

BAB II

TINJAUAN PUSTAKA

II.1 *Elearning*

E-Learning adalah pembelajaran jarak jauh (*distance Learning*) yang memanfaatkan teknologi komputer, jaringan komputer dan/atau Internet. *E-Learning* memungkinkan pembelajar untuk belajar melalui komputer di tempat mereka masing-masing tanpa harus secara fisik pergi mengikuti pelajaran/perkuliah di kelas. *E-Learning* sering pula dipahami sebagai suatu bentuk pembelajaran berbasis *web* yang bisa diakses dari intranet di jaringan lokal atau internet. Sebenarnya materi *E-Learning* tidak harus didistribusikan secara *on-line* baik melalui jaringan lokal maupun internet, distribusi secara *off-line* menggunakan media *CD/DVD* pun termasuk pola *E-Learning*. Dalam hal ini aplikasi dan materi belajar dikembangkan sesuai kebutuhan dan didistribusikan melalui media *CD/DVD*, selanjutnya pembelajar dapat memanfaatkan *CD/DVD* tersebut dan belajar di tempat di mana dia berada.

Ada beberapa pengertian berkaitan dengan *E-Learning* sebagai berikut :

1. Pembelajaran jarak jauh.

E-Learning memungkinkan pembelajar untuk menimba ilmu tanpa harus secara fisik menghadiri kelas. Pembelajar bisa berada di Semarang, sementara “instruktur” dan pelajaran yang diikuti berada di tempat lain, di kota lain bahkan di negara lain. Interaksi bisa dijalankan secara *on-line* dan *real-time* ataupun secara *off-line* atau *archieved*.

Pembelajar belajar dari komputer di kantor ataupun di rumah dengan memanfaatkan koneksi jaringan lokal ataupun jaringan Internet ataupun menggunakan media *CD/DVD* yang telah disiapkan. Materi belajar dikelola oleh sebuah pusat penyedia materi di kampus/universitas, atau perusahaan penyedia *content* tertentu. Pembelajar bisa mengatur sendiri waktu belajar, dan tempat dari mana ia mengakses pelajaran.

2. Pembelajaran dengan perangkat computer

E-Learning disampaikan dengan memanfaatkan perangkat komputer. Pada umumnya perangkat dilengkapi perangkat *multimedia*, dengan *CD Drive* dan koneksi Internet ataupun Intranet lokal. Dengan memiliki komputer yang terkoneksi dengan *Intranet* ataupun Internet, pembelajar dapat berpartisipasi dalam *E-Learning*. Jumlah pembelajar yang bisa ikut berpartisipasi tidak dibatasi dengan kapasitas kelas. Materi pelajaran dapat diketengahkan dengan kualitas yang lebih standar dibandingkan kelas konvensional yang tergantung pada kondisi dari pengajar.

3. Pembelajaran formal vs informal

E-Learning bisa mencakup pembelajaran secara formal maupun informal. *E-Learning* secara formal, misalnya adalah pembelajaran dengan kurikulum, silabus, mata pelajaran dan tes yang telah diatur dan disusun berdasarkan jadwal yang telah disepakati pihak-pihak terkait (pengelola *E-Learning* dan pembelajar sendiri). Pembelajaran seperti ini biasanya tingkat interaksinya tinggi dan diwajibkan oleh perusahaan pada karyawannya, atau pembelajaran jarak jauh yang dikelola oleh universitas dan perusahaan-perusahaan (biasanya perusahaan

konsultan) yang memang bergerak di bidang penyediaan jasa *E-Learning* untuk umum. *E-Learning* bisa juga dilakukan secara informal dengan interaksi yang lebih sederhana, misalnya melalui sarana *mailing list*, *e-newsletter* atau *website* pribadi, organisasi dan perusahaan yang ingin mensosialisasikan jasa, program, pengetahuan atau keterampilan tertentu pada masyarakat luas (biasanya tanpa memungut biaya).

4. Pembelajaran yang ditunjang oleh para ahli di bidang masing-masing.

Walaupun sepertinya *E-Learning* diberikan hanya melalui perangkat komputer, *E-Learning* ternyata disiapkan, ditunjang, dikelola oleh tim yang terdiri dari para ahli di bidang masing-masing, yaitu:

- a. *Subject Matter Expert (SME)* atau nara sumber dari pelatihan yang disampaikan
- b. *Instructional Designer (ID)*, bertugas untuk secara sistematis *mendesain* materi dari *SME* menjadi materi *E-Learning* dengan memasukkan unsur metode pengajaran agar materi menjadi lebih interaktif, lebih mudah dan lebih menarik untuk dipelajari.
- c. *Graphic Designer (GD)*, mengubah materi *text* menjadi bentuk grafis dengan gambar, warna, dan *layout* yang enak dipandang, efektif dan menarik untuk dipelajari
- d. Ahli bidang *Learning Management System (LMS)*. Mengelola sistem di *website* yang mengatur lalu lintas interaksi antara instruktur dengan siswa, antar siswa dengan siswa lainnya.

II.2 Aplikasi

Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media.

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau *suite* aplikasi (*application suite*). Contohnya adalah *Microsoft Office* dan *Open Office.org*, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi. Sering kali, mereka memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna. Contohnya, suatu lembar kerja dapat dibenamkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.

