BAB II

LANDASAN TEORI

2.1 Pengajaran

Pengajaran adalah kegiatan yang dilakukan guru dalam menyampaikan pengetahuan kepada siswa. Pengajaran juga diartikan sebagai interaksi belajar dan mengajar. Pengajaran berlangsung sebagai suatu proses yang saling mempengaruhi antara guru dan siswa.

Pembelajaran sebagai aktivitas atau kegiatan yang berfokus pada kondisi dan kepentingan pembelajar atau biasa disebut *learner centered*. Istilah pembelajaran digunakan untuk menggantikan istilah "pengajaran" yang lebih bersifat sebagai aktivitas yang berfokus pada guru atau *teacher centered* (Miarso, 2005).

2.1.1 Pengajaran Berbantuan Komputer

Komputer merupakan jenis media yang secara virtual dapat menyediakan respon yang segera terhadap hasil belajar yang dilakukan oleh siswa. Lebih dari itu, komputer memiliki kemampuan menyimpan dan memanipulasi informasi sesuai dengan kebutuhan. Perkembangan teknologi yang pesat saat ini telah memungkinkan komputer memuat dan menayangkan beragam bentuk media di dalamnya.

Saat ini teknologi komputer tidak lagi hanya digunakan sebagai sarana komputasi dan pengolahan kata (*word processor*) tetapi juga sebagai sarana belajar multi media yang memungkinkan peserta didik membuat desain dan

rekayasa suatu konsep dan ilmu pengetahuan. Sajian multimedia berbasis komputer dapat diartikan sebagai teknologi yang mengoptimalkan peran komputer sebagai sarana untuk menampilkan dan merekayasa teks, grafik, dan suara dalam sebuah tampilan yang terintegrasi. Dengan tampilan yang dapat mengkombinasikan berbagai unsur penyampaian informasi dan pesan, komputer dapat dirancang dan digunakan sebagai media teknologi yang efektif untuk mempelajari dan mengajarkan materi pembelajaran yang relevan.

2.1.2 Ruang Lingkup Pengajaran Berbantuan Komputer

Pengajaran berbantuan komputer memiliki tiga komponen utama yaitu :

1. Perangkat keras berupa komputer dan piranti-pirantinya.
2. Perangkat lunak berupa sistem operasi, perangkat lunak untuk mengembangkan modul, program komputer dan mempresentasikan materi (aplikasi tutorial).
3. Manusia pembuat atau pengembang modul atau pengguna pengajaran berbantuan komputer.

Apliaksi tutorial merupakan salah satu bentuk perangkat lunak pengajaran berbantuan komputer. Komponen-komponen aplikasi tutorial ini terdiri dari domain materi, sisem pengendali pengajaran dan antar muka pemakai.

1. Domain materi

   Pembuatan domain materi dilakukan dengan memindahkan bahan-bahan yang diperoleh dari buku referensi ke suatu program komputer.
2. Sistem pengendali pengajaran

Dengan membuat pengendali pengajaran, diharapkan diperoleh suatu aplikasi tutorial yang sisematis dan terarah.

3. Antar muka pemakai

   Antar muka pemakai adalah bagian yang sangat penting dalam aplikasi tutorial, sebab bagian inilah yang berhadapan langsung dengan pemakai.

Sistem-sistem komputer dapat menyampaikan pengajaran secara langsung kepada para pengguna melalui cara berinteraksi dengan materi yang diprogramkan kedalam sistem. Ada berbagai macam kemungkinan penggunaannya yang meliputi model-model mengajar sehingga komputer dapat memberikan kemudahan paling efekif.

2.2 Komputer Dalam Pendidikan

Komputer dapat didefinisikan sebagai alat yang dapat menerima informasi yang diterapkan untuk prosedur pemrosesan informasi dan memberikan hasil informasi baru dalam bentuk yang mudah digunakan oleh pemakai.

