BAB 2

LANDASAN TEORI

2.1 Jaringan Komputer dan Internet

Menurut Shelly, Woods, Dorin (2008, p2), jaringan adalah dua atau lebih komputer yang terhubung untuk saling berbagi sumber daya dan informasi. Jaman sekarang, jalur data baik dari yang berkecepatan rendah, menengah, sampai tinggi, terhubung ke jaringan. Jalur data ini mengijinkan data untuk berpindah dari satu komputer ke komputer lainnya.

Menurut Shelly, Woods, Dorin (2008, p2), Internet adalah koleksi jaringan komputer di seluruh dunia yang menghubungkan jutaan komputer yang digunakan untuk bisnis, pemerintahan, institusi pendidikan, organisasi, dan individual menggunakan modem, kabel telepon, kabel televisi, satelit, dan peralatan komunikasi lainnya.

Menurut Shelly, Woods, Dorin (2008, p2*), Internet backbone* adalah kumpulan jalur data berkecepatan tinggi yang menghubungkan sistem komputer utama yang berlokasi di seluruh dunia.

Gambar 2.1 Internet

Menurut Shelly, Woods, Dorin (2008, p2), Penyedia jasa internet (*Internet Service Provider / ISP*) adalah sebuah perusahaan yang mempunyai koneksi yang permanen ke *internet backbone***.** ISP memanfaatkan jalur data berkecepatan menengah atau tinggi untuk mengijinkan perorangan atau perusahaan-perusahaan untuk terhubung ke (*backbone*) untuk mengakses internet. (Halaman 2)

Sampai saat ini, sudah lebih dari 950 juta orang di 240 negara terkoneksi melalui internet menggunakan komputer-komputer mereka di rumah-rumah, kantor-kantor, sekolah-sekolah, dan lokasi-lokasi publik seperti perpustakaan.

Pengguna dengan komputer yang terkoneksi ke internet akan dapat mengakses berbagai macam layanan seperti *e-mail, social networking, online shopping,* dan *world wide web.*

Gambar 2.2 Berbagai Layanan yang Disediakan Internet

2.2 World Wide Web

Menurut Shelly, Woods, Dorin (2008, p3), *World Wide Web* adalah bagian dari internet yang mendukung multimedia dan terdiri dari sebuah koleksi dokumen yang terhubung, biasanya dipanggil sebagai *web* saja.

Untuk mendukung *multimedia*, web bergantung pada *Hypertext Transfer Protocol* (HTTP).

HTTP adalah seperangkat aturan untuk saling bertukar teks, grafik, suara, video, dan file multimedia lainnya.

World Wide Web terdiri dari berbagai macam situs *web* yang memiliki fungsi-fungsi yang berbeda tergantung pada yang membuat situs web tersebut.

Situs w*eb* (*website*) adalah koleksi halaman web yang berhubungan, yang dibuat dan dipelihara oleh individual, perusahaan, institusi pendidikan, atau organisasi lainnya.

Halaman *web* adalah dokumen-dokumen yang terhubung atau halaman-halaman yang berisi informasi yang ada di *Web*. Oleh karena *web* dapat mendukung teks, grafik, suara, dan video, halaman *web* dapat memiliki elemen multimedia seperti yang disebutkan di atas.

Web selalu berubah dan terdiri dari ratusan juta halaman *web*, hal ini disebabkan oleh kemudahan dalam membuat halaman *web*, dan masih akan terus bertambah setiap saat.

Gambar 2.3 Contoh sebuah situs web

Setiap situs *web* memiliki *home page,* yang adalah dokumen pertama yang dilihat pengguna ketika mereka mengakses internet.

Halaman-halaman *web* yang ada disimpan di *web server* yaitu sebuah komputer yang menyimpan dan mengirim halaman-halaman yang diminta dan file lainnya. Setiap komputer yang mempunyai software *web server* yang terinstall di dalamnya, dan juga terhubung ke internet, dapat menjadi *web server*, setiap situs *web* yang ada, disimpan, dijalankan dari satu atau lebih *web server*. Sebuah situs *web* yang cukup besar bisa saja tersebar pada beberapa server di lokasi geografik yang berbeda.

Menurut Shelly, Woods, Dorin (2008, p4), *Publishing* adalah proses menyalin halaman-halaman *web* dan file lainya ke *web server*. Ketika sebuah halaman sudah di-*publish*, semua orang yang memiliki akses ke internet dapat melihatnya, dimanapun *web server* dan pengguna itu berada. Ketika halaman *web* di *publish*, halaman *web* tersebut dapat dibaca oleh hampir seluruh komputer, tidak masalah anda menggunakan Mac, Windows, ataupun Linux, anda dapat mengakses jutaan halaman *web* yang telah di *publish*.

Menurut Shelly, Woods, Dorin (2008, p6) , ada tiga jenis situs web, yaitu internet, intranet, dan ekstranet.

1. Internet

   Internet dikenal juga sebagai situs *web*, adalah sebuah situs yang secara umum tersedia untuk umum. Perorangan, grup, perusahaan, dan institusi pendidikan menggunakan situs internet atau situs *web* untuk berbagai macam kebutuhan, seperti untuk menjual produk dan layanan, menyediakan dukungan teknis dan produk untuk pelanggan, menyediakan informasi tentang suatu perusahaan sesuai kebutuhan, dan lain-lain.

2. Intranet

   Intranet adalah jaringan pribadi yang menggunakan teknologi internet untuk berbagi informasi perusahaan diantara para karyawannya. Intranet terletak dalam suatu jaringan organisasi atau perusahaan. Beberapa intranet juga menggunakan sistem *password* untuk memberikan keamanan dari akses yang tidak memiliki hak.

