaan entitas biasanya berdiri sendiri dan digambarkan (direpresentasikan) dengan sekumpulan atribut.

b. Atribut

   Setiap entitas mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut memiliki sesuatu yang dapat mengidentifikasikan isi elemen satu dengan yang lain. Gambar atribut diwakili oleh simbol elips. Ada 2 jenis atribut yaitu:

   1. *Stored Attribute*: atribut yang langsung terlihat pada entitas (atribut nama, atribut alamat).
   2. *Derived Attribute*: merupakan atribut hasil perhitungan dari atribut yang lain (misal atribut umur dihitung dari atribut tanggal lahir).

c. Hubungan Relasi

   Relasi adalah hubungan antara suatu himpunan dengan himpunan entitas yang lainnya. Pada penggambaran diagram hubungan entitas, relasi adalah perekat yang menghubungkan suatu entitas dengan entitas lainnya. Relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dalam satu *database* yaitu (Abdul Kadir, 2002: 48):

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.

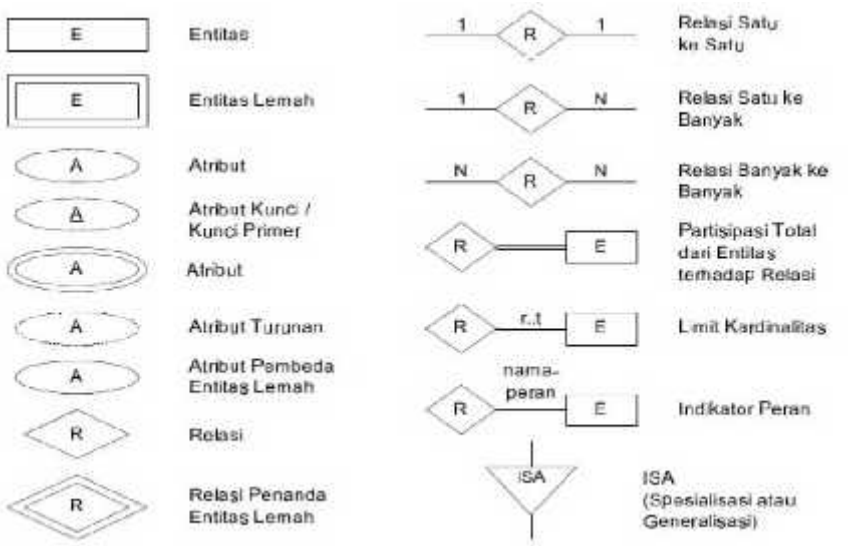
2. Satu ke banyak (*One to many*)

   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 banyak (*Many to many*)

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

Simbol Pada ERD



Gambar 2.1 Simbol Pada ERD

(Sumber: http://id.scribd.com/doc/53178289/Laporan-Database)

2.4 *Data Flow Diagram* (DFD)

DFD adalah alat pembuat model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. DFD ini sering disebut juga dengan *bubble chart, bubble diagram,* model proses, diagram alur kerja atau model fungsi.

DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada

data yang dimanipulasi oleh sistem. Dengan kata lain DFD adalah alat pembuat model yang memberikan penekanan hanya pada fungsi sistem.

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

Syarat-syarat pembuatan DFD ini adalah:

1. Pemberian nama untuk tiap komponen DFD
2. Pemberian nomor untuk komponen proses
3. Penggambaran DFD sesering mungkin agar enak dilihat
4. Penghindaran penggambaran DFD yang rumit
5. Pemastian DFD yang dibentuk tiu konsisten secara logika

2.5 Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang *database* relasional. Pada dasarnya normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel relasional.

Adapun tujuan dari normalisasi adalah yaitu:

1. Untuk menhilangkan kerangkapan data
2. Untuk mengurangi kompleksitas
3. Untuk mempermudah pemodifikasian data

Proses normalisasi, merupakan proses pengelompokan data elemen menjadi tabel-tabel yang menunjukkan entitas dan relasinya. Pada proses normalisasi selalu diuji beberapa kondisi, apakah ada kesulitan saat menambah, menghapus, mengubah dan membaca pada suatu *database*.

Berikut ini adalah proses dari normalisasi:

1. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat.
2. Apabila tabel yang diuji belum memenuhi syarat tertentu, maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal.

