

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Perancangan**

Desain atau perancangan dalam pengembangan perangkat lunak merupakan upaya untuk mengkonstruksi sebuah sistem yang memberikan kepuasan (mungkin informal) akan spesifikasi kebutuhan fungsional memenuhi target, memenuhi kebutuhan secara implisit dan eksplisit dari segi performansi maupun penggunaan sumber daya, kepuasan batasan pada proses desain dari segi biaya, waktu, dan perangkat. (Rosa A.S, M. Shalahuddin; Modul Pembelajaran Rekayasa Perangkat Lunak ; 2011: 20).

#### **II.2. Aplikasi**

Perangkat Lunak (*software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*). Sebuah program komputer tanpa terasosiasi dengan dokumentasinya maka belum dapat dikatakan perangkat lunak (*software*).

Karakter perangkat lunak adalah sebagai berikut :

- a. Perangkat lunak dibangun dengan rekayasa (*software engineering*) bukan diproduksi secara manufaktur atau pabrikan.
- b. Perangkat lunak tidak pernah usang (*wear out*) karena kecacatan dalam perangkat lunak dapat diperbaiki.

- c. Barang produksi pabrikan biasanya komponen barunya akan terus diproduksi, sedangkan perangkat lunak biasanya terus diperbaiki seiring bertambahnya kebutuhan.

Aplikasi dari perangkat lunak adalah sebagai berikut :

- a. Perangkat lunak sistem (*system software*)

Adalah kumpulan program dimana program yang satu ditulis untuk memenuhi kebutuhan program lainnya.

- b. Perangkat lunak waktu nyata (*real time software*)

Merupakan perangkat lunak yang memonitor, menganalisis, mengontrol sesuatu secara waktu nyata (*real time*). Reaksi yang dibutuhkan pada perangkat lunak harus langsung menghasilkan respon yang diinginkan.

- c. Perangkat lunak bisnis (*business software*)

Merupakan perangkat lunak pengelola informasi bisnis (seperti akuntansi, penjualan, pembayaran, dan penyimpanan (*inventory*)).

- d. Perangkat lunak untuk keperluan rekayasa dan keilmuan (*engineering and scientific software*)

Merupakan perangkat lunak yang mengimplementasikan algoritma yang terkait dengan keilmuan ataupun perangkat lunak yang membantu keilmuan, misalkan perangkat lunak di bidang astronomi, di bidang matematika, dan lain sebagainya.

- e. Perangkat lunak tambahan untuk membantu mengerjakan suatu fungsi dari perangkat lunak yang lainnya (*embedded software*)

Misalnya perangkat lunak untuk mencetak dokumen ditambahkan agar perangkat lunak yang memerlukan dapat mencetak laporan, maka perangkat lunak untuk mencetak dokumen ini disebut *embedded software*.

f. Perangkat lunak komputer personal (*personal computer software*)

Merupakan perangkat lunak untuk PC misalnya perangkat lunak pemroses teks, pemroses grafik, dan lain sebagainya.

g. Perangkat lunak berbasis web (*web based software*)

Merupakan perangkat lunak yang dapat diakses dengan menggunakan web browser.

h. Perangkat lunak inteligensia buatan (*artificial intelligence software*)

Merupakan perangkat lunak yang menggunakan algoritma tertentu untuk mengelola data sehingga seakan-akan memiliki inteligensia seiring bertambahnya data yang diproses.

Produk perangkat lunak yang dibuat oleh pengembang (*developer*) perangkat lunak terdiri dari dua jenis :

a. Produk Generik

Produk perangkat lunak yang dibuat oleh pengembang perangkat lunak untuk dijual atau dipopulerkan (*open source*) tanpa ada yang memesan terlebih dahulu, perangkat lunak yang termasuk dalam produk generik misalnya perangkat lunak sistem operasi, perangkat lunak pendukung perkantoran untuk membuat dokumen, *slide* presentasi, atau perhitungan dalam bentuk *papersheet* dan lain sebagainya.