3. Ekstranet

   Ekstranet adalah jaringan pribadi yang menggunakan teknologi internet untuk berbagi informasi bisnis dengan mitra perusahaan tertentu atau dengan pelanggan utama. Kebanyakan ekstranet dilindungi oleh password untuk membatasi akses kepada *supplier*, *vendor*, mitra, ataupun pelanggan. Perusahaan dapat berbagi panduan manual produk, modul pelatihan, status inventori, dan informasi pesanan.

Untuk dapat melihat halaman *web* dalam situs *web*, sebuah komputer memerlukan *web browser*, atau biasa disebut *browser*.*Web browser* adalah sebuah program yang menterjemahkan dan menampilkan halaman *web* dan memungkinkan pergguna untuk melihat dan berinteraksi dengan halaman *web*. Contoh *web browser* yang terkenal sekarang adalah Microsoft Internet Explorer, Mozilla Firefox, dan Apple Safari.

Browser menyediakan berbagai macam fitur, termasuk kemampuan untuk menemukan halaman *web*, untuk maju dan mundur antara halaman web, untuk menandai halaman *web*, dan untuk konfigurasi keamanannya.

Untuk menemukan sebuah halaman web menggunakan *URL*, kita harus mengetikan *URL* (*Uniform Resource Locator*) di bagian alamat atau lokasi pada

browser. *URL* adalah alamat dari sebuah dokumen atau file lain yang dapat diakses melalui internet.

Hyperlink (biasa disebut juga *link*) adalah elemen yang digunakan untuk menghubungkan satu halaman web ke halaman *web* lain pada *web server* yang berlokasi sama ataupun berbeda di berbagai tempat di dunia.*Hyperlink*juga dapat digunakan untuk berpindah ke bagian lain dari halaman web yang sama.

Hyperlink ini merupakan bagian penting dari *world wide web*. Dengan *hyperlink* ini, pengguna situs *web* tidak harus melihat informasi yang terdapat di halaman *web* secara linear, tetapi mereka dapat meng-klik *hyperlink* yang tersedia untuk melihat informasi dalam berbagai macam cara. Banyak elemen-elemen dari halaman web yang dapat dijadikan *hyperlink*, seperti teks, grafik, animasi.

Gambar 2.4 URL dan *Hyperlink*

2.3 HTML

Menurut Shelly, Woods, Dorin (2008, p8), halaman-halaman *web* dibuat menggunakan HTML yaitu sebuah bahasa penulisan yang digunakan untuk membuat dokumen dalam *world wide web*. HTML menggunakan seperangkat instruksi khusus dinamakan *tags* atau *markup* untuk mendefinisikan struktur dan susunan dari sebuah dokumen dan menentukan bagaimana halaman tersebut ditampilkan dalam *browser*.

Sebuah halaman *web* adalah sebuah file yang berisi teks dan HTML *tag*. HTML *tag* digunakan untuk menandakan teks, menentukan bagaimana suatu teks ditampilkan dalam suatu halaman pada *web*. HTML memiliki ratusan *tag* yang dapat digunakan untuk mengatur tampilan pada halaman *web*, juga untuk membuat *hyperlink* ke dokumen lain atau halaman web lain.

Contoh beberapa tag:

a. dan untuk menebalkan teks

b. dan untuk menunjukkan paragraf baru

c. <script> dan </script> untuk mendefinisikan *script* pada sisi klien

d. <title> dan </title> untuk memberikan judul pada dokumen

e. <form> dan </form> untuk membuat form untuk input pengguna

f. Dan lain-lain

HTML bersifat *platform independent***,** dalam arti kita dapat membuat atau meng-kode sebuah file HTML pada satu komputer dan kemudian menggunakan *browser* pada komputer jenis lain untuk menampilkan file tersebut sebagai halaman *web*. Halaman web akan terlihat sama tanpa melihat tipe *platform* apa yang kita gunakan.

Elemen-elemen HTML:

a. Judul

Ditandai dengan tag <title> …</title>

Digunakan untuk memberikan judul pada bar judul pada halaman web.

b. Tubuh

Ditandai dengan tag <body> …</body>

Digunakan untuk menspesifikasikan apa saja yang akan ditampilkan dalam halaman *web*. Seluruh isi dari halaman *web* dimasukkan ke dalam tag awal <body> dan tag akhir </body>.

c. Paragraf

Ditandai dengan tag …

Digunakan untuk memberikan sedikit baris kosong sebelum teks dalam paragraf.

d. Line Break

Ditandai dengan tag

Digunakan untuk memberikan baris kosong / mengganti baris (enter)

HTML dapat digunakan berbarengan dengan teknologi *web* lain untuk menyediakan fungsi-fungsi tambahan pada halaman *web*. Sebagai contoh, DHTML (Dynamic HTML) mendeskripsikan sebuah kombinasi dari *tag-tag* HTML, CSS (*Cascading Style Sheet*, dan bahasa *scripting* seperti JavaScript). DHTML memungkinkan untuk membuat halaman web yang interaktif dan beranimasi.

Cascading Style Sheet (CSS) adalah sekumpulan kode yang memungkinkan untuk mengatur elemen dalam sebuah halaman *web* ataupun dalam keseluruhan situs *web*.

Extensible Markup Language (XML) adalah bahasa *mark-up* yang menggunakan *tag* untuk mendeskripsikan struktur dan isi dari sebuah dokumen, bukan formatnya.

Extensible Hypertext Markup Language (XHTML) adalah HTML yang sesuai dengan aturan-aturan XML, XHTML menkombinasikan fitur-fitur tampilan dari HTML dan standar pengkodean yang lebih ketat yang diperlukan XML. oleh karena itu, XHTML menjamin halaman *web* yang menggunakan XHTML akan dapat dibaca oleh berbagai tipe aplikasi.