Sebuah tabel dikatakan baik (efisien) atau normal jika memenuhi 3 kriteria sebagai berikut:

1. Jika ada dekomposi (penguraian) tabel, maka dekomposinya harus dijamin aman (*Lossless-Join Decomposition*). Artinya, setelah tabel tersebut diuraikan menjadi tabel-tabel baru, tabel-tabel baru tersebut bisa menghasilkan tabel semula dengan sama persis.
2. Terpeliharanya ketergantungan fungsional pada saat perubahaan data (*Depedency Preservation*).
3. Tidak melanggar BCNF (*Boyce-Code Normal Form*).

Jika kriteria BCNF tidak dapat terpenuhi, maka paling tidak tabel tersebut tidak melanggar bentuk normal tahap ketiga (3rd Normal Form / 3NF).

Bentuk-bentuk Normal:

1. Bentuk Normal Tahap Pertama (*1st Normal Form* atau 1NF)
   a. Bentuk normal 1NF terpenuhi jika sebuah tabel tidak memiliki atribut bernilai banyak (*multivalued attribute*), atribut *composite* atau kombinasinya dalam domain data yang sama.
   b. Setiap atribut dalam tabel tersebut harus bernilai *atomic* (tidak dapat dibagi-bagi lagi).
2. Bentuk Normal Tahap Kedua (*2nd Normal Form* atau 2NF)
   a. Bentuk normal 2NF terpenuhi dalam sebuah tabel jika telah memenuhi bentuk 1NF, dan semua atribut selain *primary key*, secara utuh memiliki *Functional Dependency* pada *primary key*.
   b. Sebuah tabel tidak memenuhi 2NF, jika ada atribut yang ketergantungannya (*Functional Dependency*) hanya bersifat parsial saja (hanya tergantung pada sebagian dari *primary key*).
   c. Jika terdapat atribut yang tidak memiliki ketergantungan terhadap *primary key*, maka atribut tersebut harus dipindah atau dihilangkan.
3. Bentuk Normal Tahap (3rd Normal Form atau 3NF)

Bentuk normal 3NF terpenuhi jika telah memenuhi bentuk 2NF, dan jika tidak ada atribut *non primary key* yang memiliki ketergantungan terhadap atribut *non primary key* yang lainnya.

2.6 *Personal Home Page Hypertext Preprocessor* (PHP)

PHP dikatakan sebagai sebuah *serverside embedded script language* artinya sintaks-sintaks dan perintah yang kita berikan akan sepenuhnya dijalankan oleh *server* tetapi disertakan pada halaman HTML biasa. Aplikasi-aplikasi yang dibangun oleh PHP pada umumnya akan memberikan hasil pada web *browser*, tetapi prosesnya secara keseluruhan dijalankan di *server*.

Pada prinsipnya *server* akan bekerja apabila ada permintaan dari *client*. Dalam hal ini *client* menggunakan kode-kode PHP untuk mengirimkan permintaan ke *server*. Ketika menggunakan PHP sebagai *serverside embedded script language* maka *server* akan melakukan hal-hal sebagai berikut:

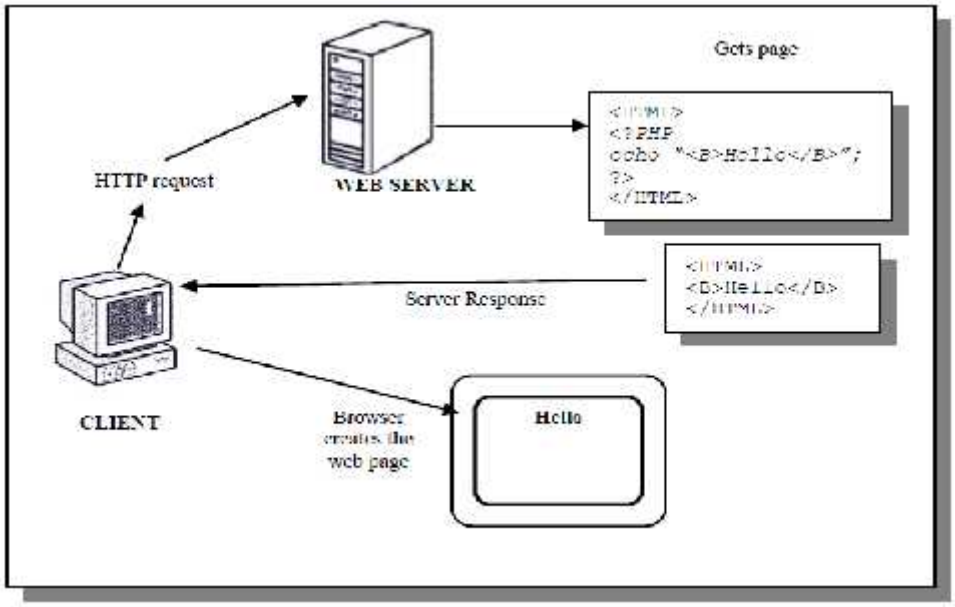
a. Membaca permintaan dari *client* atau *browser*

