

BAB II LANDASAN TEORI

2.1 Pengantar *E-Learning*

2.1.1 Pengertian *E-Learning*

Istilah *e-Learning* mengandung pengertian yang sangat luas, sehingga banyak pakar yang menguraikan tentang definisi *e-Learning* dari berbagai sudut pandang. Salah satu definisi yang cukup dapat diterima banyak pihak misalnya dari *Darin E. Hartley* yang menyatakan:

e-Learning merupakan suatu jenis belajar mengajar yang memungkinkan tersampainya bahan ajar ke siswa dengan menggunakan media Internet, Intranet atau media jaringan komputer lain.

LearnFrame.Com dalam *Glossary of e-learning Terms* menyatakan suatu definisi yang lebih luas bahwa:

e-Learning adalah sistem pendidikan yang menggunakan aplikasi elektronik untuk mendukung belajar mengajar dengan media Internet, jaringan komputer, maupun komputer standalone.

2.1.2 Keuntungan Menggunakan *E-Learning*

Keuntungan menggunakan *e-Learning* diantaranya adalah sebagai berikut:

- Menghemat waktu proses belajar mengajar
- Mengurangi biaya perjalanan
- Menghemat biaya pendidikan secara keseluruhan (infrastruktur, peralatan, buku-buku)
- Menjangkau wilayah geografis yang lebih luas
- Melatih pembelajar lebih mandiri dalam mendapatkan ilmu pengetahuan

2.1.3 Aplikasi *E-Learning* dari masa ke masa

Uraian singkat tentang perkembangan *e-Learning* dari masa ke masa adalah seperti di bawah [Cross, 2002]:

1990: CBT (*Computer Based Training*)

Era dimana mulai bermunculan aplikasi *e-learning* yang berjalan dalam PC *standalone* ataupun berbentuk kemasan CD-ROM. Isi berupa materi dalam bentuk tulisan maupun *multimedia* (video dan audio) dalam format MOV, MPEG-1 atau AVI. Perusahaan perangkat lunak Macromedia mengeluarkan tool pengembangan bernama *Authorware*, sedangkan *Asymetrix* (sekarang bernama Click2learn) juga mengembangkan perangkat lunak bernama *Toolbook*.

1994: Paket-Paket CBT

Seiring dengan mulai diterimanya CBT oleh masyarakat, sejak tahun 1994 muncul CBT dalam bentuk paket-paket yang lebih menarik dan diproduksi secara massal.

1997: LMS (*Learning Management System*)

Seiring dengan perkembangan teknologi internet di dunia, masyarakat dunia mulai terkoneksi dengan internet. Kebutuhan akan informasi yang cepat diperoleh menjadi mutlak, dan jarak serta waktu bukanlah halangan lagi. Disinilah muncul sebutan *Learning Management System* atau biasa disingkat dengan LMS. Perkembangan LMS yang semakin pesat membuat pemikiran baru untuk mengatasi masalah interoperability antar LMS yang ada dengan suatu standard. Standard yang muncul misalnya adalah standard yang dikeluarkan oleh AICC (Airline Industry CBT Committee), IMS, IEEE LOM, ARIADNE

1999: Aplikasi *e-Learning* Berbasis Web

Perkembangan LMS menuju ke aplikasi *e-learning* berbasis web secara total, baik untuk pembelajar (*learner*) maupun administrasi belajar mengajarnya. LMS mulai digabungkan dengan situs-situs portal yang pada saat ini boleh dikata menjadi barometer situs-situs informasi, majalah, dan surat kabar dunia. Isi juga semakin kaya dengan berpaduan *multimedia*, *video streaming*, serta penampilan interaktif dalam berbagai pilihan format data yang lebih standard, berukuran kecil dan stabil.

2.2 Learning Management System [DWI09]

Dalam *e-learning*, materi pembelajaran perlu dilakukan pengelolaan pembelajaran agar menghasilkan aktivitas, komunikasi dan interaksi. Pengelolaan pembelajaran yang dimaksud adalah seperti gambar yang ada di bawah ini

Pengelolaan Pembelajaran		
Dosen:		Mahasiswa:
Menentukan tujuan	Administrator:	Memilih matakuliah
Membuat silabi	Mendaftar mahasiswa	Mengikuti kuliah
Menyiapkan materi	Jadwal kuliah	Mencari sumber belajar
Menyampaikan materi	Jadwal ujian	Membaca materi
Mengelola kelas	Mengolah nilai	Mengerjakan tugas
Berinteraksi		Mengerjakan tugas
Mengevaluasi		Berinteraksi
		ujian

Gambar 2. Pengelolaan Pembelajaran

Dan dari gambar diatas, muncul pertanyaan bagaimana semua kegiatan pengelolaan pembelajaran tersebut dapat diintegrasikan dalam suatu portal *e-learning* ? jawabannya adalah LMS (*Learning Management System*). LMS merupakan perangkat lunak yang digunakan untuk membuat materi perkuliahan *online* (berbasis web) dan mengelola kegiatan pembelajaran serta hasil-hasilnya.