Komputer digital adalah informasi yang ditangani oleh komputer ini diubah kedalam bentuk digital yaitu kode berdasarkan sistem biner (*binary)* yang hanya menggunakan simbol angka 0 dan 1 sebelum diproses. Komputer ini bisa digunakan untuk tujuan pendidikan. Komputer analog adalah komputer yang didesain untuk menangani data yang belum dikonversikan kedalam bentuk digit.

2.2.1. Peranan Komputer Dalam Bidang Pendidikan

Penggunaan komputer dalam pendidikan adalah sebagai alat untuk belajar mandiri, dimana komputer memegang peranan penting dalam program belajar mengajar. Jika dihubungkan dengan peranan tersebut biasanya komputer digunakan sebagai salah satu dari dua metode yang berbeda. Pertama, komputer digunakan sebagai tutorial pengganti, dimana user atau siswa dapat mengikuti suatu diskusi yang sedang berlangsung melalui sebuah alat terminal untuk belajar secara interaktif. Kedua, komputer digunakan sebagai labolatorium simulasi yang dilengkapi berbagai fasilitas, dimana user atau siswa dapat melaksanakan eksperimen dengan sistem model yang telah diprogramkan kedalam komputer.

2.2.2. Faktor Penggunaan Komputer Dalam Pendidikan

Komputer dan teknologi informasi baru secara potensial mampu menyebabkan berbagai perubahan yang besar dan jauh untuk dijangkau oleh sistem pendidikan kita. Berbagai perubahan tersebut tidak dapat dibandingkan dengan apa yang pernah dihasilkan oleh media cetak sebagai teknologi informasi lama. Beberapa faktor yang mempengaruhi komputer dalam bidang pendidikan diantaranya :

1. Faktor-faktor Teknik :

   Secara umum untuk mempelajari komputer diperlukan kemauan untuk mempelajari hal-hal yang bersifat mendasar yaitu harus mempelajari berbagai bahasa baru dalam komputer dengan tujuan untuk berkomunikasi dengan komputer.

2. Faktor sifat pendidik

Sikap para guru atau dosen didalam memperkenalkan penggunaan teknologi baru tidak baik apabila orang yang akan mempergunakan teknologi memandang atau memanfaatkan teknologi baru tersebut masih berwawasan dan berperan secara tradisional.

3. Faktor Pendidikan Lainnya

Suatu hal nyata ditakuti oleh beberapa ahli pendidikan adalah dengan penggunaan komputer dan aspek-aspek informasi teknologi baru lainnya secara meningkat, maka akan mengarah pada situasi yang sangat tergantung pada sistem belajar bermedia yang bertentangan dengan belajar mengajar.

2.3 Konsep Dan Database

2.3.1 Pengertian Database

1. Database adalah sekumpulan data yang saling berhubungan (berelasi) dan disimpan di media penyimpanan elektronik.
2. Database adalah sekumpulan data yang terintegrasi yang diorganisasi untuk memenuhi kebutuhan pemakai untuk keperluan organisasi.

2.3.2 Istilah Dalam Database

1. Database : Sekumpulan data yang saling berhubungan untuk mencapai suatu tujuan.
2. Data : fakta-fakta yang dapat disimpan dan mempunyai arti tertentu.

3. Tabel : Tempat untuk menyimpan data, tabel terdiri dari field dan record.
4. Field : disebut juga dengan kolom, yaitu bagian tabel tempat menyimpan sebuah item data.
5. Record : disebut juga dengan baris, yaitu satu bagian informasi yang disimpan dalam tabel, misal data seorang mahasiswa akan disimpan dalam satu record yang terdiri dari beberapa kolom/field.

2.3.3 Mengenal Microsoft SQL Sever

SQL Server adalah sistem manajemen database relasional (RDBMS) yang dirancang untuk aplikasi dengan arsitektur client atau server. Istilah client, server, dan client/server dapat digunakan untuk merunjuk kepada konsep yang sangat umum atau hal yang spesifik dari perangkat keras atau perangkat lunak.