b. Produk Pemesanan

Produk perangkat lunak yang dibuat karena ada pelanggan yang melakukan pemesanan, misalnya sebuah instansi memerlukan perangkat lunak untuk memenuhi proses bisnis yang terjadi di instansinya, maka instansi itu akan bekerja sama dengan pengembang untuk membuat perangkat lunak yang diinginkan. (Rosa A.S, M. Shalahuddin; Modul Pembelajaran Rekayasa Perangkat Lunak ; 2011: 2-4)

### **II.3. Media Pembelajaran**

Kata media merupakan bentuk jamak dari kata medium. Medium dapat didefinisikan sebagai perantara atau pengantar terjadinya komunikasi dari pengirim menuju penerima (Heinich *et.al.*, 2002; Ibrahim, 1997; Ibrahim *et.al.*, 2001). Media merupakan salah satu komponen komunikasi, yaitu sebagai pembawa pesan dari komunikator menuju komunikan (Criticos, 1996). Berdasarkan definisi tersebut, dapat dikatakan bahwa proses pembelajaran merupakan proses komunikasi. Proses pembelajaran mengandung lima komponen komunikasi, guru (komunikator), bahan pembelajaran, media pembelajaran, siswa (komunikan), dan tujuan pembelajaran. Jadi, Media pembelajaran adalah segala sesuatu yang dapat digunakan untuk menyalurkan pesan (bahan pembelajaran), sehingga dapat merangsang perhatian, minat, pikiran, dan perasaan siswa dalam kegiatan belajar untuk mencapai tujuan belajar. (I Wayan Santyasa ; Landasan Konseptual Media Pembelajaran; 2007 : 3).

#### **II.4. Pembelajaran**

Menurut Gagne, Briggs, dan wagner dalam Udin S. Winataputra (2008) pengertian pembelajaran adalah serangkaian kegiatan yang dirancang untuk memungkinkan terjadinya proses belajar pada siswa.

Menurut UU Nomor 20 tahun 2003 tentang Sisdiknas, pembelajaran adalah proses interaksi peserta didik dengan pendidik dan sumber belajar pada suatu lingkungan belajar.

Pembelajaran merupakan bantuan yang diberikan pendidik agar dapat terjadi proses pemerolehan ilmu dan pengetahuan, penguasaan kemahiran dan tabiat, serta pembentukan sikap dan kepercayaan pada peserta didik. Pembelajaran adalah proses interaksi peserta didik dengan pendidik dan sumber belajar pada suatu lingkungan belajar pembentukan sikap dan kepercayaan pada peserta didik agar dapat belajar dengan baik.

Proses pembelajaran dialami sepanjang hayat seseorang manusia serta dapat berlaku dimanapun dan kapanpun. Pembelajaran mempunyai pengertian mirip dengan pengajaran, walaupun mempunyai konotasi yang berbeda. Dalam konteks pendidikan, guru mengajar supaya peserta didik dapat belajar dan menguasai isi pelajaran hingga mencapai sesuatu objektif yang ditentukan (aspek kognitif), juga dapat mempengaruhi perubahan sikap (aspek efektif), serta keterampilan (aspek psikomotor) seseorang peserta didik.

Pengajaran memberi kesan hanya sebagai pekerjaan satu pihak, yaitu pekerjaan guru saja. Sedangkan pembelajaran juga menyiratkan adanya interaksi

antara guru dengan peserta didik Wasty Soemanto. 1987. (Psikologi Pendidikan. Jakarta: PT Bina Aksara).

Ada beberapa istilah pembelajaran, diantaranya adalah :

### 1. Metode

Metode merupakan upaya untuk mengimplementasikan rencana yang sudah disusun dalam kegiatan nyata agar tujuan yang telah disusun tercapai secara optimal. Metode digunakan untuk merealisasikan strategi yang telah ditetapkan. Strategi menunjuk pada sebuah perencanaan untuk mencapai sesuatu, sedangkan metode adalah cara yang dapat digunakan untuk melaksanakan strategi.

### 2. Pendekatan (*Approach*)

Pendekatan merupakan titik tolak atau sudut pandang kita terhadap proses pembelajaran. Strategi dan metode pembelajaran yang digunakan dapat bersumber dari pendekatan tertentu. Misalnya pendekatan dalam pembelajaran yaitu pendidikan yang berpusat pada guru dan pendekatan yang berpusat pada siswa. Pendekatan yang berpusat pada guru menurunkan strategi pembelajaran langsung, pembelajaran deduktif atau pembelajaran ekspositori. Sedangkan pendekatan yang berpusat pada siswa menurunkan strategi pembelajaran *discovery* dan inkuiri serta strategi pembelajaran induktif.