LMS sendiri mempunyai ruang lingkup :

- Administrasi
- Penyampaian materi
- Penilaian
- Monitoring
- Komunikasi

Pengguna LMS terdiri dari tiga bagian dengan fungsinya yaitu:

1. Administrator
 - Pendaftaran mata kuliah

- Menentukan dosen dan mata kuliah
 - Melakukan *back up* data
 - Pengelolaan website
2. Dosen/instruktur
- Membuat silabi
 - Manajemen file
 - Membuat soal tes, quiz
 - Memonitor aktivitas mahasiswa
 - Berinteraksi dengan mahasiswa (*forum, chat, email*)
 - Melihat kemajuan mahasiswa
3. Mahasiswa/siswa
- Akses silabi, materi
 - Melihat hasil penilaian
 - Mengerjakan tes
 - Berinteraksi dengan dosen dan mahasiswa lainnya (*forum, chat, email*)

2.3 Media Pembelajaran

Media berasal dari bahasa latin merupakan bentuk jamak dari “Medium” yang secara harfiah berarti “Perantara” atau “Pengantar” yaitu perantara atau pengantar sumber pesan dengan penerima pesan. Beberapa ahli memberikan definisi tentang media pembelajaran. Schramm (1977) mengemukakan bahwa media pembelajaran adalah teknologi pembawa pesan yang dapat dimanfaatkan untuk keperluan pembelajaran. Sementara itu, Briggs (1977) berpendapat bahwa media pembelajaran adalah sarana fisik untuk menyampaikan isi/materi pembelajaran seperti : buku, film, video dan sebagainya. Sedangkan, *National Education Assocation* (1969) mengungkapkan bahwa media pembelajaran adalah sarana komunikasi dalam bentuk cetak maupun pandang-dengar, termasuk teknologi perangkat keras. Dari ketiga pendapat di atas disimpulkan bahwa media pembelajaran adalah segala sesuatu yang dapat menyalurkan pesan, dapat

merangsang pikiran, perasaan, dan kemauan peserta didik sehingga dapat mendorong terciptanya proses belajar pada diri peserta didik.

Brown (1973) mengungkapkan bahwa media pembelajaran yang digunakan dalam kegiatan pembelajaran dapat mempengaruhi terhadap efektivitas pembelajaran. Pada mulanya, media pembelajaran hanya berfungsi sebagai alat bantu guru untuk mengajar yang digunakan adalah alat bantu visual. Sekitar pertengahan abad Ke-20 usaha pemanfaatan visual dilengkapi dengan digunakannya alat audio, sehingga lahirlah alat bantu audio-visual. Sejalan dengan perkembangan ilmu pengetahuan dan teknologi (IPTEK), khususnya dalam bidang pendidikan, saat ini penggunaan alat bantu atau media pembelajaran menjadi semakin luas dan interaktif, seperti adanya komputer dan internet. Media pembelajaran memiliki beberapa fungsi, diantaranya

1. Media pembelajaran dapat mengatasi keterbatasan pengalaman yang dimiliki oleh para peserta didik. Pengalaman tiap peserta didik berbeda-beda, tergantung dari faktor-faktor yang menentukan kekayaan pengalaman anak, seperti etnososial, buku, kesempatan melancong, dan sebagainya. Media pembelajaran dapat mengatasi perbedaan tersebut. Jika peserta didik tidak mungkin dibawa ke obyek langsung yang dipelajari, maka obyeknyalah yang dibawa ke peserta didik. Obyek dimaksud bisa dalam bentuk nyata, miniatur, model, maupun bentuk gambar-gambar yang dapat disajikan secara audio visual dan audial.
2. Media pembelajaran dapat melampaui batasan ruang kelas. Banyak hal yang tidak mungkin dialami secara langsung di dalam kelas oleh para peserta didik tentang suatu obyek, yang disebabkan, karena : (a) obyek terlalu besar; (b) obyek terlalu kecil; (c) obyek yang bergerak terlalu lambat; (d) obyek yang bergerak terlalu cepat; (e) obyek yang terlalu kompleks; (f) obyek yang bunyinya terlalu halus; (f) obyek mengandung berbahaya dan resiko tinggi. Melalui penggunaan media yang tepat, maka semua obyek itu dapat disajikan kepada peserta didik.
3. Media pembelajaran memungkinkan adanya interaksi langsung antara peserta didik dengan lingkungannya.
4. Media menghasilkan keseragaman pengamatan