b. Mencari halaman di *server*

c. Melakukan instruksi yang diberikan oleh PHP untuk melakukan modifikasi pada halaman

d. Mengirim kembali halaman tersebut kepada *client* melalui internet atau intranet

Berikut adalah gambar prinsip kerja dari PHP



Gambar 2.2 Prinsip Kerja PHP

(Sumber: http://pusdatin.deptan.go.id/admin/RB/Programming/Materi%20PHP.pdf)

Berikut adalah beberapa dari kelebihan PHP

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaanya.
2. Web *Server* yang mendukung PHP dapat ditemukan dimana - mana dari mulai apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan developer yang siap membantu dalam pengembangan.
4. Dalam sisi pemahamanan, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
5. PHP adalah bahasa *open source* yang dapat digunakan di berbagai sistem operasi seperi Linux, Unix, Macintosh, Windows dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

Kode PHP disimpan sebagai *plain text* dalam format ASCII (*American Standard Code for Information Interchange*), sehingga kode PHP dapat ditulis hampir disemua *text editor* seperti *windows notepad*, *windows wordpad*, dll. Kode PHP adalah kode yang disertakan di sebuah halaman HTML dan kode tersebut dijalankan oleh *server* sebelum dikirim ke *browser*.

Berikut adalah contoh cara penulisan PHP:

```
<html>
   <head>
      <title>Penulisan PHP</title>
   </head>

   <body>

      <?php
            echo ”Contoh text yang menggunakan kode PHP”;
      ?>

   </body>
</html>
```

Pada *file* dengan extensi .html, HTTP *server* hanya melewatkan *content* dari *file* menuju ke *browser*. *Server* tidak mencoba untuk mengerti atau memproses *file*, karena itu adalah tugas sebuah *browser*.

Pada *file* dengan ekstensi .php akan ditangani secara berbeda. Yang memiliki kode PHP akan diperiksa. Web *server* akan memulai bekerja apabila berada diluar lingkungan

kode HTML. Oleh karena itu *server* akan melewati semua *content* yang berisi kode HTML, CSS, JavaScript, *simple text* di *browser* tanpa diinterpretasikan di *server*.

Blok *scripting* PHP selalu diawali dengan <?php dan diakhiri dengan ?>. Blok *scripting* PHP dapat ditempatkan dimana saja di dalam dokumen. Pada beberapa *server* yang mendukung, blok *scripting* PHP dapat diawali dengan <? dan diakhiri dengan ?>. Namun, untuk kompatibilitas maksimum, sebaiknya menggunakan bentuk yang standar (<?php ?>).

Setiap baris kode PHP harus diakhiri dengan semikolon (;). Semikolon ini merupakan separator yang digunakan untuk membedakan satu instruksi dengan instruksi lainnya.

PHP menggunakan // untuk membuat komentar baris tunggal atau /* dan */ untuk membuat suatu blok komentar.

2.7 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen *database* SQL atau DBMS yang *multithread*, *Multiuser*, 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 (Achmad Solichin, 2012).

MySQL merupakan salah satu jenis *database server* yang sangat terkenal. MySQL juga termasuk dalam jenis RDBMS, itulah sebabnya istilah seperti tabel, baris, dan kolom digunakan dalam MySQL.

Tidak seperti Apache yang merupakan perangkat lunak yang 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.

MySQL memiliki beberapa keistimewaan, antara lain :

1. Portabilitas

   MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac OS X *Server*, Solaris, Amiga, dan masih banyak lagi.

