BAB II

LANDASAN TEORI

2.1 Pengertian Pembelajaran

Pembelajaran adalah suatu kombinasi tersusun yang meliputi unsur-unsur manusiawi, material, fasilitas perlengkapan, dan prosedur yang saling mempengaruhi mencapai tujuan pengajaran. Manusia yang terlibat dalam sistem pengajaran terdiri siswa atau mahasiswa, guru atau dosen, dan tenaga lainnya, misalnya tenaga laboratorium. Material, meliputi buku-buku, papan tulis, kapur, fotografi, slide, film, audio dan video tape. Fasilitas perlengkapan, terdiri dari ruang kelas, perlengkapan audio visual, *termasuk komputer*. Prosedur, meliputi jadwal dan metode penyimpanan informasi, praktek, belajar, dan ujian [OEM'99].

Rumusan tersebut tidak terbatas dalam ruang saja. Sistem pembelajaran dapat dilaksanakan dengan cara membaca buku, belajar di kelas atau di kampus, salah satunya dengan penggunaan fasilitas ruang praktikum sebagai tempat pembelajaran.

2.2 Penggunaan Komputer dalam Pengajaran

Di dalam dunia pendidikan berbasis komputer sudah dikenal **Pengajaran dengan Bantuan Komputer**. Pengajaran dengan bantuan komputer ini mempunyai latar belakang pengembangan yang panjang. Dimulai pada tahun 1954, pada tahun ini dikeluarkan artikel oleh *Harvard Educational Review* yang

kemudian melahirkan konsep baru di bidang pendidikan. Artikel tersebut berjudul **The Science of Learning and the Art of Teaching**, karya **B.F Skiner**, seorang ahli ilmu jiwa. Konsep tersebut kemudian dikembangkan ke dalam pengajaran berprogram, dengan menggunakan mesin pengajaran. Mesin pengajaran tersebut berfungsi sebagai alat bantu (media) guru dalam melaksanakan kegiatan mengajarnya. Dengan demikian mesin tersebut tidak dapat menggantikan peran guru tersebut.

Munculnya teknologi komputer pada tahun 1950-an membawa perkembangan baru dalam dunia pendidikan. Komputer dipandang sebagai salah satu media yang dapat membantu meningkatkan teknologi pengajaran. Sistem komputer digunakan untuk menyampaikan pengajaran kepada siswa, melalui interaksi langsung antara siswa dengan materi pelajaran yang diprogram ke dalam komputer.

Pengajaran dengan bantuan komputer menurut [NAS'95] yaitu adalah :

2.2.1 Pengajaran dengan Bantuan Komputer

Pengajaran dengan bantuan komputer adalah pengajaran yang menggunakan komputer sebagai alat bantu dan dapat digunakan sebagai mesin belajar atau teaching machine, komputer sebagai alat bantu dapat memberikan bermacam-macam bantuan, seperti :

1. Menyimpan bahan pelajaran yang dapat dimanfaatkan kapan saja diperlukan.
2. Memberi informasi tentang berbagai referensi dan sumber-sumber serta alat audio-visual yang tersedia.

3. Memberi informasi tentang ruangan belajar, dan tenaga pengajar.
4. Memberi informasi hasil belajar murid.
5. Menyarankan kegiatan-kegiatan serta menilai tugas-tugas baru untuk dikerjakan selanjutnya.

2.2.2 Komputer untuk Manajemen Pengajaran

Komputer digunakan sekaligus oleh sejumlah besar pelajar, masing-masing dengan tugas tersendiri, maju menurut kecepatan masing-masing, dapat memberi keterangan tentang jumlah siswa yang mengerjakan unit tertentu serta proporsi siswa yang telah berhasil lulus dalam tes tertentu. Selain itu tenaga pengajar akan memperoleh daftar semua pemimpin kelompok, nama-nama siswa, score masing-masing, tugas-tugas yang diberikan, saran tentang hal-hal yang khusus harus diperhatikan oleh siswa, analisis tiap butir dalam tes itu, dan rangkuman tentang keseluruhannya.

2.2.3 Komputer sebagai Sarana Pemberi Informasi

Komputer dapat membantu dalam memberikan informasi tentang berbagai hal yang diperlukan oleh murid atau tenaga pengajar, misalnya tentang tiap bidang studi. Dengan bantuan komputer ini, murid maupun guru dapat memperoleh data atau informasi tentang topik yang sedang dipelajari atau dibicarakan. Secara individual, komputer ini dapat digunakan oleh siswa untuk mengulangi, memperluas dan memperdalam pengetahuannya, atau memperoleh informasi baru.