5. Media dapat menanamkan konsep dasar yang benar, konkrit, dan realistik.
6. Media membangkitkan keinginan dan minat baru.
7. Media membangkitkan motivasi dan merangsang anak untuk belajar.
8. Media memberikan pengalaman yang integral/menyeluruh dari yang konkrit sampai dengan abstrak

Terdapat berbagai jenis media belajar, diantaranya:

1. *Media Visual* : grafik, diagram, chart, bagan, poster, kartun, komik
2. *Media Audial* : radio, tape recorder, laboratorium bahasa, dan sejenisnya
3. *Projected still media* : *slide; over head proyektor (OHP), in focus* dan sejenisnya
4. *Projected motion media* : film, televisi, video (VCD, DVD, VTR), komputer dan sejenisnya.

Sejalan dengan perkembangan IPTEK penemuan media, baik yang bersifat visual, audial, *projected still media* maupun *projected motion media* bisa dilakukan secara bersama dan serempak melalui satu alat saja yang disebut Multi Media. Contoh : dewasa ini penggunaan komputer tidak hanya bersifat *projected motion media*, namun dapat memuat semua jenis media yang bersifat interaktif.

Allen mengemukakan tentang hubungan antara media dengan tujuan pembelajaran, sebagaimana tertera dalam tabel 2.1

Tabel 2.1 hubungan antara media dengan tujuan pembelajaran

Jenis Media	1	2	3	4	5	6
Gambar Diam	S	T	S	S	R	R
Gambar Hidup	S	T	T	T	S	S
Televisi	S	S	T	S	R	S
Objek Tiga Dimensi	R	T	R	R	R	R
Rekaman Audio	S	R	R	S	R	S
Programmed Instruction	S	S	S	T	R	S
Demonstrasi	R	S	R	T	S	S
Buku teks tercetak	S	R	S	S	R	S

Keterangan tabel 2.1 :

R = Rendah S = Sedang T= Tinggi

1 = Belajar Informasi faktual

2 = Belajar pengenalan visual

3 = Belajar prinsip, konsep dan aturan

4 = Prosedur belajar

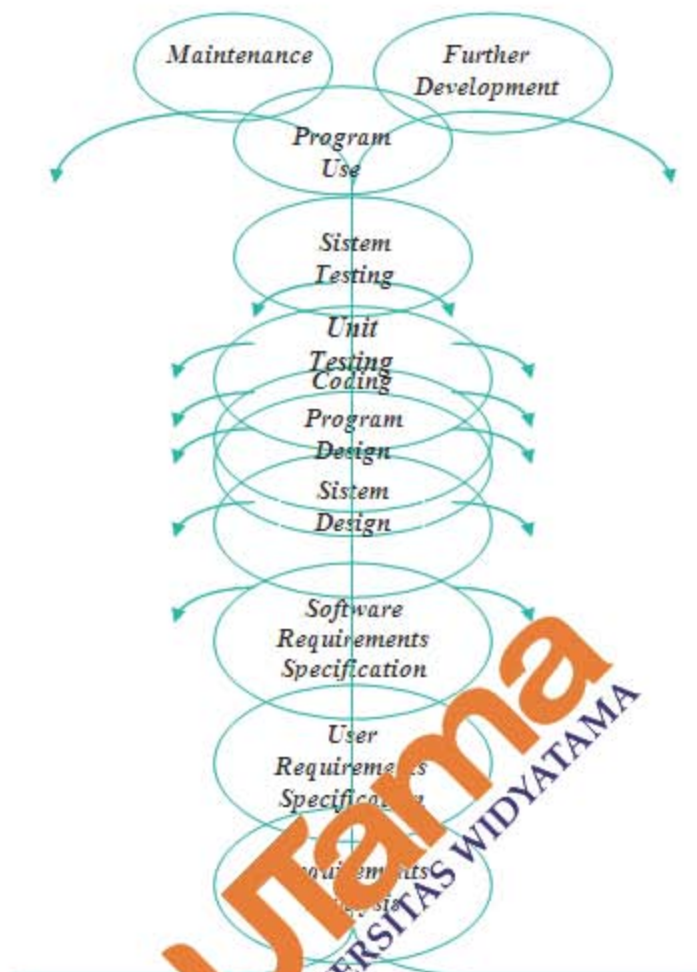
5= Penyampaian keterampilan persepsi motorik