Pada level yang sangat umum, sebuah client adalah setiap komponen dari sebuah sistem yang meminta layanan atau sumber daya dari komponen sistem lainnya. Sedangkan sebuah server adalah setiap komponen sistem yang menyediakan layanan atau sumber daya ke komponen sistem lainnya.

Sistem client/server adalah dirancang untuk memisah layanan basisdata dari client, dengan penghubungnya menggunakan jalur komunikasi data. Layanan basis data diimplementasikan pada sebuah komputer yang berdaya guna, yang memungkinkan manajemen tersentralisasi, keamanan dan berbagai sumber daya. Oleh karena itu, server dalam client/server adalah basisdata dan layanannya. Aplikasi-aplikasi client diimpelemntasikan pada berbagai platform, menggunakan berbagai kakas pemrograman.

2.3.4 Mengenal SQL (*Structure Query language*)

SQL (*Structured Query Language*) adalah sebuah bahasa yang dipergunakan untuk mengakses data dalam basis data relasional. Bahasa ini secara *de facto* merupakan bahasa standar yang digunakan dalam manajemen basis data relasional. Saat ini hampir semua server basis data yang ada mendukung bahasa ini untuk melakukan manajemen datanya.

Secara umum, SQL terdiri dari dua bahasa, yaitu *Data Definition Language* (DDL) dan *Data Manipulation Language* (DML). Implementasi DDL dan DML berbeda untuk tiap sistem manajemen basis data (SMBD), namun secara umum implementasi tiap bahasa ini memiliki bentuk standar yang ditetapkan ANSI.

***Data Definition Language* (DDL)**

DDL digunakan untuk mendefinisikan, mengubah, serta menghapus basis data dan objek-objek yang diperlukan dalam basis data, misalnya tabel, view, user, dan sebagainya. Secara umum, DDL yang digunakan adalah CREATE untuk membuat objek baru, USE untuk menggunakan objek, ALTER untuk mengubah objek yang sudah ada, dan DROP untuk menghapus objek. DDL biasanya digunakan oleh administrator basis data dalam pembuatan sebuah aplikasi basis data.

CREATE

CREATE digunakan untuk membuat basis data maupun objek-objek basis data. SQL yang umum digunakan adalah:

"CREATE DATABASE nama_database"

"CREATE TABLE nama_tabel"

***Data Manipulation Language* (DML)**

DML digunakan untuk memanipulasi data yang ada dalam suatu tabel. Perintah yang umum dilakukan adalah:

- SELECT untuk menampilkan data
- INSERT untuk menambahkan data baru
- UPDATE untuk mengubah data yang sudah ada
- DELETE untuk menghapus data

SELECT

SELECT adalah perintah yang paling sering digunakan pada SQL, sehingga terkadang istilah query dirujukkan pada perintah SELECT. SELECT digunakan untuk menampilkan data dari satu atau lebih tabel, biasanya dalam sebuah basis data yang sama. Secara umum, perintah SELECT memiliki bentuk lengkap.

Fungsi aggregat

Beberapa SMBD memiliki fungsi aggregat, yaitu fungsi-fungsi khusus yang melibatkan sekelompok data (aggregat). Secara umum fungsi aggregat adalah:

- SUM untuk menghitung total nominal data
- COUNT untuk menghitung jumlah kemunculan data
- AVG untuk menghitung rata-rata sekelompok data
- MAX dan MIN untuk mendapatkan nilai maksimum/minimum dari sekelompok data.

Fungsi aggregat digunakan pada bagian SELECT. Syarat untuk fungsi aggregat diletakkan pada bagian HAVING, bukan WHERE.

2.4 Alat Bantu Pemodelan

2.4.1 Perancangan Berorientasi Objek

Rekayasa adalah bentuk upaya manusia untuk mencari atau merancang produk yang berkualitas. Produk dapat berupa sistem atau perangkat lunak yang sedang atau akan dibangun dengan satu model pengembangan. Orientasi objek adalah cara pandang algoritma yang di implementasikan pada bahasa pemrograman berorientasi objek.