II.3 Pengertian Komputer

Komputer berasal dari bahasa latin *computare* yang mengandung arti menghitung. Karena luasnya bidang ilmu komputer, para pakar dan peneliti sedikit berbeda dalam mendefenisikan terminologi komputer.

Menurut Janner Simarmata (2010:30), Komputer adalah mesin penghitung elektronik yang cepat dan tepat menerima informasi input digital, kemudian memprosesnya sesuai dengan program yang tersimpan di memorinya dan menghasilkan berupa informasi (Janner Simarmata, Pengenalan Teknologi Komputer dan Informasi, Penerbit Andi, 2010).

Dari berbagai pendapat diatas, maka dapat disimpulkan secara umum bahwa komputer merupakan suatu peralatan elektronik yang dapat menerima input, mengolah input dengan menggunakan suatu program yang tersimpan di memori komputer, memberikan output yang berupa informasi dan dapat menyimpan program serta hasil pengolahan dalam suatu media penyimpanan ataupun hasil proses tersebut bisa dicetak dalam media kertas.

II.4 Multimedia

Multimedia berasal dari kata “multi” dan “media”. Multi berarti banyak dan media adalah merupakan sarana untuk penerapan. Multimedia dapat diartikan sebagai penggunaan beberapa media yang berbeda untuk menggabungkan dan menyampaikan informasi dalam bentuk teks, gambar/grafik, animasi, *audio* dan *video* (dapat disebut juga gabungan dari berbagai media yang terintegrasi). Beberapa definisi multimedia antara lain:

1. Kombinasi dari komputer dan *video*
2. Kombinasi dari tiga elemen: suara, gambar, dan teks .
3. Kombinasi dari paling sedikit dua media *input* atau *output*. Media ini dapat berupa *audio* (suara, musik), animasi, *video*, teks, grafik dan gambar.

4. Alat yang dapat menciptakan presentasi yang dinamis dan interaktif yang mengkombinasikan teks, grafik, animasi, *audio* dan *video*.
5. Multimedia dalam konteks komputer adalah pemanfaatan komputer untuk menggabungkan teks, grafik, *audio*, *video*, dengan menggunakan *tool* yang memungkinkan pemakai berinteraksi, berkreasi dan berkomunikasi.
(Sumber : TriDaryanto, 2005: 120)

Adapun beberapa karakteristik multimedia adalah sebagai berikut :

1. Bersifat *fleksibel* (memberikan keleluasaan bagi *user* untuk memilih materi dan menggunakannya)
2. Bersifat *self-pacing* (memberikan kesempatan user untuk belajar sesuai dengan kecepatannya dalam memahami materi)
3. Bersifat *content-rich* (memberikan informasi yang kaya baik dari isi maupun medianya)
4. Bersifat interaktif (memberikan kesempatan bagi *user* untuk melakukan respon dan mencoba simulasi).

II.5 Gambar

Gambar merupakan suatu representasi *spatial* dari suatu obyek, dalam pandangan 2D atau 3D. Gambar digital merupakan suatu fungsi dengan nilai-nilai yang berupa intensitas cahaya pada tiap-tiap titik pada bidang yang telah diquantisasikan (diambil sampelnya pada interval diskrit).

Format file gambar yang ada saat ini antara lain : Bitmap (.BMP), Joint Photographic Expert Group (.JPEG/JPG), Graphics Interchange Format (.GIF), Portable Network Graphics (.PNG), Tagged Image File Format (TIFF), Icon (ICO), Enhanced Windows Metafile (EMF), PCX, ANI (Animation), Cursor (CUR), WBMP (WAP BMP), Adobe Photoshop Document (PSD), dan Corel Draw (CDR). (Sumber : Daryanto, 2005).

II.6 Animasi

Animasi biasa disebut dengan gambar bergerak. Animasi adalah “*illusion of motion*” yang dibuat dari *image* statis yang ditampilkan secara berurutan. Pada *video* atau *film*, animasi merancu pada teknik dimana setiap *frame* dalam *film* dibuat secara terpisah. *Frame* bisa dihasilkan dari komputer, dari fotografi atau dari gambar lukisan. Ketika *frame-frame* tersebut digabungkan maka terdapat ilusi perubahan gambar, sesuai dengan teori yang disebut dengan “*persistance of vision*”.

Jenis-jenis animasi menurut (Daryanto,2005:190) yaitu :

1. Animasi *Cel*, animasi *cel* biasanya merupakan lembaran-lembaran yang membentuk animasi tunggal, masing-masing *cel* merupakan bagian yang terpisah, misalnya antara objek dengan latar belakangnya, sehingga dapat saling bergerak mandiri.
2. Animasi *Frame* adalah bentuk animasi paling sederhana, contohnya ketika kita membuat gambar-gambar yang berbeda-beda gerakannya. Dalam sebuah

film, serangkaian *frame* bergerak dengan kecepatan minimal 24 *frame* per detik.