Menurut Shelly, Woods, Dorin (2008, p11), beberapa aturan pengkodean yang harus dipenuhi untuk memastikan halaman web mengikuti standar XHTML:

a. File HTML harus mengikut-sertakan pernyataan DOCTYPE.

   Contoh: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

   <html xmlns="http://www.w3.org/1999/xhtml">

   <html></html>

b. Semua *tag* dan atribut harus dalam huruf kecil.

   Contoh:<table width=’100%’>

c. Semua nilai atribut harus ditutup dengan tanda petik satu atau dua.

   Contoh: <table width=’100%’>

d. Semua *tag* harus ditutup termasuk img, hr, dan br walaupun tidak memiliki *tag* penutup.

   Contoh:

   <hr />

   ini adalah paragraf

e. Semua elemen yang bertingkat harus ditutup dengan benar.

Contoh: paragraf yang di-bold

Menurut Shelly, Woods, Dorin (2008, p15), struktur dari situs *web*dapat dibagi menjadi 3 tipe, yaitu :

a. Linear

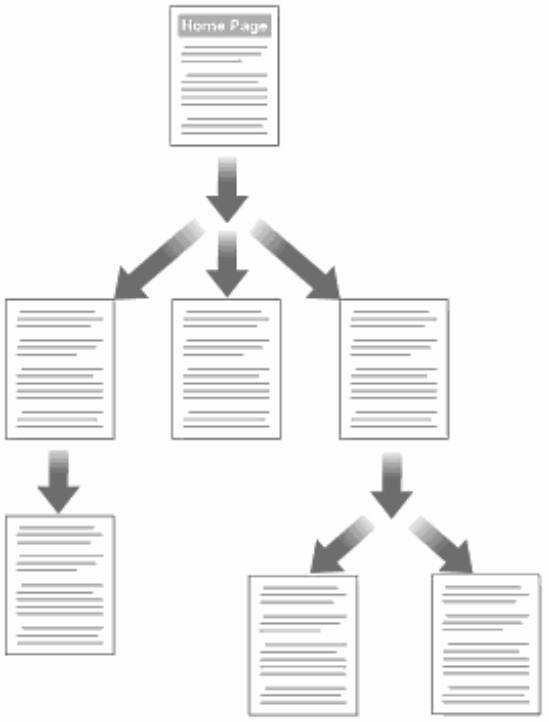
Struktur situs *web* linear menghubungkan halaman-halaman *web* dalam suatu garis lurus, seperti terlihat pada gambar di bawah ini.

Gambar 2.5 Struktur Linear

Struktur situs *web* linear tepat digunakan jika informasi pada halaman *web* harus dibaca dalam urutan tertentu. Contohnya jika informasi dari halaman *web* pertama diperlukan untuk memahami informasi pada halaman *web* kedua, maka struktur linear harus digunakan. Setiap halaman mempunyai link dari satu halaman *web* ke halaman selanjutnya, juga *link* ke halaman awal.

b. Hierarki

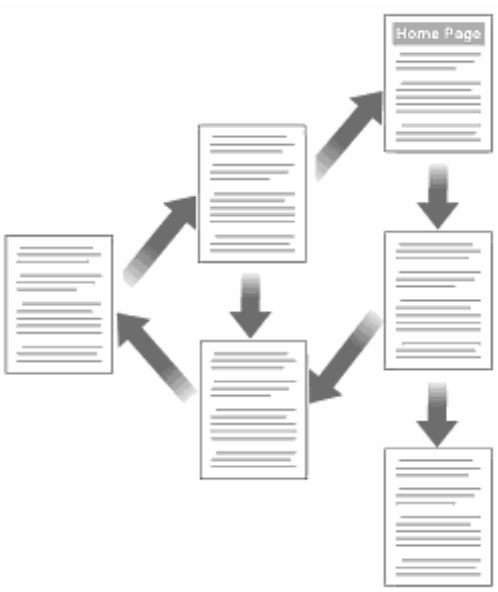
Struktur situs *web* hierarki menghubungkan halaman *web* seperti struktur pohon. Struktur hierarki efektif digunakan pada situs yang memiliki halaman index utama atau halaman daftar isi yang mempunyai *link* ke semua halaman *web* lainnya. Dengan struktur ini, halaman index utama berisi informasi yang umum, dan halaman lainnya berisi informasi yang detil.



Gambar 2.6 Struktur Hierarki

c. *Webbed*

Struktur ini tidak memiliki organisasi. Struktur ini bagus digunakan pada situs *web* dengan informasi yang tidak perlu dibaca dalam urutan tertentu, dan memiliki banyak pilihan navigasi yang dapat dipilih oleh pengguna. *World wide web* menggunakan struktur *webbed*, sehingga pengguna dapat bergerak secara bebas diantara halaman *web* dalam urutan yang dipilih oleh pengguna sendiri.



Gambar 2.7 Struktur *Webbed*

Kebanyakan situs *web* adalah kombinasi dari struktur linear, hierarki dan *webbed*. Menggunakan kombinasi dari ketiga struktur di atas dikatakan tepat apabila dapat membantu penggunamenavigasi suatusitus dengan mudah.

2.4 JavaScript

Menurut Abdul Kadir (2003,p268), asal mula JavaScript adalah LiveScript yang dikembangkan pertama kali tahun 1995 oleh Netscape Communications. Pada akhir tahun 1995, Netscape Communications dan Sun Microsystems berkolaborasi dan mengubah nama dari LiveScript menjadi JavaScript. *Browser* Internet Explorer mulai mendukung JavaScript mulai versi 3.0 ke atas dan Netscape Navigator mulai versi 2.0 ke atas.