### 3. Teknik

Teknik adalah cara yang dilakukan seseorang dalam rangka mengimplementasikan suatu metode. Misalnya, cara yang harus dilakukan agar metode ceramah berjalan efektif dan efisien. Dengan demikian, sebelum seseorang melakukan proses ceramah sebaiknya memperhatikan kondisi dan

situasi. Misalnya, berceramah pada siang hari setelah makan siang dengan jumlah siswa yang banyak tentu saja akan berbeda jika ceramah itu dilakukan pada pagi dengan jumlah siswa yang terbatas.

#### 4. Taktik

Taktik adalah gaya seseorang dalam melaksanakan suatu teknik atau metode tertentu. Taktik sifatnya lebih individual, walaupun dua orang sama-sama menggunakan metode ceramah dalam situasi dan kondisi yang sama, sudah pasti mereka akan melakukannya secara berbeda, misalnya dalam taktik menggunakan ilustrasi atau menggunakan gaya bahasa agar materi yang disampaikan mudah dipahami. Dari penjelasan di atas, dapat disimpulkan bahwa suatu strategi pembelajaran yang diterapkan guru akan tergantung pada pendekatan yang digunakan, sedangkan bagaimana menjalankan strategi itu dapat ditetapkan berbagai metode pembelajaran. Dalam upaya menjalankan metode pembelajaran guru dapat menentukan teknik yang dianggapnya relevan dengan metode, dan penggunaan teknik itu setiap guru memiliki taktik yang mungkin berbeda antara guru yang satu dengan yang lain (Djiwandono dan Siti Wuryani.2002.Psikologi Pendidikan.Jakarta:Grasindo).

(Sumber :[http://husamah.staff.umm.ac.id/files/2010/03/belajar-dan\\_pembelajaran-  
kel-1-Autosaved-2.pdf](http://husamah.staff.umm.ac.id/files/2010/03/belajar-dan_pembelajaran-<br/>kel-1-Autosaved-2.pdf) )

## **II.5. Metode Membaca Al-Quran (Metode Iqra')**

Metode iqro' disusun oleh KH. As'ad Humam dari Kotagede Yogyakarta dan dikembangkan oleh AMM (Angkatan Muda Masjid dan Musholla) Yogyakarta. 10 sifat buku Iqro' adalah :

1. Bacaan langsung
2. CBSA (Cara Belajar Santri Aktif)

Pada sistem ini guru hanya berperan sebagai penyimak saja, jangan sampai menuntun, kecuali hanya memberikan contoh pokok pelajaran.

3. Privat

Sistem ini menerapkan penyimak secara seorang demi seorang.

4. Modul
5. Asistensi
6. Praktis
7. Sistematis
8. Variatif
9. Komunikatif
10. Fleksibel

## **II.6. Pemodelan UML**

Pemodelan adalah gambaran dari realita yang simpel dan dituangkan dalam bentuk pemetaan dengan aturan tertentu. (Rosa A.S, M. Shalahuddin; Modul Pembelajaran Rekayasa Perangkat Lunak ; 2011: 116)

Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang

dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language (UML)*. *UML* hanya bergungsi untuk melakukan pemodelan. Jadi penggunaan *UML* tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya *UML* paling banyak digunakan pada metodologi berorientasi objek. (Rosa A.S, M. Shalahuddin; Modul Pembelajaran Rekayasa Perangkat Lunak ; 2011: 118)

Diagram *UML* terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori yaitu :

### *1. Structure Diagram*

Yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. Yang termasuk dalam structure diagram adalah sebagai berikut :

- a. Class diagram*
- b. Object diagram*
- c. Component diagram*
- d. Composite diagram*
- e. Package diagram*
- f. Deployment diagram*

### *2. Behavior Diagram*

Yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem. Yang termasuk dalam *behavior* diagram adalah sebagai berikut :

a. *Use case diagram*

b. *Activity diagram*

### 3. *Interaction Diagram*

Yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi antar sub sistem pada suatu sistem. Yang termasuk dalam *interaction diagram* adalah sebagai berikut :

a. *Sequence diagram*



b. *Communication diagram*





c. *Timing diagram*



d. *Interaction overview diagram* (Rosa A.S, M. Shalahuddin; Modul Pembelajaran Rekayasa Perangkat Lunak ; 2011: 120-121)

Dalam merancang UML (*Unified Modelling Language*) ada notasi-notasi yang telah ditentukan sebagaimana terdapat pada tabel II.I.