OOA adalah teknik yang model driven yang mengintegrasikan data dan proses yang disebut objek. Model OOA adalah gambar-gambar yang mengilustrasikan objek-objek sistem dari berbagai perspektif, seperti struktur, perilaku dan interasi antar objek. Contoh yang paling terkenal adalah UML (Unified Modelling Language).

Cara berpikir orienasi objek adalah segala sesuatu dipandang dengan objek. Objek dapat berupa konsep, abstraksi, atau sesuatu dengan batas-batas tegas dan mempunyai arti untuk persoalan yang ditangani. Objek mempunyai identitas dan dapat dibedakan. Ada dua kegunaan objek yaitu : untuk meningkatkan pemahaman dunia nyata dan menyediakan dasar pengetahuan praktis untuk implementasi komputer.

Perangkat lunak masih seperti model yang digunakan dalam metodologi konvensional, yaitu dapat menggunakan model *waterfall*. Dalam model *waterfall*

terdapat lima tahapan proses, yaitu : tahap rekayasa sistem, analisis perancangan, pemrograman, pengujian dan pemeliharaan. Dengan menggunakan *waterfall*, memungkinkan adanya pengembangan aplikasi dikemudian hari, sehingga aplikasi yang dibangun dapat bertahan dalam jangka waktu yang panjang.

2.4.2 UML (*Unified Modeling Language)*

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta di itulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan class dan operation dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan syntax/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML syntax mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh

OMT (Object Modeling Technique), dan Ivar Jacobson OOSE (Object-Oriented Software Engineering).

Unified Modeling Language (UML) adalah alat bantu (tool) untuk pemodelan sistem, "UML adalah bahasa yang dapat digunakan untuk spesifikasi, visualisasi, dan dokumentasi sistem objek-oriented software pada fase pengembangan. UML merupakan unifikasi dari metode Booch, OMT, dan notasi Objectory, serta ide-ide terbaik metodologi lainnya. Dengan menyatukan notasi metode-metode objek oriented tersebut, UML merupakan standar dasar dalam bidang analisis dan desain berorientasi-objek" (Quatrani, 1998).

2.4.2.1 Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya.

Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Use case diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem.

Sebuah use case dapat meng-*include* fungsionalitas use case lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa use case yang di-*include* akan dipanggil setiap kali use case yang meng-*include* dieksekusi

secara normal. Sebuah use case dapat di-*include* oleh lebih dari satu use case lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*.

2.4.2.2 Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama (dan stereotype)
2. Atribut
3. Metoda

2.4.2.3 Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan state diagram khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya state sebelumnya (*internal processing*). Oleh karena itu activity diagram tidak

menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

2.4.2.4 Sequence Diagram

Sequence Diagram adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, *display*, dan sebagainya berupa ”pesan/*message*”.

Sequence Diagram digunakan untuk menggambarkan skrenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian untuk menghasilkan *output* tertentu. Sequence. Diagram diawali dari apa yang me-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

2.4.2.5 Component Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya, komponen piranti lunak adalah modul berisi code, baik berisi source code maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time, link time*, maupun *run time*.

Umumnya komponen terbentuk dari beberapa *class* atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

2.4.2.6 Deployment Diagram

Deployment/physical diagram menggambarkan *detail* bagaimana komponen di-*deploy* dalam infrastruktur sistem, dimana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal. Sebuah node adalah server, *workstation*, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar node (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.

2.5 E-Learning

E-learning memiliki banyak istilah yang secara garis bisa dapat disimpulkan yaitu merupakan cara baru dalam proses belajar mengajar yang menggunakan media elektronik baik intranet maupun internet sebagai sistem pembelajarannya. E-learning merupakan dasar dan konsekuensi logis dari perkembangan teknologi informasi dan komunikasi.

Beberapa ahli mencoba menguraikan pengertian e-learning menurut versinya masing-masing, diantaranya :

- Jaya Kumar C. Koran (2002)

  e-learning sebagai sembarang pengajaran dan pembelajaran yang menggunakan rangkaian elektronik (LAN, WAN, atau internet) untuk menyampaikan isi pembelajaran, interaksi, atau bimbingan.