JavaScript adalah bahasa skrip yang ditempelkan pada kode HTML dan diproses pada sisi klien. Dengan adanya JavaScript maka dimungkinkan sebuah halaman web yang interaktif dan beranimasi. JavaScript bersifat *case sensitive* yang artinya program

mengenali huruf besar dan kecil pada setiap nama variabel dan fungsinya. Contoh sebuah variabel Var dan var akan memiliki nilai dan alamatnya masing-masing dan dianggap sebagai 2 variabel yang berbeda satu sama lain.

JavaScript dan Java adalah dua buah bahasa yang berbeda satu sama lain dimana perbedaan paling mendasarnya adalah kode JavaScript diinterpretasikan oleh komputer klien (kode asli program dapat dilihat langsung di sisi klien, bukan dalam bahasa mesin hasil kompilasinya) sedangkan Kode Java dikompilasi oleh pemrogram dan hasil kompilasinyalah yang dijalankan oleh komputer klien.

Semua kode JavaScript harus dimasukan dalam sebuah *tag* khusus yang diawali dengan *tag*<script> dan diakhiri dengan *tag* </script>.

Contoh cara memakai JavaScript:

```
<script language = “JavaScript”>
  document.write(“Hello to JavaScript!”);
</script>
```

2.5 Ajax

Menurut Garret (2005), Ajax merupakan singkatan dari *Asynchronous JavaScript and XML* adalah sebuah pendekatan baru yang ditujukan untuk mengembangkan *web* interaktif. Penggunaan AJAX bertujuan untuk menjadikan halaman-halaman *web* lebih responsif.

AJAX bukanlah sebuah teknologi, melainkan suatu kumpulan teknologi-teknologi yang berjalan di belakangnya. Teknologi-teknologi yang berada di balik AJAX, antara lain :

a. Presentasi Standard menggunakan HTML / XHTML (*Hypertext/ Extensible Markup Language*) dan CSS (*Cascading Stylesheet*).

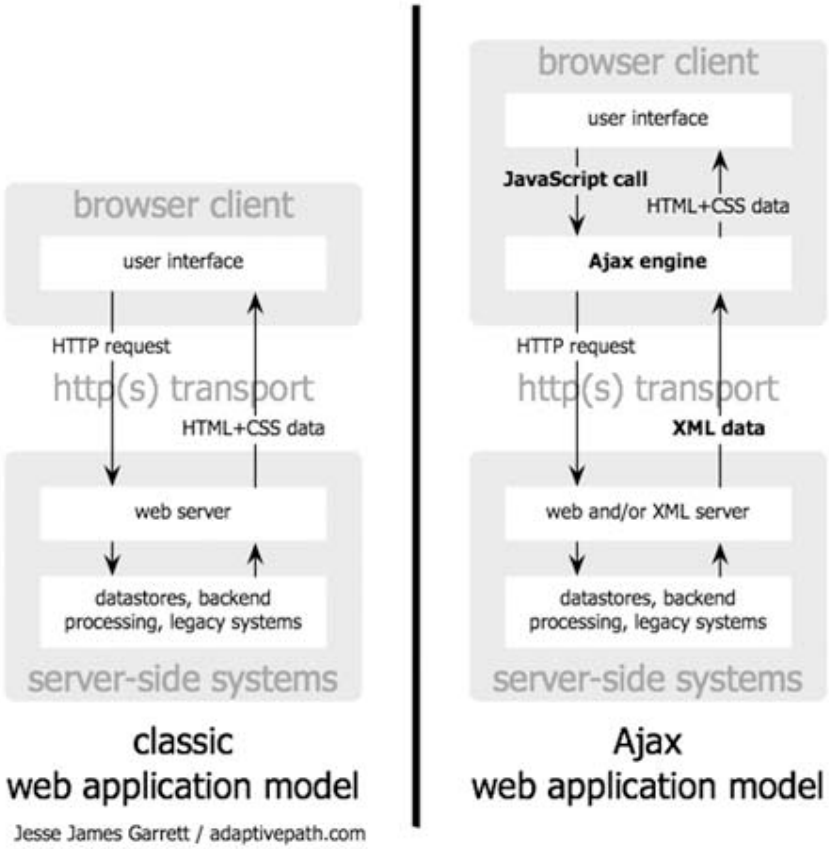
b. Tampilan dinamis dan interaksi menggunakan DOM (*Document Object Model*).

c. Pertukaran dan manipulasi data menggunakan XML (*Extensible Markup Language*) dan XSLT *(Extensible Stylesheet Language Transformations)*.

d. Pengambilan data asinkron menggunakan XMLHttpRequest.

e. Serta JavaScript yang menggabungkan semuanya.

Sebuah cara kerja *web* klasik adalah seperti ini: Semua aksi yang dilakukan pengguna akan memicu permintaan HTTP (HTTP *request*) kepada *server*. *Server* melakukan proses dan mengembalikan halaman HTML ke klien.