2.3 Rekayasa Perangkat Lunak

Perangkat lunak merupakan suatu program yang berisi sejumlah instruksi yang dibuat untuk melakukan sejumlah proses dengan tujuan sebagai alat pengendalian sistem komputer atau untuk menyelesaikan aplikasi tertentu.

Dalam membangun suatu perangkat lunak, selalu timbul beberapa kesulitan baik dari lingkungan intern maupun lingkungan ekstern. Untuk mengatasi kesulitan tersebut disusunlah suatu tata cara atau metodologi yang terangkum dalam Rekayasa Perangkat Lunak.

Rekayasa Perangkat Lunak adalah *suatu disiplin ilmu untuk membangun perangkat lunak, yang menggabungkan metode-metode untuk seluruh tahap-tahap pengembangan perangkat lunak, penentuan alat bantu yang baik untuk mengotomatisasikan metode-metode ini, pembuatan model perangkat lunak yang sangat berpengaruh untuk penerapan perangkat lunak, serta teknik-teknik yang baik untuk menjamin kualitas perangkat lunak.* [PRE 92]

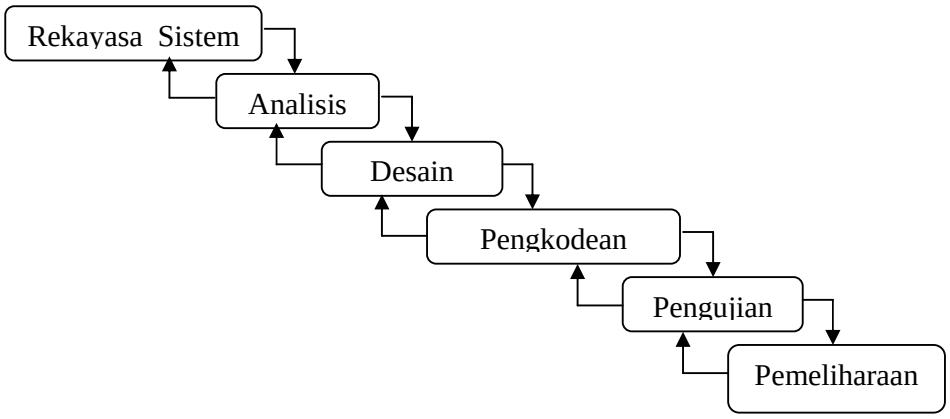
Untuk menghasilkan perangkat lunak yang baik maka :

a. Perangkat lunak yang dibuat harus mempunyai sifat yang mudah diperbaiki atau dikembangkan, sehingga dapat terus disesuaikan dengan perkembangan, baik dari teknologi maupun kebutuhan pengguna.
b. Perangkat lunak harus dapat diandalkan, yang mana kemampuan dari perangkat lunak diharapkan bekerja tanpa kesalahan.
c. Perangkat lunak harus bersifat efisien.
d. Perangkat lunak harus memberikan antarmuka yang tepat bagi pengguna.

2.4 Pendekatan Pembangunan Perangkat Lunak

Metodologi pembangunan perangkat lunak dikembangkan dan diterapkan agar dapat menghasilkan suatu perangkat lunak yang baik dan sesuai dengan kebutuhan pengguna.

Metodologi yang dipakai dalam pembangunan perangkat lunak Lelempengan adalah pendekatan Siklus Kerja Sederhana (*Classic life Cycle*) yang biasa disebut juga model *Waterfall* [bab 1], siklus ini bersifat sistematik, dengan pendekatan sekuensial untuk membangun perangkat lunak, dimulai pada tingkat sistem dan berkembang sampai dengan pengujian (*testing*) dan perbaikan kesalahan (*maintenance*). [HAN 95].



Gambar 2.1

Siklus kerja Rekayasa Perangkat Lunak dengan Metode Waterfall

Keterangan mengenai tiap-tiap langkah pembuatan perangkat lunak adalah sebagai berikut :

1. Rekayasa Sistem

   Karena perangkat lunak selalu merupakan bagian dari sistem, maka harus diawali dengan menentukan kebutuhan-kebutuhan bagi seluruh elemen-elemen sistem, dan kemudian mengalokasikan beberapa *subset* dari kebutuhan-kebutuhan tersebut bagi perangkat lunak. Rekayasa sistem mencakup kumpulan kebutuhan-kebutuhan pada setiap tingkat sistem dengan jumlah yang kecil pada tingkat teratas perancangan dan analisis.