3. Animasi *Sprite*, adalah animasi gambar yang digerakkan dengan latar belakang yang diam.
4. Animasi *Path*, adalah animasi dari objek yang bergerak sepanjang garis kurva yang ditentukan sebagai lintasan.
5. Animasi *Spline*, adalah representasi matematis dari kurva, sehingga gerakan dari objek tidak hanya mengikuti garis lurus melainkan berbentuk kurva.
6. Animasi *Vektor*, adalah garis yang memiliki ujung pangkal, arah dan panjang.
7. Animasi *Character*, animasi karakter biasanya terdapat di *film* kartun, semua bagian dalam *film* kartun selalu bergerak bersamaan.

II.7 Suara

Suara adalah fenomena fisik yang dihasilkan oleh getaran benda, getaran suatu benda yang berupa sinyal analog dengan amplitudo yang berubah secara kontinyu terhadap waktu. Konsep dasar suara dihasilkan oleh getaran suatu benda. Selama bergetar perbedaan tekanan terjadi di udara sekitarnya ini yang disebut dengan Gelombang. Gelombang mempunyai pola sama yang berulang pada interval tertentu, yang disebut sebagai “Periode”.

Suara yang berada pada batas pendengaran manusia sebagai “*audio*”, dan gelombangnya sebagai “*acoustic signals*”. Suara di luar batas pendengaran manusia dapat dikatakan sebagai “*noise*” (getaran yang tidak

teratur dan tidak berurutan dalam berbagai frekuensi, tidak dapat di dengar manusia).

Format audio yang ada saat ini antara lain : AAC (*Advanced Audio Coding*) [.m4a], *WAVEFORM AUDIO* [.WAV], *Audio Interchange File Format* [.AIF], *Audio CD* [.cda], *Mpeg Audio Layer 3* [.mp3], *MIDI (Music Instrument Digital Interface)* dan sebagainya. (Daryanto, 2005:125).

II.8 Video

Video adalah teknologi untuk menangkap, merekam, memproses, mentransmisikan dan menata ulang gambar bergerak. Biasanya menggunakan *film seluloid*, sinyal elektronik atau media digital. Aplikasi *video* pada multimedia mencakup banyak aplikasi antara lain :

1. *Entertainment: roadcast TV, VCR/DVD recording*
2. *Interpersonal: video telephony, video conferencing*
3. *Interactive : Windows*

Video digital memiliki keuntungan antara lain :

1. Interaktif, *Video* digital disimpan dalam media penyimpanan *random* contohnya *magnetic/optical disk*. Sedangkan *video* analog menggunakan tempat penyimpanan sekuensial, contohnya *magnetic disc/kaset video*. *Video* digital dapat memberikan respon waktu yang cepat dalam mengakses bagian manapun dari *video*.
2. Mudah dalam proses *edit*, apabila user ingin melakukan penambahan atau pengurangan pada *file* dapat dilakukan dengan mudah.

3. Kualitas, sinyal analog dari *video* analog akan mengalami penurunan kualitas secara perlahan karena adanya pengaruh kondisi atmosfer. Sedangkan *video* digital kualitasnya dapat diturunkan menggunakan teknik kompresi.
4. Transmisi dan distribusi mudah karena dengan proses kompresi, maka *video* digital dapat disimpan dalam CD, ditampilkan pada *web*, dan ditransmisikan melalui jaringan.

Beberapa format *video* yang ada saat ini antara lain : *Motion Picture Expert Group* (MPEG-1, MPEG-2, MPEG-4), ASF (*Advanced System Format*), MOV (*Quick Time*), AVI (*Audio Video Interleaved*), 3Gp (*3rd Generation Partnership Project*), *Windows Media Video* (WMV).

II.9 Tutorial

Tutorial adalah merupakan panduan tentang bagaimana mengoperasikan suatu sistem, baik *hardware* maupun *software*. Pada tutorial ini biasanya dilengkapi informasi tentang objeknya, dan petunjuk langkah demi langkah untuk suatu operasi yang sifatnya mengajarkan tentang sesuatu. (Sumber : <http://www.total.or.id/>)

Suatu tutorial dapat dibuat dengan banyak cara, tetapi cara yang paling banyak dibuat dan paling efisien sekarang ini adalah dengan menggunakan media *video* atau animasi, dengan menggunakan media tersebut proses pembelajaran (tutorial) akan lebih cepat dimengerti dan mudah dipahami oleh *user*.

Proses pembelajaran dengan sistem multimedia akan lebih cepat terasa bagi *user* yang ingin belajar secara cepat, karena sistem pembelajaran ini lebih cepat diterima dan lebih mudah dipahami karena menggunakan audio visual yang bisa dimengerti oleh *user*.

II.10 *Compact Disc Interaktif Multimedia*

Compact Disk yang disingkat dengan CD adalah merupakan media penyimpanan yang berbentuk *optical disk*. CD memiliki karakteristik *Non-volatile* yaitu *memory* yang dapat tetap menyimpan informasi meskipun tidak terdapat sumber daya. CD juga merupakan *secondary storage* (memerlukan *channel input / output*), *tertiary* (sistem yang akan menangani *offline storage* berdasarkan perintah komputer), dan *offline storage* (mudah dipindahkan dari *storage device*). Disebut *Compact Disk Multimedia* karena didalamnya berisi materi multimedia (teks, gambar, grafis, *video*, audio dan interaktif).