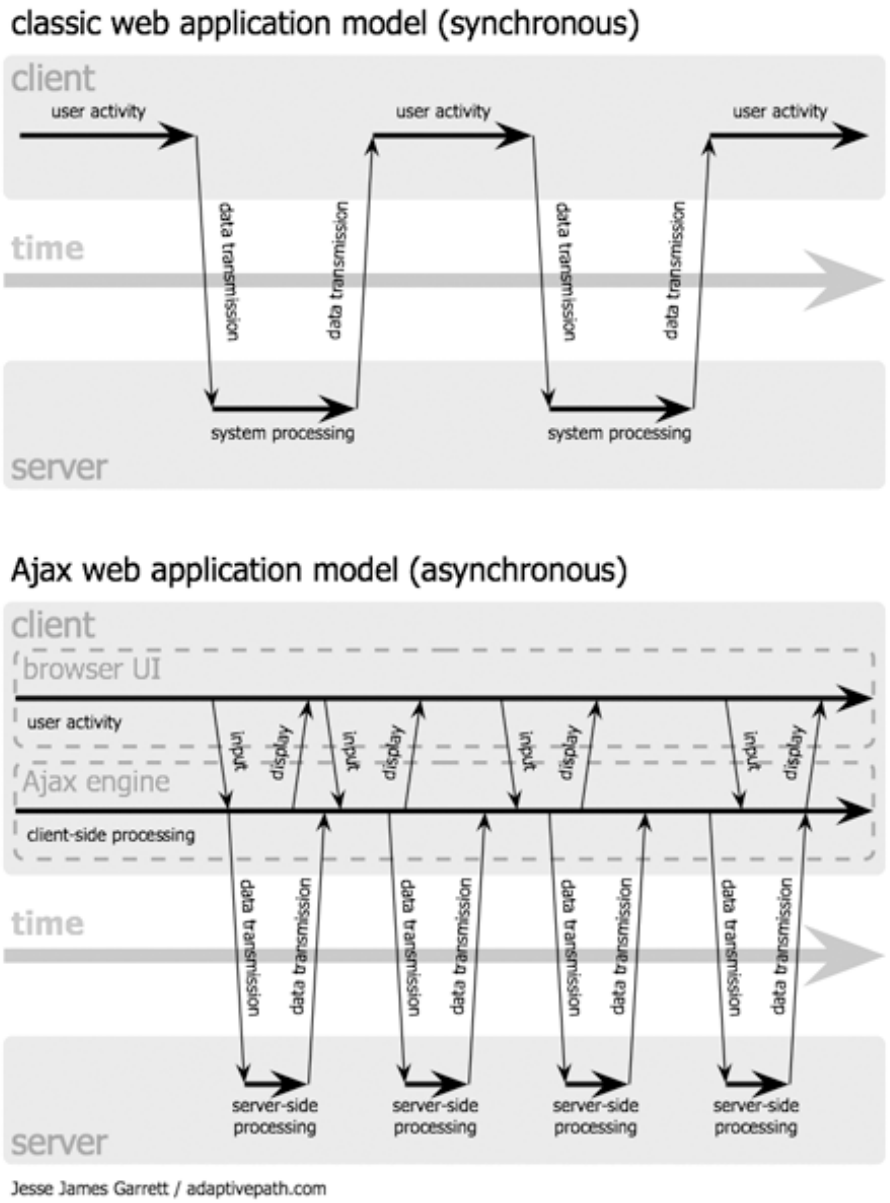


Gambar 2.8 Perbandingan Aplikasi *Web* Klasik dengan Ajax

Aplikasi *web* menggunakan Ajax mengeliminasi sifat interaksi start-stop-start-stop dalam sebuah *web* dengan memperkenalkan penengah, sebuah mesin Ajax, antara pengguna dan server. Dengan menambahkan sebuah lapisan di dalam aplikasi, sepertinya hal ini akan membuat aplikasi menjadi kurang responsif namun yang terjadi adalah sebaliknya.

Daripada memuat sebuah halaman *web*, saat dimulainya sesi, *browser* akan memuat sebuah mesin Ajax yang dibuat menggunakan JavaScript dan diletakkan di sebuah frame yang tersembunyi. Mesin ini bertanggung jawab baik memuat antarmuka yang dilihat pengguna dan berkomunikasi dengan *server*. Mesin ajax ini memungkinkan interaksi pengguna dengan aplikasi berjalan asinkron (independen terhadap komunikasi dengan *server*), sehingga pengguna tidak pernah melihat sebuah halaman kosong dengan simbol jam pasir didalamnya, menunggu *server* melakukan sesuatu.

Setiap aksi user yang seharusnya menghasilkan permintaan HTTP (HTTP *request*) akan digantikan dengan panggilan JavaScript ke mesin Ajax. Sebagai hasilnya, respon ke aksi user yang tidak memerlukan umpan balik ke *server* (seperti validasi data, mengubah data yang merada di memori komputer, dan beberapa navigasi), akan ditanganin oleh mesin Ajax tersebut. Jika mesin memerlukan sesuatu dari *server* agar dapat merespon seperti menyerahkan data untuk diproses, memuat kode antarmuka tambahan, atau mengambil data baru, mesin Ajax memuat permintaan tersebut secara asinkron, biasanya menggunakan XML, tanpa menunda interaksi pengguna dengan aplikasi.



Gambar 2.9 Model Sinkron dan Asinkron

2.6 JQuery

Menurut Resig(2006) JQuery adalah sebuah perpustakaan JavaScript(JavaScript Library) yang menyederhanakan proses perpindahan, menanganan peristiwa, animasi, dan interaksi Ajax pada dokumen HTML untuk pembuatan *web* yang cepat. jQuery didesain untuk merubah cara menulis JavaScript (JQuery Project,2006).

Untuk dapat menggunakan jQuery kita harus mengunduh terlebih dahulu kode jQuery dan memasukkannya ke dalam proyek yang akan kita buat dan menambahkan kode khusus pada halaman HTML kita yang ingin menggunakan jQuery.