- Dong (dalam Kamarga, 2002)

e-learning sebagai kegiatan belajar asynchronous melalui perangkat elektronik komputer yang memperoleh bahan belajar yang sesuai dengan kebutuhannya.

- Rosenberg (2001)

  Menekankan bahwa e-learning merujuk pada penggunaan teknologi internet untuk mengirimkan serangkaian solusi yang dapat meningkatkan pengetahuan dan keterampilan

- Darin E. Hartley [Hartley, 2001]

  eLearning merupakan suatu jenis belajar mengajar yang memungkinkan tersampaikannya bahan ajar ke siswa dengan menggunakan media Internet, Intranet atau media jaringan komputer lain.

- LearnFrame.Com dalam Glossary of eLearning Terms [Glossary, 2001]

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

Secara umum pengertian e-learning telah disebutkan diatas, namun pada penyusunan tugas akhir ini, penerapan konsep e-learning dalam pembelajaran di laboratorium komputer SMAN 5 mengambil beberapa fitur umum e-learning pada umumnya namun ada penambahan fitur di aplikasi sebagai sarana pembantu sistem KBM

.

2.6 Sistem Aplikasi Pendukung

2.6.1 Sekilas Tentang *Microsoft Visual C#*

C# adalah bahasa pemrograman baru yang diciptakan Microsoft yang digunakan oleh banyak developer .NET untuk mengembangkan aplikasi dengan platform .NET. Berikut beberapa kelebihan bahasa C# :

a. **Sederhana**

C# bersifat sederhana, karena bahasa ini didasarkan kepada bahasa C dan C++. Seperti C dan C++ atau bahkan Java dalam C# pun akan ditemukan aspek-aspek yang begitu familiar, seperti *statements*, *expression*, *operators* dan beberapa fungsi yang diadopsi langsung dari C dan C++, tetapi dengan berbagai perbaikan yang membuat bahasanya menjadi lebih sederhana.

b. **Object Oriented Language**

C# memenuhi syarat-syarat sebagai sebuah bahasa pemrograman yang bersifat Object Oriented, yaitu encapsulation, inheritance dan polymorphism.

c. **Powerfull dan Fleksibel**

C# bisa digunakan untuk membuat berbagai macam aplikasi, seperti aplikasi pengolah kata, grafik, *spreadsheets*, atau bahkan membuat kompiler untuk sebuah bahasa permrograman.

d. **Efisien**

C# tidak memiliki terlalu banyak keyword, sehingga dapat mengurangi kerumitan.

e. **Modular**

Kode C# ditulis dengan pembagian masing Class-Class (*classes*) yang terdiri dari beberapa *routines* yang disebut sebagai *member methods*. Class-Class dan metode-metode ini dapat digunakan kembali oleh program atau aplikasi lain. Hanya dengan memberikan informasi yang dibutuhkan oleh Class dan metode yang dimaksud, maka dapat dibuat suata kode yang dapat digunakan oleh satu atau beberapa aplikasi dan program (*reusable code*).

2.6.2 Telerik RadControls

Telerik RadControls adalah toolset paling inovatif dan komprehensif untuk pengembangan ASP.NET profesional, membual kemampuan AJAX, Silverlight, Windows Phone 7, WPF dan Windows Forms. Pembangunan aplikasi ini menggunakan RadControls untuk WinForms 2011.3.1229.

RadControls untuk WinForms termasuk kontrol UI yang memiliki lebih dari 50 yang dapat digunakan untuk dengan mudah membangun visual yang menarik, Line kinerja tinggi aplikasi Bisnis dan Metro. RadControls inipun menawarkan arsitektur yang dirancang dengan baik, kemampuan sentuh, dukungan untuk Visual Studio 2010 dan CAB, Telerik Windows Forms Controls banyak digunakan untuk pengembangan desktop Anda Enterprise.