CD Multimedia disebut interaktif karena memiliki :

1. Terdapat interaktifitas dua arah (antara *user* dan komputer).
2. Komputer dapat memberi respon dan *feedback* kepada pengguna.
3. Penggunaan tombol navigasi pada presentasi.

Kegunaan dari CD Interaktif Multimedia adalah :

1. Demonstrasi, bersifat mempresentasikan sesuatu, misal untuk pengenalan produk baru.
2. Tutorial, bersifat mengajarkan tentang sesuatu, misalnya CD Tutorial

Pembelajaran *Microsoft Word 2003*.

3. Simulasi, bersifat Menggambarkan fungsi riil dari *system*, terjadi interaktifitas dua arah, misalnya simulasi merubah pengaturan dalam *browser Internet Explorer*.
4. Evaluasi, Mengukur pencapaian materi, misalnya *quiz multiple choice* di setiap akhir penyampaian materi ajar.

Kombinasi berbagai media dimanfaatkan secara harmonis dan terintegrasi sehingga menghasilkan satu program multimedia yang sangat berguna.

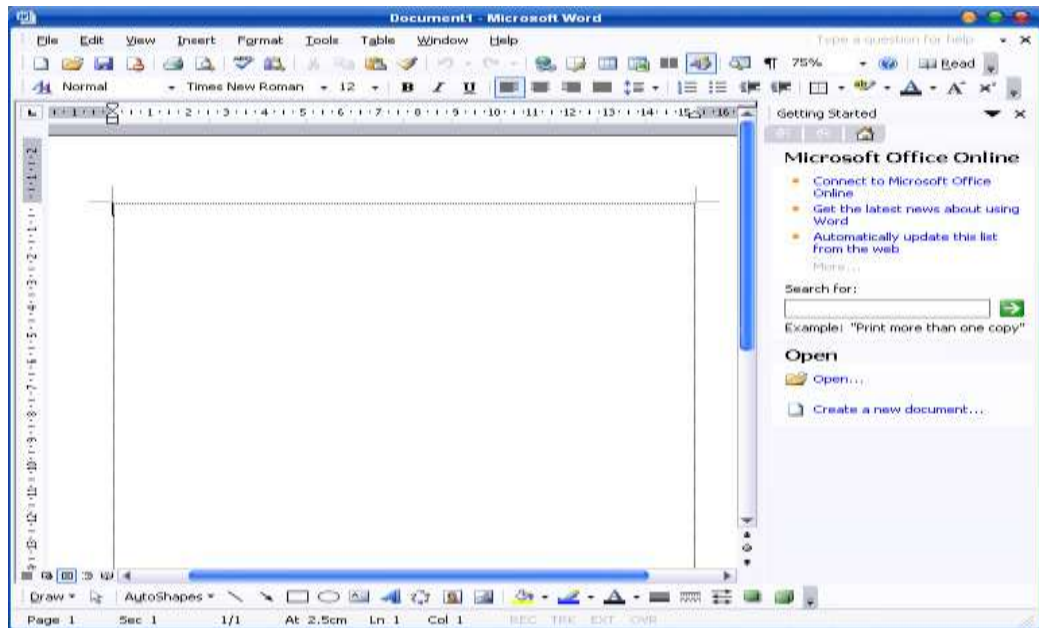
II.11 *Microsoft Word 2003*

Microsoft Word 2003 merupakan versi *update* dari *Microsoft Word* yang telah sukses sebelumnya (*Microsoft Word 2002*, *Word 2000* dan sebagainya). *Microsoft Word* adalah program aplikasi pengolahan kata yang diluncurkan oleh perusahaan *Microsoft* di *Amerika Serikat* sekitar tahun 1990 an.

Sebenarnya *Microsoft Word 2003* tidak berbeda jauh dengan *Microsoft Word* versi sebelumnya, sehingga jika anda sudah mampu menggunakan *Microsoft Word* versi sebelumnya maka versi ini lebih mudah dipelajari,

Penggunaan program aplikasi ini sangat banyak ditemukan dan digunakan baik di perusahaan, Perguruan Tinggi, sekolah–sekolah maupun di kalangan pribadi. Pada program ini terdapat berbagai fasilitas yang sangat

mudah untuk dipelajari dengan *interface* atau tampilan yang baik. Pembuatan suatu tulisan, makalah atau surat menyurat akan terasa lebih mudah dan dapat dikerjakan dengan waktu yang relatif singkat.



Gambar II.1 Tampilan *Microsoft Office 2003*

Sumber : Agus, 2004

Untuk dapat menjalankan/mengoperasikan program *Microsoft Word 2003*, komputer sebaiknya memiliki spesifikasi sebagai berikut :

Tabel II.1 Spesifikasi *Hardware* Untuk *Microsoft Word 2003*

Hardware	Minimal	Dianjurkan
Processor	Pentium 233 Mhz	Pentium III
Memori	64 MB	128 MB
Monitor	SVGA (800x600) 256 Warna	Resolusi Lebih Tinggi
Hard Disk	245 MB	2 GB
Mouse / Keyboard	Tipe Standard	Microsoft
Sistem Operasi	Windows 2000 SP3	Windows XP

Sumber : Agus, 2004

II.12 *Camtasia Studio*

Sekarang telah banyak bermunculan aplikasi atau *software* yang menawarkan kemampuan untuk merekam kegiatan di komputer sehingga dapat digunakan sebagai CD tutorial. Satu di antaranya yang paling baik adalah *Camtasia Studio*. Menggunakan *Camtasia*, seseorang dapat menerangkan ilmu pengetahuan secara audio visual kepada calon peserta didik. Tentu saja cara penyampaian seperti itu semakin memperjelas pengertian kepada orang-orang terhadap materi tersebut.