Contoh cara menggunakan jQuery:

```
<!doctype html>
<html>
<head>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
</script>
</head>
<body>
<a href="http://jquery.com/">jQuery</a>
</body>
</html>
```

Ubah atribut *src* pada tag script ke tempat kopi jquery.js berada, contoh jika jquery.js berada di direktori yang sama dengan file HTML:

```
<script type="text/javascript" src="jquery.js">
</script>
```

Web browser yang mendukung jQuery adalah Mozilla Firefox mulai versi 2.0 ke atas, Internet Explorer mulai versi 6 ke atas, Safari mulai versi 3 ke atas, Opera mulai versi 9 ke atas, dan Google Chrome mulai versi 1 ke atas

2.7 ASP.net dan C#

Menurut Liberty (2006,p1), microsoft pertama kali mengumumkan ASP.NET 1.0 dan .NET Framework pada July,2000. Pada umumnya, .NET adalah sebuah kerangka kerja (*framework*) pengembangan aplikasi yang menyediakan antar muka program aplikasi yang baru dan *Application Programming Interface* dari sistem operasi Windows klasik, terutama Windows 2000, juga menambahkan beberapa teknologi yang berbeda yang dimunculkan Microsoft pada akhir dekade 1990an.

Pada saat ini .NET terdiri dari:

a. Kompiler untuk 5 bahasa yang berbeda (C#, Visual Basic, Managed C++, J#, dan JScript)

b. Sebuah pustaka class (*class libraries*) yang saling berhubungan, dikenal sebagai *Framework Class Library* (FCL), yang menyertakan dukungan untuk aplikasi windows dan web, akses data, dan lain-lain

c. *Common Language Runtime* (CLR), sebuah mesin berorientasi object dari *Framework* ini yang menerjemahkan intermediate code yang dihasilkan kompiler bahasa ke dalam native code yang diperlukan untuk mengeksekusi aplikasi

Menurut Dietel (2008,p9), C# adalah sebuah bahasa pemrograman yang berorientasi objek dan didesain secara spesifik untuk *platform* .NET sebagai bahasa yang memungkinkan programmer untuk mudah bermigrasi ke .NET. C# memiliki akar dari C, C++, dan java, mengadaptasikan fitur-fitur terbaik dan menambahkan fitur-fitur tersendiri. C# memiliki akses *ke class library* yang kuat dari komponen-komponen yang sudah dibangun sebelumnya, memungkinakn programmer untuk mengembangkan

aplikasi dengan cepat, C# dan Visual Basic menggunakan *.NET Framework Class Library* yang sama.

Bahasa pemrograman C# yang asli distandarisasi oleh Ecma International pada Desember 2002. Sejak saat itu, Microsoft telah menawarkan beberapa ekstensi bahasa yang telah diadopsi sebagai bagian dari standar Ecma C# yang telag direvisi.

2.8 Basis data dan Standard Query Language

Menurut Connolly dan Begg (2002, p15), basisdata adalah koleksi data - data yang saling berhubungan secara logis, yang didesain agar dapat memenuhi kebutuhan dari sistem sebuah organisasi.

Keunggulan Basisdata :

a. Dapat digunakan secara serempak oleh banyak pengguna,

b. Semua data terintegrasi,sehingga mengurangi jumlah duplikasi data

Menurut Connolly dan Begg (2002, p16), *Database Management System* (DBMS) adalah sebuah sistem piranti lunak sehingga pengguna dapat menentukan, membuat, memelihara, dan mengendalikan akses dalam sebuah basisdata.

Structure Query Language (SQL) adalah bahasa yang digunakan untuk memanipulasi beberapa tabel data yang saling berhubungan. Keunggulan utama dari SQL adalah SQL sudah diakui sebagai bahasa standar pada *server* basisdata. Sehingga SQL dapat menghubungkan beberapa tabel dari *platform* yang berbeda.

Microsoft SQL Server 2005merupakan salah satu produk *Relational Database Management System* (RDMS). Microsoft SQL Server 2005 menyediakan fitur yang dibutuhkan untuk melakukan pengaturan, integrasi, analisis, dan laporan data. Fungsi

utama Microsoft SQL Server 2005 sebagai server basis data yang mengatur semua proses penyimpanan data dan transaksi suatu aplikasi

Fungsi – fungsi dasar SQL :

a. Perintah SELECT

Digunakan untuk mengambil atau menampilkan data tabel basisdata. Aturan perintah SELECT:

SELECT [DISTINCT] <nama_field> [,<nama_field n>]

FROM <nama_tabel1>[<nama_tabel n>]

WHERE <kondisi1> [OR | AND <kondisi2>];

b. Perintah INSERT

Digunakan untuk menambah *record* baru ke dalam tabel. Aturan perintah INSERT :

INSERT INTO <nama_tabel> [(<nama_field>,<field n>)]

VALUES (<isi_field1> [,<isi_field n>]);

c. Perintah DELETE

Digunakan untuk menghapus satu atau beberapa *record* dalam tabel. Aturan perintah DELETE :

DELETE FROM <nama_tabel>

WHERE <kondisi1> [OR | AND <kondisi2>];

d. Perintah UPDATE

Digunakan untuk mengubah satu atau beberapa *record* dalam tabel. Aturan perintah UPDATE :

UPDATE <nama_tabel>

SET <nama_field1 yang akan diupdate> = <nilai baru> [,<nama_field n yang akan diupdate> = <nilai baru n>

WHERE <kondisi1> [OR | AND <kondisi2>];

2.9 Entity Relationship Diagram (ERD)

Menurut Whitten, Bentley, dan Dittman (2004, p295-307), *entity relationship diagram* adalah permodelan data yang menggunakan beberapa notasi untuk menggambarkan data yang berhubungan dengan *entity* dan hubungan (*relationship*) yang dideskripsikan oleh data tersebut.