**Tabel II.1. Notasi-notasi UML (*Unified Modelling Language*)**

Notasi	Keterangan
<p><b>Actor</b></p> 	<p><i>Actor</i> menggambarkan segala pengguna <i>software</i> aplikasi (<i>user</i>). <i>Actor</i> memberikan suatu gambaran jelas tentang apa yang harus dikerjakan <i>software</i> aplikasi. Sebagai contoh sebuah actor dapat memberikan input kedalam dan menerima informasi dari <i>software</i> aplikasi, perlu dicatat bahwa sebuah <i>actor</i> berinteraksi dengan <i>use case</i>, tetapi tidak memiliki kontrol atas <i>use case</i>. Sebuah <i>actor</i> mungkin seorang manusia, satu <i>device</i>, <i>hardware</i> atau sistem informasi lainnya.</p>
<p><b>Use Case</b></p> 	<p><i>Use case</i> menjelaskan urutan kegiatan yang dilakukan <i>actor</i> dan sistem untuk mencapai suatu tujuan tertentu. Walaupun menjelaskan kegiatan, namun <i>use case</i> hanya menjelaskan apa yang dilakukan oleh <i>actor</i> dan sistem bukan bagaimana <i>actor</i> dan sistem melakukan kegiatan tersebut.</p> <p>1. <i>Use-case</i> Konkret adalah <i>use case</i> yang dibuat langsung karena keperluan <i>actor</i>. <i>Actor</i> dapat melihat</p>

	<p>dan berinisiatif terhadapnya .</p> <p>2. <i>Use-case</i> Abstrak adalah <i>use case</i> yang tidak pernah berdiri sendiri. <i>Use case</i> abstrak senantiasa termasuk didalam (<i>include</i>), diperluas dari (<i>extend</i>) atau memperumum (<i>generalize</i>) <i>use case</i> lainnya. Untuk menggambarannya dalam <i>use case</i> model biasanya digunakan <i>association relationship</i> yang memiliki <i>stereotype include, extend</i> atau <i>generalization relationship</i>. Hubungan <i>include</i> menggambarkan bahwa suatu <i>use case</i> seluruhnya meliputi fungsionalitas dari <i>use case</i> lainnya. Hubungan <i>extend</i> antar <i>use case</i> berarti bahwa satu <i>use case</i> merupakan tambahan fungsionalitas dari <i>use case</i> yang lain jika kondisi atau syarat tertentu terpenuhi.</p>
<p><b>Interface</b></p> 	<p><i>Interface</i> merupakan kumpulan operasi tanpa implementasi dari suatu <i>class</i>. Implementasi operasi dalam <i>interface</i> dijabarkan oleh operasi didalam <i>class</i>. Oleh karena itu keberadaan <i>interface</i> selalu disertai oleh <i>class</i> yang mengimplementasikan operasinya. <i>Interface</i> ini merupakan salah satu cara mewujudkan prinsip enkapsulasi dalam obyek.</p>
<p><b>Interaction</b></p> 	<p><i>Interaction</i> digunakan untuk menunjukkan baik aliran pesan atau informasi antar obyek maupun hubungan antar obyek. Biasanya <i>interaction</i> ini dilengkapi juga dengan teks bernama operation <i>signature</i> yang tersusun dari nama operasi, parameter yang dikirim dan tipe parameter yang dikembalikan.</p>
<p><b>Dependency</b></p> 	<p><i>Dependency</i> merupakan relasi yang menunjukkan bahwa perubahan pada salah satu elemen memberi pengaruh pada elemen lain. Elemen yang ada di bagian tanda panah adalah elemen yang tergantung pada elemen yang ada dibagian tanpa tanda panah. Terdapat 2 <i>stereotype</i> dari <i>dependency</i>, yaitu <i>include</i> dan <i>extend</i>. <i>Include</i> menunjukkan bahwa suatu bagian dari elemen (yang ada digaris tanpa panah) memicu eksekusi bagian dari elemen lain (yang ada di garis dengan panah). <i>Extend</i> menunjukkan bahwa suatu bagian dari elemen di garis tanpa panah bisa disisipkan kedalam elemen yang ada di garis dengan panah.</p>
<p><b>Association</b></p> 	<p><i>Association</i> menggambarkan navigasi antar <i>class</i> (<i>navigation</i>), berapa banyak obyek lain yang bisa berhubungan dengan satu obyek (<i>multiplicity</i> antar <i>class</i>) dan apakah suatu <i>class</i> menjadi bagian dari <i>class</i> lainnya (<i>aggregation</i>). <i>Navigation</i> dilambangkan dengan penambahan tanda panah di akhir garis. <i>Bidirectional navigation</i> menunjukkan bahwa dengan mengetahui salah</p>