6 = Mengembangkan sikap, opini dan motivasi

Kriteria yang paling utama dalam pemilihan media bahwa media harus disesuaikan dengan tujuan pembelajaran atau kompetensi yang ingin dicapai. Contoh : bila tujuan atau kompetensi peserta didik bersifat menghafalkan kata-kata tentunya media audio yang tepat untuk digunakan. Jika tujuan atau kompetensi yang dicapai bersifat memahami isi bacaan maka media cetak yang lebih tepat digunakan. Kalau tujuan pembelajaran bersifat motorik (gerak dan aktivitas), maka media film dan video bisa digunakan. Di samping itu, terdapat kriteria lainnya yang bersifat melengkapi (komplementer), seperti: biaya, ketepatangunaan; keadaan peserta didik, ketersediaan; dan mutu teknis.

2.4 Metode Rekayasa Perangkat Lunak dengan *Fountain Model*

Metode *fountain* model merupakan model metode rekayasa perangkat lunak yang diterapkan untuk pembangunan sistem yang bersifat *object oriented*. Tahapan-tahapan pengembangan yang dilakukan adalah Analisa kebutuhan, spesifikasi kebutuhan *user*, perancangan sistem, perancangan program, *coding*, unit *testing*, sistem *testing*, implementasi, pemeliharaan dan pengembangan lebih lanjut. Tahapan-tahapan tersebut saling terkait dan saling mempengaruhi (*life cycle*). Keterkaitan tahapan-tahapan dapat dilihat pada gambar 2.2 halaman II-8.



Gambar 2.2 Metode *Fountain Model*

- a. Tahap *requirements analysis*: menganalisa kebutuhan sistem dan semua yang berkaitan dengan proses yang terjadi pada sistem yang diperlukan untuk membangun sistem sehingga dapat berjalan dengan baik dan maksimal.
- b. Tahap *user requirement specification*: menjelaskan tentang kebutuhan dari sisi pengguna (*user*), sehingga *user* dapat menjalankan sistem dengan baik.
- c. Tahap *software requirements specification*: menjelaskan tentang kebutuhan sistem dari sisi *software* sehingga sistem yang ada dapat berjalan dengan baik dan maksimal.
- d. Tahap *sistem design*: Pada tahap perancangan ini diberikan gambaran umum yang jelas kepada pengguna dan rancang bangun yang lengkap tentang sistem yang akan dikembangkan kepada pihak-pihak yang terlibat dalam pengembangan sistem. Perancangan disini dilakukan dengan permodelan menggunakan metode *Object Oriented* dengan *tools* UML (*Unified*

Modeling Language). Tahapan perancangan sistem disini dibagi menjadi dua bagian, yaitu perancangan global dan perancangan rinci. Perancangan global dilakukan untuk memberikan gambaran umum kepada pengguna tentang sistem yang dirancang dan sebagai persiapan untuk tahap perancangan rinci. Perancangan rinci dilakukan untuk memberikan gambaran rancang bangun yang lengkap kepada pemrogram dan pihak-pihak lain yang terlibat dalam pengembangan sistem sebagai persiapan untuk tahap implementasi^[9]

- e. **Tahap program design:** proses perancangan dan pembuatan kerangka program yang mengacu pada tahap sebelumnya, sehingga program yang dibuat dapat memenuhi kebutuhan dari sistem dan mencakup penyelesaian masalah yang ada.
- f. **Tahap unit testing :** proses pengujian terhadap setiap unit yang ada pada sistem sehingga dapat unit tersebut dapat melakukan kerja sesuai dengan yang diharapkan.
- g. **Tahap coding:** proses penterjemahan tahap perancangan ke dalam bahasa pemrograman. Proses coding meliputi unit-unit yang ada pada keseluruhan sistem.
- h. **Tahap testing:** Pengujian perangkat lunak adalah elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, desain, dan pengkodean^[9]. Sejumlah aturan yang berfungsi sebagai sasaran pengujian adalah:
 1. Pengujian adalah proses eksekusi suatu program dengan maksud menemukan kesalahan.
 2. *Test case* yang baik adalah *test case* yang memiliki *probabilitas* yang tinggi untuk menemukan kesalahan yang belum pernah ditemukan sebelumnya.
 3. Pengujian yang sukses adalah pengujian yang mengungkap semua kesalahan yang sebelumnya.

Pengujian perangkat lunak tidak pernah berakhir, pengujian hanya akan berpindah dari pengembang ke pelanggan. Setiap kali pelanggan menggunakan perangkat lunak tersebut berarti pengujian sedang dilakukan^[9].