Menurut Connolly (2002,p279), perancangan basis data merupakan proses menciptakan perancangan untuk basis data yang akan mendukung operasi dan tujuan perusahaan. Dalam merancang suatu basis data, digunakan metodologi-metodologi yang membantu dalam tahap perancangan basis data. Metodologi perancangan adalah pendekatan struktur dengan menggunakan prosedur, teknik, alat, serta bantuan dokumen untuk membantu dan memudahkan dalam proses perancangan. Menurut Connolly (2002,p418) dengan menggunakan teknik metode disain ini dapat membantu dalam merencanakan, mengatur, mengontrol, dan mengevaluasi *database development project.*

2.10 Lima Faktor Manusia Terukur

Menurut Shneiderman (2010,p31), 5 faktor manusia terukur adalah:

a. Waktu belajar : berapa lama waktu yang dibutuhkan bagi tipe komunitas pengguna tertentu untuk belajar bagaimana melakukan aksi yang sesuai untuk sekumpulan tugas?

b. Kecepatan kinerja :berapa lama waktu yang dibutuhkan untuk melakukan tugas tertentu?

c. Tingkat kesalahan oleh pengguna : berapa banyak dan jenis kesalahan apa yang dibuat oleh pengguna?

d. Daya ingat : seberapa baik pengguna mempertahankan atau mengingat pengetahuannya setelah beberapa waktu?

e. Kepuasan subjektif : bagaimana selera pemakai terhadap berbagai aspek sistem?

2.11 Delapan Aturan Emas Perancangan Antarmuka Pengguna

Menurut Shneiderman (2010,p88), delapan aturan emas perancangan antarmuka pengguna adalah:

a. Berusaha untuk konsisten

Aksi yang konsisten diperlukan dalam situasi dan kondisi yang mirip atau memiliki kesamaan. Konsistensi diperlukan agar pengguna dapat lebih mudah dan terbiasa dalam menggunakan aplikasi.

b. Menyediakan kemudahan penggunaan yang universal

Mengenali kebutuhan berbagai pengguna yang berbeda dan memfasilitasi perubahan dari isi. Perbedaan antara pemula dengan ahli, perbedaan usia pemakai, *user* yang abnormal, dan perbedaan teknologi dapat mempengaruhi kebutuhan dalam melakukan desain.

c. Memberikan umpan balik yang informatif

Untuk setiap aksi pengguna, harus ada umpan balik kepada pengguna.

d. Berikan dialog untuk memberikan penutupan (keadaan akhir).

Serangkaian aksi harus disusun menjadi kelompok, dengan awal, pertengahan dan akhir. Umpan balik yang informatif diberikan saat suatu rangkaian aksi telah selesai.

e. Memberikan penanganan kesalahan yang sederhana.

Sebisa mungkin, harus merancang sistem sehingga pengguna tidak dapat melakukan kesalahan yang serius. Jika pengguna melakukan kesalahan, antarmuka harus mendeteksi kesalahan, dan menawarkan penanganan kesalahan yang sederhana.

f. Memungkinkan pembalikan aksi yang mudah.

Sebisa mungkin, aksi yang dilakukan harus dapat reversible untuk menghilangkan kegelisahan karena pengguna mengetahui bahwa jika terjadi kesalahan dapat dilakukan pembalikan aksi.

g. Mendukung pusat kendali internal / *internal locus of control*

Pengguna mahir/ahli menginginkan perasaan bahwa mereka mempunyai kontrol pada sistem dan sistemlah yang merespon sesuai dengan aksi mereka. Buatlah pengguna merasa mereka sebagai pihak yang memulai aksi bukan sebagai perespon.

h. Mengurangi beban ingatan jangka pendek.

Keterbatasan kapasitas manusia untuk memproses informasi dalam ingatan jangka pendek mengharuskan desainer menghindari antarmuka yang membuat pengguna harus mengingat informasi dari satu layar dan menggunakan informasi tersebut pada layar lainnya.

2.12 Unified Modeling Language (UML)

Menurut Whitten (2004,p408), UML adalah satu kumpulan konvensi pemodelan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak yang terkait dengan objek. UML tidak menentukan metode untuk sistem-sistem pengembangan, hanya sebuah catatan yang saat ini telah diterima luas sebagai standar untuk pemodelan objek.

Konsep sistem untuk pemodelan Objek:

1. Objek, Atribut, Metode, dan Enkapsulasi
2. Kelas, Generalisasi, dan Spesialisasi
3. Hubungan Objek /Kelas
4. Pesan dan Mengirim Pesan
5. Polymorphism

Menurut Whitten (2004,p418), UML menawarkan diagram yang dikelompokan menjadi lima perspektif berbeda untuk memodelkan suatu sistem.

Kelompok-kelompok diagram tersebut adalah:

1. Diagram Model *Use-Case*

   Secara grafis, menggambarkan interaksi antara sistem, sistem eksternal dan pengguna.

2. Diagram Struktur Statis

   UML menawarkan dua diagram untuk memodelkan struktur objek sistem:

   a. Diagram Kelas: menggambarkan struktur objek sistem. Diagram ini menunjukkan kelas objek yang menyusun sistem dan juga hubungan antar kelas objek tersebut.

b. Diagram Objek: serupa dengan diagram kelas, tetapi daripada menggambarkan kelas objek, diagram objek memodelkan sebuah instance objek aktual dengan menunjukkan nilai-nilai saat ini.