	satu <i>class</i> bisa didapatkan informasi dari <i>class</i> lainnya. Sementara <i>UniDirectional navigation</i> hanya dengan mengetahui <i>class</i> diujung garis <i>association</i> tanpa panah kita bisa mendapatkan informasi dari <i>class</i> di ujung dengan panah, tetapi tidak sebaliknya. <i>Aggregation</i> mengacu pada hubungan has-a , yaitu bahwa suatu <i>class</i> memiliki <i>class</i> lain, misalnya Rumah memiliki <i>class</i> Kamar.
<p><b>Generalization</b></p> 	<i>Generalization</i> menunjukkan hubungan antara elemen yang lebih umum ke elemen yang lebih spesifik. Dengan <i>generalization</i> , <i>class</i> yang lebih spesifik ( <i>subclass</i> ) akan menurunkan atribut dan operasi dari <i>class</i> yang lebih umum ( <i>superclass</i> ) atau <i>subclass</i> is <i>superclass</i> . Dengan menggunakan notasi <i>generalization</i> ini, konsep <i>inheritance</i> dari prinsip hirarki dapat dimodelkan.
<p><b>Realization</b></p> 	<i>Realization</i> menunjukkan hubungan bahwa elemen yang ada di bagian tanpa panah akan merealisasikan apa yang dinyatakan oleh elemen yang ada di bagian dengan panah. Misalnya <i>class</i> merealisasikan <i>package</i> , <i>component</i> merealisasikan <i>class</i> atau <i>interface</i> .

**Sumber : (Rosa A.S, M. Shalahuddin; 2011: 123-139)**

## II.7. Pemrograman *Visual Basic 2008*

*Microsoft Visual Basic Studio 2008* merupakan kelanjutan dari *Microsoft Visual Studio* sebelumnya, yaitu *Visual Studio.Net 2003* yang diproduksi oleh *Microsoft*. Pada bulan Februari tahun 2002 *Microsoft* memproduksi teknologi *.Net Framework versi 1.0*, Teknologi *.Net* ini didasarkan atas susunan berupa *.Net Framework*, sehingga setiap produk baru yang terkait dengan teknologi *.Net* akan selalu berkembang mengikuti perkembangan *.Net Framework*-nya. Pada perkembangan nantinya, mungkin untuk membuat program dengan teknologi *.Net*, dan memungkinkan para pengembang perangkat lunak akan dapat menggunakan lintas sistem operasi, yaitu dapat dikembangkan di sistem operasi

*Windows* juga dapat dijalankan pada sistem operasi *Linux*, seperti yang telah dilakukan pada pemrograman *Java* oleh *Sun Microsystem*. Pada saat ini perusahaan-perusahaan sudah banyak meng-*update* aplikasi yang lama yang dibuat dengan *Microsoft Visual Basic 6.0* ke teknologi *.Net* karena kelebihan-kelebihan yang ditawarkan, terutama memungkinkan pengembang perangkat lunak secara cepat mampu membuat program yang *robust*, serta berbasiskan intergrasi ke *internet* yang dikenal dengan *XML Web Service*. (Ketut Darmayuda ; Pemrograman Aplikasi *Database* dengan *Microsoft Visual Basic .Net 2008* ; 2009 :1-2).