- i. **Tahap *program use***: proses pengaplikasian dan pemakaian program yang telah dibuat untuk memastikan apakah tujuan yang semula telah terpenuhi.
- j. **Tahap *maintenance***: pada tahap *maintenance*, Setelah dilakukan pengujian dan sistem diyakini telah *valid*, selanjutnya sistem tersebut didistribusikan kepada pengguna. Banyak *software* tidak dapat bertahan 10 sampai 15 tahun, meskipun *software* tersebut dibuat dengan desain dan teknik pengkodean terbaik saat dibuat. *Software* akan memburuk dengan semestinya ketika pindah *platform* sistem operasi yang berbeda, kebutuhan baru dari pengguna atau *software* tidak cukup bisa menjawab kebutuhan fungsional^[9]. Pemeliharaan lebih dari sekedar “memperbaiki kesalahan”, ada empat perbedaan aktifitas dalam pemeliharaan yaitu:

1. *Corrective Maintenance*, adalah sama dengan insensi untuk *software*, atau memperbaiki kesalahan *software*.
2. *Adaptive Maintenance*, adalah memodifikasi *software* untuk *platform* sistem operasi yang berbeda atau memodifikasi *software* untuk mengakomodasi lingkungan eksternal.
3. *Perpective Maintenance* atau *Enhancement*, adalah menambah dan mengenali fungsi tambahan yang bermanfaat diluar kebutuhan fungsional aslinya.
4. *Preventive Maintenance* atau *Reengineering*, adalah pembangunan kembali *software* yang sudah memburuk kinerjanya. Aktifitas ini lebih mudah disamping aktifitas pemeliharaan yang lain.

Hanya 20% dari semua pekerjaan pemeliharaan digunakan untuk melakukan perbaikan kesalahan, sedangkan sisanya 80% digunakan untuk memodifikasi *software* untuk mengakomodasi lingkungan eksternal, menambah dan mengenali fungsi tambahan dan pembangunan kembali *software* untuk digunakan di masa depan^[9].




- k. **Tahap *further development***: pada tahap ini dijelaskan kemungkinan yang ada untuk proses pengembangan sistem.

2.5 Perancangan *System* dengan UML (*Unified Modeling Language*)

2.5.1 *Use Case Diagram*

Diagram yang bersifat statis, diagram ini memperlihatkan himpunan *use case* dan *actor* (suatu jenis khusus dari kelas) diagram ini sangat penting untuk mengorganisasikan dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna. Adapun simbol-simbol yang sering digunakan dalam *Use Case Diagram* adalah :


Tabel 2.2 Simbol-simbol pada *Use Case Diagram*



No.	Simbol	Keterangan
1.		Simbol <i>Actor</i> , menggambarkan aktor pada diagram.
2.		Simbol <i>Use Case</i> , menggambarkan <i>Use Case</i> pada diagram
3.		Simbol <i>Unidirectional Association</i> , menggambarkan relasi antar aktor dan use case

2.5.2 *Class Dalam Model Analisis*

Elemen model yang terdapat dalam model analisis disebut kelas analisis (*analysis class*). Kelas analisis adalah kelas ber-*stereotype* "Boundary", "Control", atau "Entity" yang menggambarkan sebuah konsep awal mengenai "benda" dalam sistem aplikasi yang memiliki tanggung jawab dan perilaku. Kelas analisis akhirnya berkembang menjadi kelas didalam model desain. Adapun simbol-simbol yang sering digunakan dalam *Class* dalam model analisis adalah :

Tabel 2.3 Simbol-simbol pada *Class Model Analisis*

No.	Simbol	Keterangan
1.		Simbol <i>Boundary</i> , menggambarkan batasan kelas pada diagram. Dimana kelas yang memodelkan interaksi antara satu atau lebih <i>actor</i> dengan sistem

2.		Simbol <i>Control</i> , menggambarkan unsure kendali pada diagram.
3.		<i>Entity</i> menggambarkan kelas entitas pada diagram.

2.5.3 Class Diagram

Diagram yang bersifat statis, diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan berorientasi objek, meskipun bersifat statis sering pula diagram kelas ini memuat kelas-kelas aktif. Adapun simbol-simbol yang sering digunakan dalam *Class Diagram* adalah :



Tabel 2.4 Simbol-simbol pada *Class Diagram*





No.	Simbol	Keterangan
1.		Simbol <i>Class</i> , menggambarkan aktor pada diagram.
2.		Simbol <i>Agregation</i> , menggambarkan relasi agregasi
3.		Simbol <i>Association</i> , menggambarkan relasi asosiasi.