3. Diagram Interaksi

Diagram interaksi memodelkan sebuah interaksi, terdiri dari satu set objek, hubungan-hubungannya, dan pesan yang terkirim di antara objek. UML menawarkan dua diagram untuk memodelkan interaksi ini:

a. Diagram rangkaian/*Sekuens*: secara grafis menggambarkan bagaimana objek berinteraksi dengan satu sama lain melalui pesan pada eksekusi sebuah use case atau operasi. Diagram ini mengilustrasikan bagaimana pesan terkirim dan diterima di antara objek dan dalam sekuens apa.

b. Diagram kolaborasi: diagram ini menggambarkan interaksi/kolaborasi antar objek dalam format jaringan

4. Diagram *State*

Diagram juga memodelkan tingkah laku dinamis dari sistem. Ada 2 buah diagram untuk bagian ini:

a. Diagram *statechart*: digunakan untuk memodelkan tingkah laku objek khusus yang dinamis. Diagram ini mengilustrasikan siklus hidup objek, berbagai keadaan yang dapat diasumsikan oleh objek dan kejadian-kejadian yang menyebabkan objek beralih dari satu state ke state lain.

b. Diagram aktivitas: secara grafis digunakan untuk menggambarkan rangkaian aliran aktivitas baik proses bisnis maupun use-case. Diagram ini juga dapat digunakan untuk memodelkan aksi yang akan

dilakukan saat sebuah operasi dieksekusi, dan memodelkan hasil dari aksi tersebut.

5. Diagram Implementasi

Diagram implementasi juga memodelkan struktur sistem informasi:

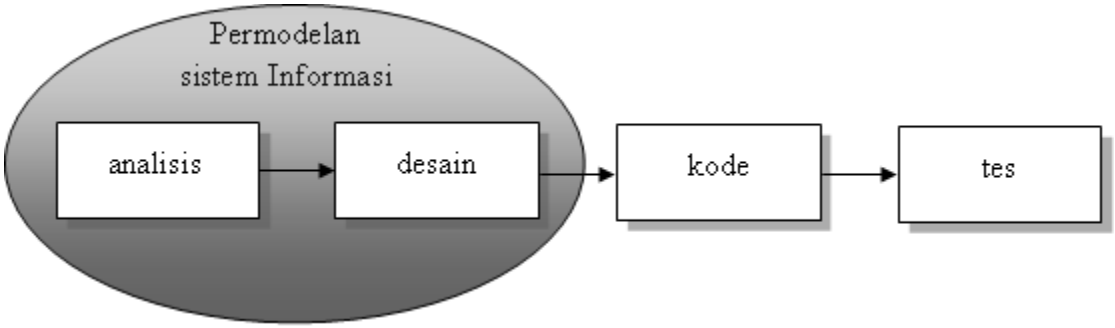
a. Diagram komponen: digunakan untuk menggambarkan organisasi dan ketergantungan komponen-komponen software sistem.

b. Diagram penguraian/dekomposisi: mendeskripsikan arsitektur fisik dalam istilah "node" untuk hardware dan software dalam sistem.

2.13 Proses Pengembangan Sistem dengan Model Sekuensial Linier

Menurut Pressman (2002, p36), model sekuensial linier sering disebut juga dengan "siklus kehidupan klasik" atau "model air terjun (*Waterfall Model*)". Model ini adalah paradigma rekayasa perangkat lunak yang paling tua dan paling banyak dipakai.

Model ini mengusulkan sebuah pendekatan kepada pengembangan perangkat lunak yang sistematik dan sekuensial yang dimulai pada sebuah tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian dan pemeliharaan.



Gambar 2.10 Model Sekuensial Linier

Dimodelkan setelah siklus rekayasa konvensional model, aktivitas-aktivitas yang menglingkupi model sekuensial linier adalah:

1. Rekayasa dan pemodelan sistem/ informasi

   Perangkat lunak merupakan bagian dari sebuah sistem yang lebih besar. Oleh karena itu pekerjaan selalu dimulai dengan membangun syarat dari semua elemen sistem dan mengalokasikan beberapa subset/bagian dari kebutuhan ke dalam perangkat lunak. Rekayasa dan analisis sistem menyangkut pengumpulan kebutuhan pada tingkat sistem dengan sejumlah kecil analisis serta desain tingkat puncak. Proses ini juga mencakup pengumpulan kebutuhan pada tingkat bisnis strategis dan tingkat area bisnis

2. Analisis kebutuhan perangkat lunak

   Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada perangkat lunak. Untuk memahami sifat program yang dibangun, perekaya perangkat lunak (analis) harus memahami domain informasi, tingkah laku, unjuk kerja, dan antarmuka yang diperlukan. Kebutuhan baik untuk sistem maupun perangkat lunak didokumentasikan dan dilihat lagi dengan pelanggan.

3. Desain

   Desain perangkat lunak berfokus pada empat atribut sebuah program berbeda yaitu struktur data, arsitektur perangkat lunak, representasi antarmuka, dan detail (algoritma) prosedural. Proses desain menerjemahkan syarat/kebutuhan ke dalam sebuah representasi perangkat lunak yang dapat diperkirakan demi kualitas sebelum pemunculan kode. Kebutuhan sistem, kebutuhan perangkat lunak, dan desain wajib di dokumentasikan dan menjadi bagian dari konfigurasi perangkat lunak.

4. Generasi kode

   Proses penerjemahan desain ke dalam bentuk mesin yang bisa dibaca.

5. Pengujian

   Proses pengujian berfokus pada logika internal perangkat lunak dengan memastikan bahwa semua pernyataan sudah diuji dan juga pada eksternal fungsional, dengan mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa input yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

6. Pemeliharaan

   Perangkat lunak akan sangat mungkin mengalami perubahan setelah disampaikan kepada pelanggan, hal itu dapat terjadi karena kesalahan-kesalahan ditemukan, karena perangkat lunak harus disesuaikan untuk mengakomodasi perubahan-perubahan di dalam lingkungan eksternalnya, atau karena pelanggan membutuhkan perkembangan fungsional atau unjuk kerja. Pemeliharaan perangkat lunak mengaplikasikan lagi setiap fase program sebelumnya dan tidak membuat baru lagi.