Lingkungan kerja pada *Microsoft Visual Basic. Net 2008* umumnya memiliki lingkungan kerja sebagai berikut :

1. *Tilte Bar*, berfungsi untuk menampilkan *project* yang aktif atau yang sedang dikembangkan.
2. *Menu Bar*, berfungsi untuk pengelolaan fasilitas yang dimiliki oleh *Visual Basic. Net 2008*, sedangkan *Tool Bar* berfungsi untuk melakukan perintah khusus secara cepat.
3. *Form*, adalah objek utama yang berfungsi untuk meletakkan objek-objek yang terdapat pada *Tool Box* yang digunakan dalam melakukan perancangan sebuah tampilan program aplikasi.
4. *Tool Box*, berfungsi untuk menyediakan objek-objek atau komponen yang digunakan dalam merancang sebuah *form* pada program aplikasi.
5. *Solution Explorer*, berfungsi untuk menampilkan *project* beserta *file-file* pendukung yang terdapat pada sebuah program aplikasi.

6. *Project Windows*, berfungsi untuk mengatur *properties* pada objek (*setting object*) yang diletakkan pada sebuah *form*.

(Ketut Darmayuda ; Pemrograman Aplikasi *Database* dengan *Microsoft Visual Basic .Net 2008* ; 2009 :14-16).

*Microsoft Visual Basic 2008* merupakan bagian dari kelompok bahasa pemrograman *Visual Studio 2008* yang dikembangkan oleh *Microsoft*. *Visual Studio 2008* terdiri dari beberapa bahasa pemrograman diantaranya adalah *Microsoft Visual Basic 2008*, *Microsoft C#*, dan *Visual Web Developer 2008*.

*Visual Studio* ini telah mengalami perubahan versi mulai dari *Visual Studio 6.0*, *Visual Studio 2005*, *Visual Studio 2006* dan yang terakhir adalah *Visual Studio 2008*.

*Microsoft Visual Basic 2008* setara dengan *Microsoft Visual Basic 9.0* yang memiliki kelebihan-kelebihan yaitu suport dengan bahasa *query Language-Integrated Query (LINQ)* dan suport dengan *database Microsoft SQL Server Compact 3.5*. Selain itu, kelebihan lain adalah memiliki *Object Relation Designer (O/R Designer)* untuk membantu mengedit *LINQ* ke *SQL* dihubungkan dengan *database* dan *fiture* lain, seperti *WPF (Windows Presentation Foundation)* dan *WCF (Windows Communication Foundation)*. Semua hal yang baru tersebut di atas menambah kelengkapan aplikasi *Microsoft Visual Basic 2008* dalam membuat media dan dokumen. (Hendrayudi ; 2011 : 1).

## II.8. 3DS Max Design 2009

*3DS Max Design 2009* merupakan *software graphic* keluaran *Autodesk* yang dirilis untuk mendampingi *3DS Max 2009*. *3DS Max Design 2009* merupakan nama baru dari aplikasi serupa milik *Autodesk* yakni *3DS Viz*, aplikasi ini sudah *discontinue* atau tidak dijual lagi. Baik *3DS Max Design* atau *3DS Max*, keduanya sama yakni merupakan *software* yang ditujukan untuk membuat *3D modelling*. Bedanya, *3DS max design* lebih ditujukan untuk menunjang kinerja *engineer* dalam bidang arsitektur, desain interior, dan bidang *engineer* lainnya. Sedangkan *3DS Max* ditujukan untuk para animator, pembuat *game*, pembuat film dan bidang animasi *3D* lainnya. *3DS Max Design 2009* memadukan antara *graphic vector* dengan *raster image* sehingga menghasilkan *Virtual Reality*. (Wahana Komputer ; *3D Studio Max Design 2009* ; 2009 : 1).