2. Perangkat lunak yang *open source*

   MySQL didistribusikan sebagai perangkat lunak *open source*, dibawah lisensi GPL sehingga dapat digunakan secara gratis.

3. *Multiuser*

   MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. *Performance tuning*

   MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. Ragam tipe data

   MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed* atau *unsigned integer*, *float*, *double*, *char*, *text*, *date*, *timestamp*, dan lain-lain.

6. Perintah dan fungsi

   MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).

7. Keamanan

   MySQL memiliki beberapa lapisan keamanan seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.

8. Skalabilitas dan pembatasan

   MySQL mampu menangani *database* dalam skala besar, dengan jumlah rekaman lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. Konektivitas

   MySQL dapat melakukan koneksi dengan *client* menggunakan protokol TCP/IP, Unix soket (UNIX), atau *Named Pipes* (NT).

10. Lokalisasi

    MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.

11. Antar muka

    MySQL memiliki antar muka terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi *Application Programming Interface* (API).

12. Klien dan peralatan

    MySQL dilengkapi dengan berbagai peralatan yang dapat digunakan untuk administrasi *database*, dan pada setiap peralatan yang ada disertakan petunjuk *online*.

13. Struktur tabel

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan *database* lainnya semacam PostgreSQL ataupun Oracle.

2.8 *Asynchronous JavaScript and XML* (AJAX)

AJAX, terdiri dari HTML, Javascript, DHTML dan DOM yang kemudian digabungkan dengan bahasa pemograman web di sisi server seperti PHP dan ASP, sehingga membentuk suatu aplikasi berbasis web yang interaktif.

AJAX bukanlah bahasa pemograman baru, tetapi adalah teknik baru untuk membuat aplikasi web lebih baik, lebih cepat dan lebih interaktif. Dengan AJAX, Javascript dapat langsung berkomunikasi dengan server dengan menggunakan objek XMLHttpRequest. Dengan objek ini, javascript dapat melakukan transaksi data dengan server web, tanpa harus me-*reloading* halaman web tersebut secara keseluruhan.

Berikut adalah teknologi yang termasuk dalam aplikasi AJAX:

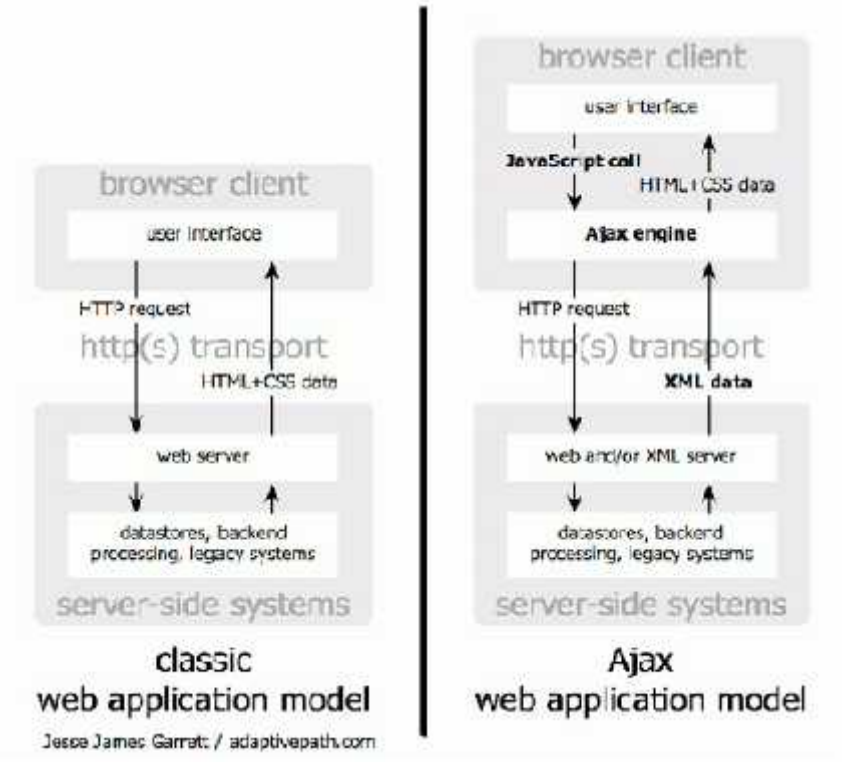
1. HTML yang digunakan untuk membuat Web forms dan mengindentifikasikan filed-field yang akan digunakan dalam aplikasi.
2. JavaScript adalah kode inti untuk menjalankan aplikasi Ajax dan untuk membantu memfasilitasi komunikasi dengan aplikasi.