Aplikasi keluaran *TechSmith Corporation* ini diluncurkan sekitar tahun 1999. Munculnya aplikasi ini mendapat sambutan yang baik, karena mendukung berbagai macam format seperti *AVI*, *MWV*, *Mp3*, *MOV*, *Real Media* dan *GIF*.

Camtasia Studio merupakan *software* yang paling banyak digunakan untuk membuat *video* tutorial dikarenakan kemudahan dalam mengoperasikannya dan mempunyai banyak format *video* yang didukungnya.



Gambar II.2 Tampilan *Camtasia Studio* 5.0

Sumber : Firmansyah, 2008

Aplikasi ini dapat merekam segala aktifitas yang terdapat dalam monitor komputer, mulai dari gerakan *mouse*, membuka aplikasi, sampai dengan pemutaran *video* dan dapat melakukan perekaman suara melalui *microphone* yang dihubungkan ke komputer. Setelah melakukan perekaman, *file* dapat dikonfersikan kedalam bentuk *AVI*, *WMV* atau *SWF*.

Camtasia Studio merupakan aplikasi *recording all in one*, dimana proses perekaman sudah dilakukan maka hasil perekaman dapat di *edit* dengan memberikan efek animasi sederhana ataupun suara latar, dan setelah dilakukan proses *editing video* perekaman bisa disimpan dalam format apa saja, untuk aplikasi yang dirancang penulis maka hasilnya akan disimpan dalam format *SWF*.

II.13 Aspek Pembelajaran

Dalam pembuatan tutorial pembelajaran *Microsoft Word 2003* ini ada beberapa pembelajaran yang dilakukan sebagai dasar pengenalan *Microsoft Word 2003* kepada pengguna awam yang baru belajar *Microsoft Word 2003*, antara lain sebagai berikut :

1. Mengetahui program *Microsoft Word 2003* dan mengelola *file*
2. Membuat dan menyimpan dokumen
3. Mengelola dokumen pada *Microsoft Word 2003*
4. Format karakter dan paragraf
5. Format *Bullet* dan *number list*
6. Mengganti margin dan nomor halaman

7. Menggunakan tabel dan grafik
8. Gambar dan *drawing object*
9. Mencetak dokumen

II.14 PHP

PHP merupakan script untuk pemrograman *script web server-side*, *script* yang membuat dokumen *HTML* secara *on the fly*, dokumen *HTML* yang dihasilkan dari suatu aplikasi bukan dokumen *HTML* yang dibuat dengan menggunakan editor teks atau editor *HTML*. Dengan menggunakan *PHP* maka *maintenance* atau situs *web* menjadi lebih mudah. Proses *update* data dapat dilakukan dengan menggunakan aplikasi yang dibuat dengan menggunakan *script PHP*.

PHP/FI merupakan nama awal dari *PHP*. *PHP* atau *Personal Home Page*, *FI* adalah *Form Interface*. *PHP* saat ini terdiri dari dua macam yaitu *PHP3* dan *PHP4*. *PHP3* adalah bahasa *PHP* yang pertama kali dibuat oleh sekelompok orang yang terdiri dari Rasmus Lerdoff, Andi Gutmans, Zeev Suraski, Stig Bakken, Shane Caraveo dan Jim Winstead. Selaku turunan dari bahasa *PHP/FI* dan *PHP2* yang dikembangkan secara signifikan, *PHP3* yang telah melalui berbagai penyempurnaan masih terus dikembangkan sampai sekarang. *PHP3* sangat cocok untuk web yang tidak terlalu kompleks dan besar. Secara umum, *PHP3* juga mendukung aplikasi *database* dan aplikasi pendukung lainnya. *PHP4* adalah bahasa *PHP* versi terbaru yang dilengkapi *zend engine* yaitu sebuah fasilitas yang memungkinkan proses pemrograman berlangsung lebih cepat, kuat,

stabil dan mudah untuk berinteraksi dengan berbagai aplikasi pendukung lainnya, seperti *MYSQL*, *JAVA*, *FTP Client*, *ODBC*, *GNU*, *WAP* dan lain-lain. *PHP4* mempunyai ekstensi *PHP* yang lebih sederhana dan mudah dipergunakan serta kompatibel bila berpasangan dengan *PHP3*.

Kelebihan lain dari *PHP* adalah kemampuannya untuk berjalan dalam *web server* dan *system* operasi yang berbeda. *PHP* dapat berjalan pada *system* operasi *UNIX*, *Windows 98*, *Windows NT* dan *Macintosh*.