### II.8.1. Kebutuhan Sistem

Berikut adalah kebutuhan sistem yang harus dimiliki komputer untuk menggunakan *3DS Max Design 2009* dengan lancar :

1. Sistem operasi *Windows XP Professional SP2*
2. Prosesor *Intel Pentium 4* atau *AMD Athlon XP*
3. Memori minimal sebesar 512 MB
4. Ruang kosong *hardisk* minimal 500 MB
5. *VGA* yang mendukung *OpenGL* dan *Direct3D*
6. *Drive DVD-ROM* (Wahana Komputer ; *3D Studio Max Design 2009* ; 2009 : 1)

### **II.8.2. Area Kerja 3DS Max Design 2009**

Lingkungan area kerja *3DS Max Design 2009* umumnya memiliki lingkungan kerja sebagai berikut :

1. *Title Bar*, berfungsi untuk menampilkan nama aplikasi dan nama *file* yang aktif.
2. *Menu Bar* berisi menu untuk mengakses perintah *3DS Max Design 2009*.
3. *Tool Bar*, berisi ikon-ikon untuk mempermudah mengakses perintah.
4. *Main Tool Bar*, terdiri dari beberapa tombol.
5. *Viewport*, terdiri dari empat buah bidang dengan ukuran yang sama besar.
6. *Command Panel*, berfungsi untuk melakukan pengaturan terhadap objek *3DS Max Design 2009* yang dibuat menggunakan parameter objek, selain itu *command panel* juga dapat digunakan untuk mengakses perintah.
7. *Status Bar*, merupakan fasilitas yang digunakan untuk melihat informasi yang ada, misalnya melihat posisi objek terpilih terhadap sumbu X, Y, dan Z.
8. *Viewprot Navigation Control*, berfungsi untuk mengganti navigasi secara otomatis. (Wahana Komputer ; *3D Studio Max Design 2009* ; 2009 : 6-17)

### **II.9. Database (Basis Data)**

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. (Rosa A.S, M. Shalahuddin; Modul Pembelajaran Rekayasa Perangkat Lunak ; 2011: 44)

DBMS (*Database Management System*) atau dalam bahasa Indonesia sering disebut Sistem Manajemen Basis Data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data. Suatu sistem aplikasi disebut DBMS jika memenuhi persyaratan minimal sebagai berikut ;

- a. Menyediakan fasilitas untuk mengelola akses data
- b. Mampu menangani integritas data
- c. Mampu menangani akses data yang dilakukan secara bersamaan
- d. Mampu menangani back-up data (Rosa A.S, M. Shalahuddin; Modul Pembelajaran Rekayasa Perangkat Lunak ; 2011: 45)

Berikut ini adalah 4 macam DBMS versi komersial yang paling banyak digunakan di dunia saat ini, yaitu :

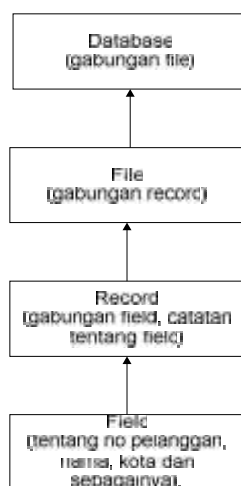
1. *Oracle*
2. *Microsoft SQL Server*
3. *IBM DB2*
4. *Microsoft Access*

Sedangkan DBMS versi open source yang cukup berkembang dan paling banyak digunakan saat ini adalah sebagai berikut :

1. *MySQL*
2. *PostgreSQL*
3. *Firebird*
4. *SQLite*

Hampir semua DBMS mengadopsi SQL sebagai bahasa untuk mengelola data pada DBMS. (Rosa A.S, M. Shalahuddin; Modul Pembelajaran Rekayasa Perangkat Lunak ; 2011: 46)

Sistem *database* merupakan gabungan *file* yang membentuk *file* utama (*master file*) yang saling terkait, dalam *database file* merupakan gabungan *record* dan *record* merupakan gabungan dari *field* atau *atribut* dari entitas. Terlihat pada gambar II.1.



**Gambar II.1. Skema Hirarki Data dalam Database**  
**Sumber : Mardi (2011 : 34)**

### **II.9.1. ERD (*Entity Relationship Diagram*)**

ERD (*Entity Relationship Diagram*) adalah gambar atau diagram yang menunjukkan informasi dibuat, disimpan, dan digunakan dalam sebuah sistem bisnis. Entitas biasanya menggambarkan jenis informasi yang sama. Dalam entitas digunakan untuk menghubungkan antar entitas yang sekaligus menunjukkan hubungan antar data. Pada akhirnya ERD juga bisa digunakan untuk menunjukkan aturan-aturan bisnis yang ada pada sistem informasi yang akan dibangun.