3. DHTML, atau Dynamic HTML, membantu untuk membuat form atau web secara dinamis dengan menggunakan tag , dan elemen HTML dinamis lainya.
4. DOM, Document Object Model, akan digunakan melalui kode JavaScript untuk bekerja dengan kedua struktur dari HTML dan XML yang dalam beberapa kasus berasal dari server.

Objek yang harus dimengerti adalah XMLHttpRequest yakni objek javascript. Untuk membuatnya dapat dilihat seperti ditunjukkan pada kode di bawah ini .

```
<script language="javascript" type="text/javascript">
      var xmlHttp = new XMLHttpRequest();
</script>
```

Untuk mendapatkan dan mengirim data dari suatu *database* atau file di server menggunakan javascript tradisional, maka dibuatlah sebuah form. Dan user harus mengklik tombol "*Submit*" untuk mengirim atau mendapatkan informasi dan menunggu respon dari server. Kemudian, halaman yang baru berupa hasilnya akan di-*load*. Karena server selalu memberikan halaman baru setiap user tekan tombol submit, aplikasi web sederhana akan berjalan lambat dan kurang interaktif ataupun *user-friendly*.



Gambar 2.3 Perbandigan Web Aplikasi Tradisional Dengan Ajax

Dengan objek XMLHttpRequest, suatu halaman web dapat membuat request dan mendapatkan respon dari server web tanpa me-*reload* halaman secara keseluruhan. User akan selalu tetap dengan halaman yang sama. Bahkan user tidak akan tahu jika data yang dikirim maupun diterima oleh server sedang diproses, karena javascript melakukan transaksi data dibalik layar. Bagusnya lagi permintaan dikirim *asynchronous*, yang berarti bahwa kode JavaScript dan pengguna tidak menunggu pada server untuk merespon. Sehingga pengguna dapat terus mengolah data dan menggunakan aplikasi.

Kode JavaScript bahkan bisa mendapatkan data, melakukan perhitungan, dan mengirim permintaan lain, semua tanpa campur tangan pengguna. Hal ini tidak terlepas dari kelebihan sebuah XMLHttpRequest. Hal ini dapat bicara secara bolak-balik dengan server untuk semua yang diinginkan, tanpa pernah tahu dari pengguna tentang apa yang sebenarnya terjadi.

Objek XMLHttpRequest disupport hampir semua browser seperti Internet Explorer, Firefox, Chrome, Opera, dan Safari. Untuk membuat sebuah objek XMLHttpRequest agar dapat didukung oleh beberapa browser adalah sebagai berikut.

```
if (window.XMLHttpRequest){
      // kode untuk IE7+, Firefox, Chrome, Opera, Safari
            return new XMLHttpRequest();
}
if (window.ActiveXObject){
      // kode untuk IE6, IE5
      return new ActiveXObject("Microsoft.XMLHTTP");
}
```

2.9 *Short Message Service* (SMS)

Visualtron (2012) dalam “Layanan pesan singkat” *Short Message Service* atau dalam bahasa Indonesia Layanan Pesan Singkat adalah sebuah layanan yang dilaksanakan dengan sebuah telepon genggam untuk mengirim atau menerima pesan-pesan pendek. Pada mulanya SMS dirancang sebagai bagian daripada GSM, tetapi sekarang sudah didapatkan pada jaringan bergerak lainnya termasuk jaringan UMTS.

Sebuah pesan SMS maksimal terdiri dari 140 *bytes*, dengan kata lain sebuah pesan bisa memuat 140 karakter 8-*bit*, 160 karakter 7-*bit* atau 70 karakter 16-*bit* untuk bahasa Jepang, bahasa Mandarin dan bahasa Korea yang memakai Hanzi (Aksara Kanji / Hanja). Selain 140 *bytes* ini ada data-data lain yang termasuk. Adapun beberapa metode untuk mengirim pesan yang lebih dari 140 *bytes*, tetapi seorang pengguna harus membayar lebih dari sekali.