PHP awalnya merupakan program *CGI* yang dikhususkan untuk menerima input melalui *form* yang ditampilkan dalam *browser web*. *Software* ini disebar dan dilisensikan sebagai perangkat lunak *open source*. *PHP* secara resmi merupakan kependekan dari *Hypertext Preprocessor* yang merupakan bahasa *script server side* yang disisipkan pada *HTML*.

Berikut adalah contoh yang umum digunakan untuk menjelaskan tentang *PHP* sebagai *script* yang disisipkan dalam dokumen *HTML* :

```
<html>
```

```
<head>
```

```
<title>Contoh</title>
```

```
</head>
```

```
<body>
```

```
<?php
```

```
Echo "Hai..saya dari script PHP ";
```

```
?>
```

```
</body>
```

</html>

Pemrograman tidak harus menuliskan semua dokumen *HTML* sebagai bagian dari keluaran dari *script PHP*, cukup menuliskan bagian mana saja yang berupa *tag HTML* dan bagian mana yang harus ditulis atau dihasilkan dari program *script PHP*. Kode / *Script PHP* diapit dengan menggunakan *tag* awal dan tag akhir yang khusus yang memungkinkan pemrograman untuk masuk dan keluar dari *mode script PHP*.

(Abdul Kadir, “Dasar Pemrograman *WEB* Dinamis Menggunakan *PHP*”, 2005, hal : 1).

II.14.1 Kemampuan *PHP*

PHP secara mendasar dapat mengerjakan semua yang dapat dikerjakan oleh program *CGI*, seperti mendapatkan data dari *form*, menghasilkan isi halaman *web* yang dinamis. Kemampuan *PHP* yang paling diandalkan dan signifikan adalah mendukung kepada banyak *database*. Membuat halaman *web* yang menggunakan data dari *database* dengan sangat mudah dapat dilakukan. Berikut adalah nama *database* yang didukung oleh *PHP* :

- 1 *DBM*
- 2 *dBase*
- 3 *Ingres*
- 4 *FilePro*
- 5 *Hyperwave*
- 6 *Interbase*
- 7 *Msql*

8 *Mysql*

9 *Oracle*

10 *Postgre Sql*

11 *Sybase*

(Abdul Kadir, “Dasar Pemrograman *WEB* Dinamis Menggunakan PHP”, 2005, hal : 6).

II.14.2 Bentuk *Script PHP*

Setiap program *PHP* disebut dengan *script*. *Script* berupa *file* teks yang dapat dibuat dengan menggunakan editor *file* teks biasa seperti *notepad*, *edit*, *vi* (dalam lingkungan *Unix/Linux*). Editor teks yang digunakan sebaiknya editor teks yang memungkinkan membuat program *PHP* lebih mudah.

Aturan-aturan yang berlaku dalam pemrograman *PHP* antara lain :

1 Case Sensitive

Script PHP menerapkan aturan *case sensitive* yakni adanya perbedaan penulisan dengan huruf besar dan huruf kecil. Fungsi-fungsi yang ada di dalam *PHP* tidak *case sensitive*, namun tidak demikian untuk variabelnya. Dalam penulisan variable setiap scriptnya harus mengikuti aturan penulisan yang telah ditentukan terutama dalam penulisan huruf besar dan huruf kecil.

2 Kode atau Tag

Kode atau *script PHP* diawali dengan tanda/tag `<?` Kemudian diakhiri dengan tanda/tag `?>`. *script PHP* diawali dan diakhiri dengan menggunakan tag khusus.

3 Perintah atau *Statement*

Setiap baris perintah/*statement* harus diakhiri dengan menggunakan tanda titik koma (;). Umumnya setiap *statement* dituliskan dalam satu baris.

4 *Variabel*

Variabel tempat penyimpanan data di dalam *PHP* diawali dengan karakter \$ diikuti dengan huruf sebagai karakter pertama setelah \$, kemudian kombinasi karakter dan angka. Tidak boleh ada spasi dan tanda baca dalam penamaanya kecuali karakter _ (garis bawah, *under score*).

Contoh nama variabel yang benar adalah :

- 1) \$namauser
- 2) \$password
- 3) \$kota2
- 4) \$tempat_lahir

Contoh nama variabel yang salah adalah :

- 1) \$nama user
- 2) \$password
- 3) \$kota 2
- 4) \$tempat lahir

5 Komentar

Komentar yang dimaksud dalam *script* adalah bagian dari *script PHP* yang tidak akan dieksekusi, karena merupakan catatan terhadap fungsi dari *script* atau menjelaskan maksud dari sebagian dari *script* yang dituliskan. Sebuah komentar biasanya diawali dengan /*, diikuti komentar dan diakhiri dengan */.

(Abdul Kadir, “Dasar Pemrograman *WEB* Dinamis Menggunakan PHP”, 2005, hal : 2).

II.14.3 Konsep Kerja *PHP*

Konsep kerja *HTML* diawali dengan permintaan suatu halaman *web* oleh *browser*. Berdasarkan alamat *URL* (*Uniform Resource Locator*) yang juga dikenal dengan alamat internet ini, *browser* mendapatkan alamat dari *web server*, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *web server*. Informasi yang disampaikan ke *web server* antara lain adalah nama *browser*, versinya, dan system operasinya.