Elemen-elemen dari ERD adalah sebagai berikut :

1. *Entitas*, biasanya berupa orang, kejadian, atau benda dimana data akan dikumpulkan. Untuk menjadi entitas suatu objek harus menampilkan beberapa kali *event*. Terlihat pada gambar II.2.



**Gambar II.2. Lambang *Entity***

**Sumber : (Kusrini ; 2007 : 21)**

2. *Atribut*, informasi yang diambil tentang sebuah *entitas*, hanya digunakan oleh organisasi yang dimasukkan dalam model, nama atribut harus merupakan kata benda, kadang nama *entitas* diletakkan di depan nama *atribut* untuk ketelitian. Terlihat pada gambar II.3.

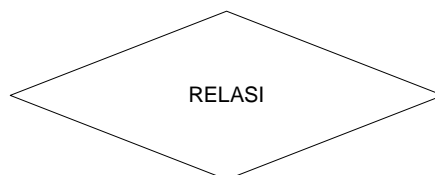


**Gambar II.3. Lambang *Atribut***

**Sumber : (Kusrini ; 2007 : 22)**

3. *Identifier*, satu atau lebih atribut dapat menjadi identifier entitas, yang secara unik mengidentifikasi setiap anggota dari entitas, *identifier* gabungan terdiri dari beberapa atribut, bisa saja artifisial, seperti membuat nomor ID, dan tidak akan dikembangkan sampai fase desain.
4. *Relationship*, hubungan antar entitas, entitas pertama dalam *relationship* disebut entitas induk, entitas kedua disebut entitas anak. *Relationship*

harus memiliki nama berupa kata kerja, *relationship* berjalan dua arah. Terlihat pada gambar II.4.



**Gambar II.4. Lambang Relasi**  
**Sumber : (Kusrini ; 2007 : 21)**

5. *Kardinalitas*, mengacu pada berapa kali *instance* dari suatu entitas.
6. *Modalitas*, mengacu apakah suatu *instance* dari entitas anak dapat ada tanpa suatu telasi dengan *instance* induk atau tidak. (Hanif Al Fatta ; Analisa dan Perancangan Sistem Informasi ; 2007: 121-127).

## II.10. MySQL

*MySQL* merupakan *software* RDBMS (*Relation Database Management System*) atau *server database* yang dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak *user* (*multi-user*), dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi-threaded*).

Saat ini, *MySQL* banyak digunakan di berbagai kalangan untuk melakukan penyimpanan dan pengolahan data. Mulai dari kalangan akademis sampai ke industri, baik industri kecil, menengah ataupun besar.

Lisensi *MySQL* terbagi menjadi dua. Anda dapat menggunakan *MySQL* sebagai produk *open source* di bawah *GNU General Public License* (gratis) atau

dapat membeli lisensi dari versi komersialnya. *MySQL* versi komersial tentu memiliki nilai lebih atau kemampuan yang tidak disertakan pada versi gratis.

Pada kenyataannya, untuk keperluan industri menengah ke bawah, versi gratis masih dapat digunakan dengan baik. Beberapa contoh aplikasi yang menggunakan *MySQL* adalah : *Joomla* ([www.joomla.org](http://www.joomla.org)), *Wordpress* ([www.wordpress.com](http://www.wordpress.com)), *MyBB* ([www.mybb.com](http://www.mybb.com)), *phpBB* ([www.phpbb.com](http://www.phpbb.com)), dan masih banyak sekali yang lainnya. (Budi Raharjo ; Belajar Otodidak Membuat Database Menggunakan MySQL ; 2011 : 21-22)

Beberapa alasan menggunakan *MySQL* sebagai *server database* untuk aplikasi-apliksi yang dikembangkan :

1. Fleksibel
2. Performa tinggi
3. Lintas platform
4. Gratis
5. Proteksi data yang handal
6. Komunitas luas (Budi Raharjo ; 2011 : 22-23)