SMS bisa juga untuk mengirim gambar, suara dan film. SMS bentuk ini disebut MMS.

Pesan SMS dikirim dari sebuah telepon genggam ke pusat pesan (SMSC dalam bahasa Inggris), di sini pesan disimpan dan mencoba mengirimnya selama beberapa kali. Setelah sebuah waktu yang telah ditentukan, biasanya 1 hari atau 2 hari, lalu pesan dihapus. Seorang pengguna bisa mendapatkan konfirmasi dari pusat pesan ini.

SMS sangat populer di Eropa, Asia dan Australia. Di Amerika Serikat, SMS secara relatif jarang digunakan. SMS populer karena relatif murah. Di Indonesia, tergantung perusahaannya sebuah SMS berkisar antara Rp. 250,- sampai Rp. 350,-.

Karena kesulitan mengetik atau untuk menghemat tempat, biasanya pesan SMS disingkat-singkat. Tetapi kendala kesulitan sekarang sudah teratasi karena banyak telepon genggam yang memiliki fungsi kamus.

2.10 SMS *Gateway*

David Sudana (2009) dalam Mengenal Cara Kerja SMS Gateway, menjelaskan SMS gateway adalah sebuah perangkat yang menawarkan layanan transit SMS, mentransformasikan pesan ke jaringan selular dari media lain, atau sebaliknya, sehingga memungkinkan pengiriman atau penerimaan pesan SMS dengan atau tanpa menggunakan ponsel.

Sebuah sistem SMS *Gateway*, umumnya terdiri komponen *Hardware* (*Server* atau komputer yang dilengkapi dengan perangkat jaringan) dan *Software* (Aplikasi yang digunakan untuk pengolahan pesan). Dan untuk sebuah sistem yang besar umumnya menggunakan *database* untuk penyimpanan data.

2.10.1 Manfaat SMS *Gateway*

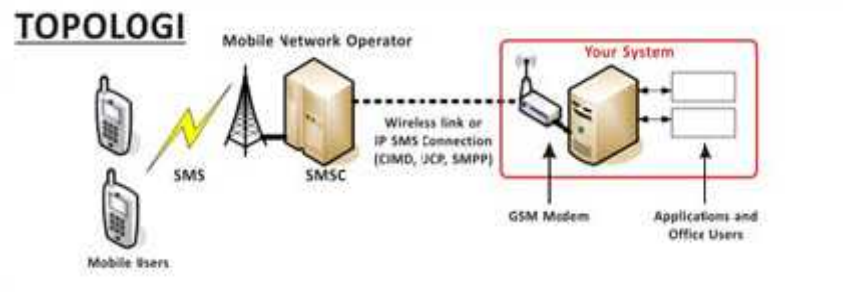
SMS *Gateway* merupakan pintu gerbang bagi penyebaran informasi dengan menggunakan SMS. SMS *gateway* juga dapat menyebarkan pesan ke banyak nomor secara otomatis dan cepat yang langsung terhubung dengan *database* tanpa harus mengetik ratusan nomor dan pesan.

Selain itu SMS Gateway juga dapat mengelola pesan-pesan yang ingin dikirim. Dengan menggunakan program tambahan yang dapat dibuat sendiri, pengirim pesan dapat lebih fleksibel dalam mengirim berita.

2.10.2 Cara Kerja SMS *Gateway*

Daud Edison Tarigan (2012) dalam bukunya yang berjudul Membangun SMS Gateway Berbasis Web Dengan Codeigniter, mengatakan cara kerja SMS *Gateway* pada dasarnya hampir sama dengan mengirimkan sms melalui *handphone* pada umumnya. Letak perbedaan SMS Gateway ini adalah perangkat yang mengirimkan pesan bukanlah *handphone*, akan tetapi menggunakan Modem GSM. Modem ini lah yang dikendalikan oleh PC menggunakan aplikasi SMS *Gateway*.

Adapun blok diagram dari sistem SMS Gateway ini dapat dilihat pada gambar 2.3



Gambar 2.4 Blok diagram SMS Gateway

(Sumber: http://padi.net.id/upload/content/topologi-sms-gateway.jpg)