Selanjutnya *web server* akan mencari berkas yang diminta dan memberikan isinya ke *browser*. *Browser* akan segera melakukan proses penerjemah kode *HTML* dan menampilkan ke layer pemakai.

(Abdul Kadir, “Dasar Pemrograman *WEB* Dinamis Menggunakan PHP”, 2005, hal : 3).

II.15 *Unified Modelling Language*

Unified Modeling Language (UML) merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek (Whitten L. Jeffery et al, 2004). Sementara menurut Henderi (2007) UML adalah sebuah bahasa pemodelan yang telah menjadi standar dalam industri *software* untuk visualisasi, merancang, dan mendokumentasikan sistem perangkat lunak. Bahasa Pemodelan UML lebih cocok untuk pembuatan perangkat lunak dalam bahasa pemrograman berorientasi

objek (C++, Java, VB.NET), namun demikian tetap dapat digunakan pada bahasa pemrograman prosedural (Djon Irwanto, Perancangan *Object Oriented Software* dengan UML, 2007).

UML merupakan pemodelan objek yang fokus pada pendefinisian struktur statis dan model sistem informasi yang dinamis daripada mendefinisikan data dan model proses yang tujuannya adalah pengembangan tradisional. UML menawarkan diagram yang dikelompokkan menjadi lima perspektif berbeda untuk memodelkan suatu sistem. Adapun diagram yang terdapat dalam uml adalah sebagai berikut (*Sumber: Djon Irwanto, Perancangan Object Oriented Software dengan UML, 2007*):

1. Model *Use Case* Diagram

Use Case Diagram secara grafis menggambarkan interaksi antara sistem, sistem eksternal, dan pengguna. Dengan kata lain *Use Case* diagram secara grafis mendeskripsikan siapa yang akan menggunakan sistem dan dalam cara apa pengguna (*user*) mengharapkan interaksi dengan sistem itu. *Use Case* secara naratif digunakan untuk secara tekstual menggambarkan sekuensi langkah-langkah dari setiap interaksi.

2. Diagram Struktur Statis

UML menawarkan dua diagram untuk memodelkan struktur statis sistem informasi, yaitu:

- a. *Class Diagram*

Menggambarkan struktur object sistem. Diagram ini menunjukkan class object yang menyusun sistem dan juga hubungan antara class object tersebut.

b. Object Diagram

Serupa dengan class diagram, tetapi object diagram memodelkan *instance object* aktual dengan menunjukkan nilai-nilai saat ini dari *atribut instance*. Object Diagram menyajikan “*snapshot/potret*” tentang objek sistem pada point waktu tertentu. Diagram ini tidak digunakan sesering Class Diagram, tetapi saat digunakan dapat membantu seorang *developer* memahami struktur sistem secara lebih baik.

3. Diagram Interaksi

Diagram interaksi memodelkan sebuah interaksi, terdiri dari satu set objek, hubungan-hubungannya, dan pesan yang terkirim di antara objek. Model diagram ini memodelkan *behavior* (kelakuan) sistem yang dinamis dan UML memiliki dua diagram untuk tujuan ini, yaitu:

a. Diagram rangkaian/*Sequence* Diagram

Secara grafis menggambarkan bagaimana objek berinteraksi dengan satu sama lain melalui pesan pada sekuensi sebuah *use case* atau operasi. Diagram ini mengilustrasikan bagaimana pesan terkirim dan diterima di antara objek dan dalam sekuensi atau *timing* apa.

b. Diagram kolaborasi/*Collaboration* Diagram

Serupa dengan diagram rangkaian/sekuensi, tetapi tidak fokus pada timing atau sekuensi pesan. Diagram ini justru menggambarkan interaksi (atau kolaborasi) antara objek dalam sebuah format jaringan. Diagram rangkaian maupun diagram kolaborasi merupakan isomorphic artinya kita dapat mengubah dari satu diagram ke diagram lain.

4. Diagram *State/State* Diagram

UML memiliki sebuah diagram untuk memodelkan *behavior* objek khusus yang kompleks (*statechart*) dan sebuah diagram untuk memodelkan *behavior* dari sebuah *use case* atau sebuah metode, yaitu:

a. *Diagram statechart*

Digunakan untuk memodelkan *behavior* objek khusus yang dinamis. Diagram ini mengilustrasikan siklus hidup objek-berbagai keadaan yang dapat diasumsikan oleh objek dan *event-event* (kejadian) yang menyebabkan objek beralih dari satu *state* ke *state* lain.

b. *Diagram aktivitas/Activity Diagram*

Secara grafis digunakan untuk menggambarkan rangkaian aliran aktivitas baik proses bisnis maupun *use case*. *Activity* diagram dapat juga digunakan untuk memodelkan *action* yang akan dilakukan saat sebuah operasi dieksekusi, dan memodelkan hasil dari *action* tersebut.