2.5.4 Sequence Diagram

Diagram yang bersifat dinamis, diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu. Adapun simbol-simbol yang sering digunakan dalam *Sequence Diagram* adalah :

Tabel 2.5 Simbol-simbol pada *Sequence Diagram*





No.	Simbol	Keterangan
1.		Simbol <i>Actor</i> , menggambarkan aktor pada diagram.
2.		Simbol <i>Boundary</i> , menggambarkan batasan kelas pada diagram.



3.		Simbol <i>Control</i> , menggambarkan unsure kendali pada diagram.
4.		<i>Entity</i> menggambarkan kelas entitas pada diagram.
5.		<i>Object Message</i> , menggambarkan pesan antar dua objek.
6.		<i>Message to Self</i> , menggambarkan pesan yang menuju dirinya sendiri.

2.5.5 Collaboration Diagram

Diagram yang bersifat dinamis, diagram kolaborasi adalah diagram interaksi yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan. Adapun simbol-simbol yang sering digunakan dalam *Collaboration Diagram* adalah :

Tabel 2.6 Simbol-simbol pada *Collaboration Diagram*





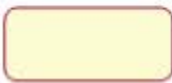
No.	Simbol	Keterangan
1.		Simbol <i>Actor</i> , menggambarkan aktor pada diagram.
2.		Simbol <i>Boundary</i> , menggambarkan batasan kelas pada diagram.
3.		Simbol <i>Control</i> , menggambarkan unsure kendali pada diagram.
4.		<i>Entity</i> menggambarkan kelas entitas pada diagram.
5.		<i>Link to Self</i> , menggambarkan bahwa suatu objek memanggil operasinya sendiri
6.		<i>Object Link</i> , menggambarkan lintasan komunikasi antar dua objek.

7.		<i>Link Message</i> , menggambarkan pesan antar dua objek, atau dari suatu objek ke dirinya sendiri.
8.		<i>Reverse Link Message</i> , menggambarkan pesan dalam arah berlawanan antar dua objek, atau dari suatu objek ke dirinya sendiri.

2.5.6 *Activity Diagram*

Diagram yang bersifat dinamis, diagram aktivitas adalah tipe khusus dalam diagram *state* yang memperlihatkan aliran dari sesuatu aktifitas ke aktifitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi dalam suatu sistem dan memberi gambaran alirannya antar objek. Adapun simbol-simbol yang sering digunakan dalam *Activity Diagram* adalah :

Tabel 2.7 Simbol-simbol pada *Activity Diagram*





No.	Simbol	Keterangan
1.		Simbol <i>Start state</i> , menggambarkan aliran kerja berawal.
2.		Simbol <i>End state</i> , menggambarkan aliran kerja berakhir.
3.		Simbol <i>Decision</i> , menggambarkan Titik keputusan pada aliran kerja.
4.		<i>State Transition</i> , menggambarkan transisi dari suatu aktivitas ke aktivitas yang lain.
5.		<i>State</i> , menggambarkan state untuk suatu objek.

2.5.7 *Statechart Diagram*

Diagram yang bersifat dinamis, diagram ini memperlihatkan *state*, transisi event, serta aktifitas. Diagram ini penting terutama untuk memperlihatkan sifat

dinamis dari antar muka (*interface*), kelas, kolaborasi, dan terutama penting pada pemodelan sistem-sistem yang reaktif. Adapun simbol-simbol yang sering digunakan dalam Statechart Diagram adalah :

Tabel 2.8 Simbol-simbol pada Statechart Diagram

No.	Simbol	Keterangan
1.		Simbol <i>Start state</i> , menggambarkan <i>state</i> awal. Pada diagram.
2.		Simbol <i>End state</i> , menggambarkan <i>state</i> berakhir pada diagram.
3.		Simbol <i>Transition to Self</i> , menggambarkan transisi yang mengarah pada <i>state</i> tunggal.
4.		<i>State Transition</i> , menggambarkan transisi pada diagram.

2.6 Implementasi

2.6.1 PHP

PHP adalah kependekan dari *Hypertext Preprocessor* atau Profesional Homepage, yaitu sebuah bahasa scripting yang dieksekusi di sisi server (*Server-side Scripting Language*). Fungsinya adalah membuat sebuah web yang interaktif dan dinamis. PHP dibuat pertama kali pada tahun 1995 oleh Rasmus Lerdorf, seorang software engineer anggota tim pengembangan web server Apache. Pada tahun 1996 Rasmus menulis kode script Perl untuk diterapkan pada hal yang lebih kompleks dengan bahasa pemrograman C.