2. Analisis

   Merupakan proses pengumpulan kebutuhan perangkat lunak. Untuk memahami program yang akan dibangun, harus dipahami wawasan informasi dari perangkat lunak, fungsi yang dibutuhkan, performansi dan antarmuka. Beberapa target yang ingin dicapai antara lain :

   1. Mengetahui keinginan pengguna
   2. Efisiensi biaya, sumber daya dan waktu pembangunan perangkat lunak
   3. Ketersediaan dukungan teknik

3. Desain

   Perancangan perangkat lunak sebenarnya merupakan kumpulan proses yang difokuskan pada 4 (empat) atribut yang berbeda, yaitu struktur data, arsitektur perangkat lunak, rincian prosedur dan karakteristik antarmuka.

Tahap ini merupakan tahap penerjemahan dari keperluan atau data yang telah dianalisa ke dalam bentuk yang mudah dimengerti oleh pemakai.

Beberapa target dari tahap ini adalah :

1. Spesifikasi perangkat lunak harus dapat dimengerti oleh pembangun perangkat lunak.
2. Perancangan modul-modul memungkinkan digunakan kembali baik secara parsial maupun total.
3. Pembentukan dokumentasi perancangan perangkat lunak.

4. Pengkodean

   Pada tahap ini perancangan harus diterjemahkan kedalam bentuk yang dapat dibaca oleh mesin. Yang menjadi target dari tahap ini adalah menuliskan program secara rinci pada setiap modul.

5. Pengujian

   Proses pengujian difokuskan pada *logical internal* dari perangkat lunak untuk memastikan bahwa semua pernyataan-pernyataan telah diuji, dan pada *functional external* untuk menemukan kesalahan-kesalahan dan memastikan bahwa masukan sesuai dengan kebutuhan.

   1. Yang menjadi target pada tahap ini adalah :
   2. Mendeteksi setiap kesalahan yang terjadi
   3. Memperbaiki setiap kesalahan yang terjadi
   4. Mencegah kesalahan baru selama perangkat lunak dimodifikasi
   5. Perangkat lunak dapat dipercaya / diandalkan.

6. Pemeliharaan

   Merupakan tahap perbaikan atau penyempurnaan perangkat lunak yang telah selesai akibat perubahan-perubahan atau penambahan yang terjadi sesuai kebutuhan atau permintaan pemakai. Untuk melakukannya, dipergunakan kembali setiap langkah pada siklus kerja terdahulu.

2.5 Pengembangan Perangkat Lunak

Secara umum, pengembangan perangkat lunak [MAH 96] terdiri dari 3 (tiga) tahap, yaitu :

1. Definisi

   Pada tahap ini dirumuskan apa yang akan diproses, fungsi dan performansi yang diinginkan, interface yang akan digunakan, serta prediksi kendala-kendala, batasan-batasan dan kriteria validasi yang diperlukan untuk menunjang keberhasilan sistem.

2. Pengembangan

   Pada tahap ini dijabarkan mengenai struktur data dan arsitektur perangkat lunak, prosedur rinci untuk penerapan, proses penerjemahan ke dalam pemrograman, serta perlakuan uji coba.

3. Perubahan

   Ada 3 (tiga) jenis perubahan atas perangkat lunak, yaitu :

   1. Perbaikan, yakni perubahan karena kekurangan.
   2. Adaptasi, yaitu perubahan yang mengacu terhadap lingkungan.
   3. Perluasan, yaitu perubahan karena permintaan pemakai.

2.6 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah teknik grafik yang digunakan untuk menjelaskan aliran informasi dan transformasi data yang bergerak dari masukan ke keluaran. DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur. Teknik ini sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau akan disimpan.

DFD menjelaskan bagaimana data masukan diubah menjadi keluaran, dimana setiap bagian pada diagram menjelaskan proses transformasi yang berbeda. DFD dapat digunakan untuk menggambarkan sistem atau perangkat lunak pada banyak tingkatan pada dari suatu abstraksi. DFD juga dapat digunakan untuk menggambarkan sistem atau perangkat lunak pada setiap tingkat dan dapat dipecah-pecah.

DFD dapat dibagi ke dalam tingkatan-tingkatan yang menggambarkan bertambahnya aliran data dan rincian fungsional. Level 1 pada DFD menyatakan prosedur-prosedur dari keseluruhan sistem yang digambarkan pada model konteks (*Context Model*).

Model konteks pada DFD merupakan model sistem pokok yang menggambarkan keseluruhan seluruh elemen perangkat lunak sebagai sebuah lingkaran dengan data masukan dan data keluaran dinyatakan dengan anak panah masuk atau keluar, berturut-turut. Model konteks menjelaskan bagaimana

gambaran sistem dan hubungan dengan pihak luar. Setiap proses dan jalur-jalur aliran data digambarkan pada tingkat yang lebih tinggi (misalnya level 2) yang menyatakan proses-proses dari keseluruhan prosedur yang digambarkan pada DFD level 1.