5. Diagram Implementasi

Diagram implementasi juga memodelkan struktur sistem informasi, yaitu:

a. *Diagram komponen/Component Diagram*

Digunakan untuk menggambarkan organisasi dan ketergantungan komponen-komponen *software* sistem. Komponen diagram dapat digunakan untuk menunjukkan bagaimana kode pemrograman dibagi menjadi modul-modul (atau komponen).

b. *Diagram penguraian/Deployment*

Digunakan untuk mendeskripsikan arsitektur fisik dalam istilah "node" untuk *hardware* dan *software* dalam sistem. Diagram ini menggambarkan konfigurasi komponen-komponen *software real-time*, prosesor, dan peralatan yang membentuk arsitektur sistem.

UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan bahasa pemodelan visual yang memungkinkan pengembang sistem membuat *blue print* atas visinya dalam bentuk yang baku. UML berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui sejumlah elemen grafis yang bisa dikombinasikan menjadi diagram. UML mempunyai banyak diagram yang dapat mengakomodasi berbagai sudut pandang dari suatu perangkat lunak yang akan dibangun. (Djon Irwanto, Perancangan *Object Oriented Software* dengan UML, 2007):

Diagram-diagram tersebut digunakan untuk :

1. Mengkomunikasikan ide
2. Melahirkan ide-ide baru dan peluang-peluang baru
3. Menguji ide dan membuat prediksi
4. Memahami struktur dan relasi-relasinya

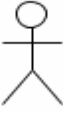
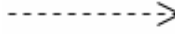


Berdasarkan uraian di atas maka penulis membuat sebuah alur sistem yang di tampilkan dalam bentuk *Use Case diagram*, *Activity diagram*, dan *Class diagram* dalam model *Unified Modelling Language* (UML).







II.15.1 Use Case diagram

Use Case diagram adalah model fungsional sebuah sistem yang menggunakan *actor* dan *use case*. *Use Case* adalah layanan (*services*) atau fungsi-fungsi yang disediakan oleh sistem untuk pengguna-penggunanya (Djon Irwanto, Perancangan *Object Oriented Software* dengan UML, 2007). *Use Case* adalah suatu pola atau gambaran yang menunjukkan kelakuan atau kebiasaan sistem. Setiap *Use Case* adalah suatu urutan (*sequence*) transaksi yang saling berhubungan dan dilakukan oleh sebuah *actor* dan sistem dalam bentuk sebuah dialog (Henderi, 2007). *Use Case diagram* dibuat untuk memvisualisasikan/menggambarkan hubungan antara *Actor* dan *Use Case*. *Use Case diagram* mempresentasikan kegunaan atau fungsi-fungsi sistem dari perspektif pengguna.

Simbol-simbol yang digunakan dalam *Use Case diagram* adalah sebagai berikut

Tabel II.2 Daftar simbol dalam *Use Case diagram*

GAMBAR	NAMA	KETERANGAN
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .

	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Sumber : Djon Irwanto, Perancangan *Object Oriented Software* dengan UML, 2007






II.15.2 Activity diagram

Activity diagram adalah diagram yang menggambarkan sifat dinamis secara alamiah sebuah sistem dalam bentuk model aliran dan kontrol dari aktivitas ke aktivitas lainnya (Sumber : Djon Irwanto, Perancangan *Object Oriented Software dengan UML*, 2007). Sebuah aktivitas merepresentasikan suatu operasi pada beberapa *class* dalam sistem yang menghasilkan suatu perubahan keadaan (*state*) dari sistem tersebut. Secara khusus, *activity diagram* biasa digunakan untuk memodelkan diagram alir sebuah sistem kerja (*workflow*) atau proses bisnis dan operasi-operasi secara internal (Miller Randy, 2008). Sementara menurut Whitten L. Jeffery et al (2004), *activity diagram* adalah sebuah diagram yang dapat

digunakan untuk menggambarkan secara grafis aliran proses bisnis, langkah-langkah sebuah *use case* atau logika *behavior* (metode) *object*. Karena sebuah *activity diagram* adalah bentuk khusus dari *statechart diagram*, maka *activity diagram* kadangkala digunakan untuk memodelkan suatu kebiasaan sesuai dengan ketentuan/kaedah bisnis.

Simbol-simbol yang digunakan dalam *Activity diagram* adalah sebagai berikut:

Tabel II.3 Daftar simbol dalam *Activity diagram*

GAMBAR	NAMA	KETERANGAN
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Action</i>	<i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi.
	<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
	<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan.
	<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.





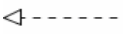
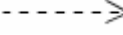

Sumber : Djon Irwanto, Perancangan *Object Oriented Software* dengan UML, 2007

II.15.3 *Class diagram*

Class adalah kumpulan objek-objek dengan dan yang mempunyai struktur umum, *behavior* umum, relasi umum, dan *semantic*/kata yang umum. *Class-class* ditentukan/ditemukan dengan cara memeriksa objek-objek dalam *sequence diagram* dan *collaboration diagram*. Sebuah *class* digambarkan seperti sebuah bujur sangkar dengan tiga bagian ruangan. *Class* sebaiknya diberi nama

menggunakan kata benda sesuai dengan domain/bagian/kelompoknya (Whitten L. Jeffery et al, 2004).

Tabel II.4 Daftar simbol dalam *Class* diagram

GAMBAR	NAMA	KETERANGAN
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>N-Ary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

Sumber : Djon Irwanto, Perancangan *Object Oriented Software* dengan *UML*, 2007