Pada bulan Oktober 2000 PHP merilis versi 4.0 dengan mengintegrasikan Zend Engine, dan merupakan versi pengembangan dari PHP 3 Script Engine.

Keunggulannya dari sifatnya yang server-side tersebut antara lain :

- Tidak diperlukan kompatibilitas browser atau harus menggunakan browser tertentu, karena server-lah yang akan mengerjakan skrip PHP. Hasil yang dikirimkan kembali ke browser apapun.
- Dapat memanfaatkan sumber-sumber aplikasi yang dimiliki oleh server, misalnya koneksi ke database.

- Skrip tidak dapat “diintip” dengan menggunakan fasilitas view HTML source.

Sebagai contoh, perhatikan skrip PHP sederhana berikut ini:

Contoh:

```
Skrip PHP </TITLE>
<?
Echo "<H1> Hallo, saya skrip PHP</H1>";
?>
```

Setelah skrip ini dikerjakan di server, server akan memberikan output kepada browser sebagai berikut :

Contoh :

```
Skrip PHP </Title>
Hallo, saya skrip PHP </H1>
```

Output inilah yang dieksekusi oleh browser. Dengan output berupa tag-tag HTML biasa, maka output tersebut dapat dieksekusi oleh browser apapun.

Adapun kelebihan dari PHP yaitu dapat “melakukan” semua aplikasi program CGI, seperti menginput form, menghasilkan halaman web yang dinamis, mengirim dan menerima cookie. PHP juga dapat berkomunikasi dengan layanan-layanan yang menggunakan protokol IMAP, SNMP, NNTP, POP3, HTTP, dan lain-lain.

Namun tampaknya kelebihan PHP yang paling signifikan adalah kemampuannya untuk melakukan koneksi dengan berbagai macam database. Saat ini, database yang didukung PHP adalah : Adabas D, InterBase, PostgreSQL, dBase, FrontBase, Solid, Empress, mSQL, Sybase, FilePro(read-only), Direct MS-SQL, Velocis, IBM DB2, MySQL, Unix dbm, Informix, Semua database yang mempunyai provider ODBC, Ingres, Oracle (OCI7 and OCI8).

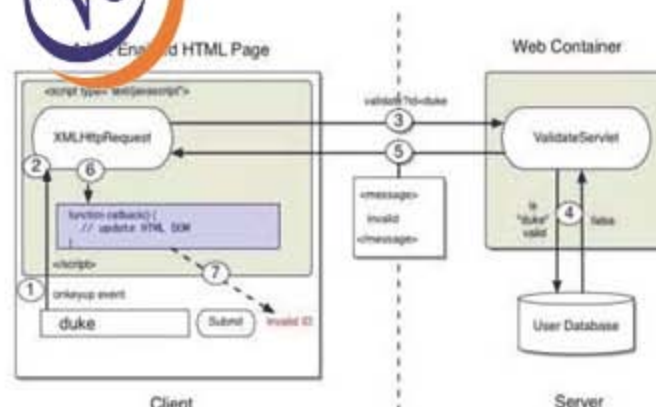
2.6.2 AJAX

AJAX itu sendiri diperkenalkan oleh seorang pakar pembuat program khususnya pemrograman berbasis web. Orang yang sangat berjasa tersebut tidak

lain dan tidak bukan adalah Rasmus Lerdorf, dari tangan dingin Rasmus Lerdorf ini pulalah beberapa waktu yang lalu lahirlah PHP.

Sebenarnya tidak ada hal baru dalam AJAX, karena yang digunakan adalah teknologi javascript, yang notabene sudah lama digunakan. Oleh karena itu, bagi yang sudah terbiasa dengan javascript, akan mudah sekali mempelajari AJAX. Pembahasan ini tidak menekankan pada salah satu teknologi (Javascript, AJAX, HTML, PHP), akan tetapi keseluruhan dari hal tersebut. Hal ini karena penggunaan AJAX tidak hanya pada sisi client (browser), tapi juga melibatkan respon dari server. Dengan menggunakan konsep ini, page HTML bisa membuat koneksi secara asinkron ke server dengan cara mengambil XML atau dokumen text. Selanjutnya XML atau dokumen text ini akan digunakan oleh javascript untuk meng-update atau memodifikasi Document Object Model (DOM) pada page HTML. Untuk selanjutnya teknologi ini juga dikenal web remoting atau remote scripting.