Untuk membuat DFD, digunakan simbol-simbol. Simbol-simbol ini dirancang untuk mempermudah penggambaran kepada pemakai. Berikut ini adalah simbol-simbol menurut *Yourdan–DeMarco.*

Persegi empat : Menyatakan entitas luar (*external entity*), yaitu sebuah sistem atau sistem lain yang menerima / menghasilkan informasi dari / bagi perangkat lunak.

Lingkaran : Menyatakan proses yang menggunakan dan mengubahnya informasi ke dalam bentuk lain.

Anak panah : Menyatakan satu atau banyak item data. Seluruh anak panah harus diberi nama.

Garis ganda : Menyatakan tempat penyimpanan data (*data storage*), yang menyimpan isi dari aliran data yang digunakan oleh sistem atau perangkat lunak.

2.7 Perancangan Antar Muka

Antarmuka suatu perangkat lunak menggambarkan bagaimana cara sebuah program akan dilihat dan digambarkan oleh pemakai. Perancangan sebuah antarmuka meliputi perumusan tampilan layar dan struktur menu.

Secara umum, fungsi antar muka bagi pemakai adalah sebagai berikut :

1. Penyampaian pesan dari pemakai ke sistem dan sebaliknya.
2. Validasi oleh sistem terhadap pesan dari pemakai dan pengiriman pesan kesalahan dari sistem ke pemakai.

Beberapa pertimbangan dari pedoman dalam melakukan perancangan antar muka, antara lain :

1. Sederhana
2. Konsisten
3. Membatasi jumlah parameter dan perintah
4. Menyediakan nilai dasar (*default value*)
5. Memberikan kemudahan penanganan kesalahan
6. Dapat membatalkan suatu aksi
7. Dukungan kontrol oleh pemakai
8. Memiliki fasilitas pembatalan (*undo*)

Berikut ini adalah penjelasan dari beberapa komponen antar muka yang penting dan harus dirancang :

2.7.1 Perancangan Menu

Beberapa pedoman perancangan menu adalah sebagai berikut :

1. Konsisten
2. Terorganisir
3. Terurut
4. Dapat kembali ke menu awal

5. Tersedia alternatif cara pemilihan item
6. Mempertimbangkan luas layar
7. Mempertimbangkan visual menu dalam hal warna, bentuk maupun letaknya
8. Memberikan keterangan tentang masukan
9. Menyediakan bantuan langsung
10. Mempertimbangkan waktu tanggap dan kecepatan tampilan

2.7.2 Perancangan Pengisian Masukan

1. Penggunaan istilah yang konsisten.
2. Memperhatikan frekuensi pemakaian.

2.7.3 Perancangan Fasilitas Bantuan Langsung Dan Pesan Kesalahan

Beberapa pedoman dalam perancangan fasilitas bantuan adalah sebagai berikut.

1. Singkat, jelas, konsisten dan mudah dipahami.
2. Pemanggilan bantuan dapat dilakukan dari semua titik pada sistem setiap saat.
3. Penayangan bantuan tidak mengganggu pekerjaan.

Beberapa pedoman dalam perancangan pesan kesalahan adalah sebagai berikut.

1. Dimunculkan jika pemakai membuat kesalahan saat interaksi.
2. Menjelaskan tentang jenis kesalahan dan cara mengatasinya.

3. Pesan tidak terlalu umum.
4. Lokasi dalam penayangannya harus konsisten.

2.8 Tinjauan Perangkat Lunak

2.8.1 Bahasa Pemrograman *VisualBasic*

Dalam pembuatan sebuah program, visual basic menggunakan sistem yang disebut RAD (*Rapid Aplication Development*). Sistem ini memanfaatkan bahasa pemrograman visual yang mempermudah bagi pemakainya mendesain tampilan program (*user interface*). Cara ini sangat bermanfaat untuk membuat program yang bekerja dalam sistem windows yang memang tampilan layarnya lebih rumit dibandingkan dengan sistem DOS dulu. Dengan bahasa pemrograman biasa (non visual), waktu seorang programmer akan lebih banyak dihabiskan untuk mendesain atau memperindah tampilan program daripada menulis program utamanya sendiri.

Program Visual Basic dapat dikembangkan dan bekerja di lingkungan windows 98 atau lebih. Visual Basic menjadikan pemakai tidak perlu waktu untuk membuat tampilan kotak dilayar, dialog, perangkat kontrol yang rumit dan menyita waktu dalam menuliskan program. Beberapa keunggulan lainnya adalah kecepatan eksekusinya yang cepat, kemudahan akses berbagai format database, sehingga menjadikan penulis memilih Visual Basic untuk merancang program aplikasi dari kecerdasan buatan yang dibuat.