Developer Web dapat mengkombinasikan plug in, java applets atau hidden frame untuk mengemulasikan interaksi antara page HTML dengan dokumen XML. Dalam teknologi Javascript telah disediakan sebuah object yaitu XMLHttpRequest. Object ini sudah banyak disupport oleh banyak browser seperti Internet Explorer, Firefox, Opera, Netscape dan lain sebagainya. Interaksi AJAX dengan Server dapat dilihat pada gambar 2.3



Gambar 2.3 Interaksi AJAX dengan Server

Konsep AJAX ini ide awalnya dikemukakan oleh Rasmus Lerdorf. Rasmus Lerdorf mengemukakan konsep AJAX ini dalam sebuah miling list (milist). Berawal dari milist tersebut konsep AJAX kemudian dikenal dan

dikembangkan oleh banyak orang. Berikut ini isi milist yang dikirimkan oleh Rasmus Lerdorf.

List: php-general
 Subject: [PHP] Rasmus' 30 second AJAX Tutorial - [was Re: [PHP] AJAX & PHP]
 From: Rasmus Lerdorf <rasmus () lerdorf ! com>
 Date: 2005-07-21 22:50:56
 Message-ID: 42E026D0.3090601 () lerdorf ! com

I find a lot of this AJAX stuff a bit of a hype. Lots of people have been using similar things long before it became "AJAX". And it really isn't as complicated as a lot of people make it out to be. Here is a simple example from one of my apps. First the Javascript:

```
function createRequestObject() {
    var ro;
    var browser = navigator.appName;
    if(browser == "Microsoft Internet Explorer"){
        ro = new ActiveXObject("Microsoft.XMLHTTP");
    }else{
        ro = new XMLHttpRequest();
    }
    return ro;
}

var http = createRequestObject();

function send(action) {
    http.open('get', 'rpc.php?action='+action);
    http.onreadystatechange = handleResponse;
    http.send(null);
}

function handleResponse() {
    if(http.readyState == 4){
        var response = http.responseText;
        var update = new Array();

        if(response.indexOf('|' != -1)) {
            update = response.split('|');
        }

        document.getElementById(update[0]).innerHTML =
        update[1];
    }
}
}
```

This creates a request object along with a send request and handle response function. So to actually use it, you could include this js in your page. Then to make one of these backend requests you would tie it to something. Like an onclick event or a straight href like this:

```
<a href="javascript:sndReq('foo')">[foo]</a>
```

That means that when someone clicks on that link what actually happens is that a backend request to `rpc.php?action=foo` will be sent.

In `rpc.php` you might have something like this:

```
switch($_REQUEST['action']) {
    case 'foo':
        / do something /
        echo "foo|foo done";
        break;
    ...
}
```

Now, look at `handleResponse`. It parses the `foo|foo done` string and splits it on the `|` and uses whatever is before the `|` as the `dom` element id in your page and the part after as the new innerHTML of that element. That means if you have a `div` tag like this in your page:

```
<div id="foo">
</div>
```

Once you click on that link that will dynamically be changed to:

```
<div id="foo">
foo done
</div>
```

That's all there is to it. Everything else is just building on top of this. Replacing my simple response `"id|text"` syntax with a richer XML format and make the request much more complicated as well. Before you blindly install large "AJAX" libraries, have a go at rolling your own functionality so you know exactly how it works and you only make it as complicated as you need. Often you don't need much more than what I have shown here.

Expanding this approach a bit to send multiple parameters in the request, for example, would be really simple. Something like:

```
function sndReqArg(action, arg) {
    http.open('get',
'rpc.php?action='+action+'&arg='+arg);
    http.onreadystatechange = handleResponse;
```

```
    http.send(null);  
}
```

And your `handleResponse` can easily be expanded to do much more interesting things than just replacing the contents of a `div`.

-Rasmus

2.6.3 MySQL

MySQL dikembangkan oleh sebuah perusahaan Swedia bernama MySQL AB, yang pada saat itu bernama TcX Data Konsult AB, sejak sekitar 1994 – 1995. MySQL versi 1.0 dirilis Mei 1996 secara terbatas kepada empat orang. Baru di bulan Oktober versi 3.11.0 dilepas ke publik. Versi pertama ini hanya berjalan di Linux dan Solaris serta sebagian besar masih belum terdokumentasi itu berangsur-angsur diperbaiki dan ditambah fitur demi fiturnya.

Barulah di versi-versi akhir 3.22 – sepanjang 1998 – 1999 MySQL menjadi semakin populer dan dilirik orang. Kalau di versi ini MySQL mulai diadopsi banyak orang dan meningkat jumlah penggunaannya, maka di versi 5.0.41-lah terjadi banyak peningkatan dari sisi teknologi.

