

## **BAB 2**

### **LANDASAN TEORI**

#### **2.1. *Student Centered Learning***

*Student Centered Learning* (SCL) adalah sebuah pendekatan dalam edukasi yang lebih mengutamakan pada kebutuhan dari pelajarnya dari pada bagian-bagian lain yang termasuk dalam proses edukasi seperti guru. SCL ini memiliki potensi untuk mendorong pelajar lebih aktif, mandiri, sesuai dengan irama belajarnya masing-masing, sesuai dengan perkembangan yang berjalan. SCL bukan berarti "pelajar belajar sendiri" namun berupa sebuah proses belajar yang mengoptimalkan kemandirian pelajar sebagai pribadi yang dewasa dengan menyeimbangkan kemampuan kognisi dan emosi (Sumber: [http://en.wikipedia.org/w/index.php?title=Student-centered\\_learning](http://en.wikipedia.org/w/index.php?title=Student-centered_learning)). Hal-hal yang membantu proses belajar dalam *e-learning* adalah kerja kelompok, diskusi, presentasi, dan menulis hasil penelitian maupun makalah.

Karakteristik *Student Centered Learning* yaitu:

- Pelajar aktif dalam belajar
- Pelajar membuat keputusan tentang apa yang akan dia pelajari dan bagaimana mewujudkannya
- Pelajar menambah pengetahuan dan kemampuan baru berdasarkan kemampuannya saat ini dan dibantu oleh pengajar dengan infrastruktur
- Pelajar melakukan kerja sama dengan pelajar lain yang memiliki tujuan yang sama

- Pelajar mempresentasikan hasil belajarnya di depan pengajar maupun terhadap pelajar-pelajar yang lainnya
- Pengajar membantu pelajar dalam memonitor perkembangan pelajar dan bersedia menjawab pertanyaan-pertanyaan pelajar yang sedang mengalami kesulitan.
- Pelajar menentukan strategi belajarnya sendiri.

## 2.2. **E-Learning**

*Electronic Learning* atau *e-learning* adalah pendekatan pembelajaran yang menggunakan elektronik sebagai media penyampaiannya, yang di masa sekarang ini media tersebut tidak lepas dari penggunaan teknologi komputer. *E-learning* dapat berupa *CD-ROM based (stand alone)*, *Network-based*, *Intranet-based*, atau *Internet-based*. *E-learning* mencakup *text*, *video*, *audio*, *animation*, dan *virtual environment* (Sumber: <http://en.wikipedia.org/wiki/e-learning> dan <http://www.howstuffworks.com/elearning1.htm>).

Keuntungan menggunakan *e-learning* adalah fleksibilitas yang tinggi, kenyamanan, dan kemampuan untuk bekerja sesuai dengan kecepatan dan kemampuan diri sendiri. *E-classes* bergerak secara *asynchronous* yang berarti pelajar dapat berpartisipasi dan menyelesaikan belajarnya sesuai dengan waktu yang dimilikinya. Keuntungan lainnya adalah kemampuan untuk berdiskusi dengan pelajar lainnya yang terpisah jarak yang jauh seperti pelajar dari luar negeri, kemampuan adaptasi yang lebih tinggi terhadap kebutuhan pelajar, lebih banyak variasi dalam pengalaman belajar dengan menggunakan multimedia dan presentasi yang tidak

berbasis kata-kata dalam menyampaikan materi. Rekaman *video* dan *audio* memungkinkan pelajar dapat memutarinya kembali sesuai dengan yang diinginkan.

*E-learning* yang dibahas di sini bukan berarti menggantikan suatu proses belajar yang sudah ada, namun merupakan suatu fasilitas yang dapat digunakan untuk pengayaan terhadap materi maupun pengalaman lainnya bagi pelajar yang ada di dalamnya.

### **2.2.1. Student Centered E-Learning**

Walaupun hasil penelitian membuktikan bahwa pendekatan pembelajaran secara *student-centered style* dapat meningkatkan kualitas mental dan sosial pelajar yang menyebabkan pencapaian hasil akademik yang baik, namun pendekatan tersebut memiliki keterbatasan waktu dan ruang. Waktu untuk berpikir, berkomunikasi, membangun organisasi dan kontribusi, menyediakan materi, menyiapkan instruktur yang baik untuk menjadikan sebuah perantara yang baik dalam pembelajaran memerlukan banyak biaya termasuk waktu. Biaya-biaya tersebut dapat ditekan dengan menggunakan sebuah media baru yaitu dengan memanfaatkan *e-learning* seperti pada bagian dibawah ini:

- Bahan-bahan dan materi-materi dapat diakses melalui *internet* atau melalui sistem *E-Learning*. Catatan kuliah, daftar buku, link ke *website* profesional, dan halaman projek dapat dibuat secara elektronik. Materi-materi dapat di-*upload* dan diperbarui setiap saat. Pelajar dapat membagikan hasil pencariannya ke seluruh grup, dengan menambahkan komentar pada isi, kualitas,

dan mudah didapatkan. Hal ini akan menghemat biaya dan waktu dalam pembelajaran terutama dengan membagikan fotokopi materi, pembaharuan versi materi, pencarian material untuk keperluan pribadi.

- *Internet* sebagai penyedia pengetahuan. Khususnya pelajar jurusan ilmu komputer dan seluruh pelajar secara umum dapat menggunakan *internet* untuk mengeksplorasi dan *open learning*. Paradigma belajar seperti ini cocok dengan pendekatan cara belajar *student-centered* karena pelajar bebas untuk mencari bahan-bahan melalui *website* dan dapat bekerja sama untuk mencapai tujuan mereka.
- Situs bahan kuliah. Suatu media yang dapat mengakses seluruh materi dan informasi yang diharapkan membuat komunikasi dalam organisasi menjadi lebih efisien.
- Komunikasi dan partisipasi. Dalam pendekatan *student-centered* yang pelajarnya tertarik dan berpartisipasi dalam semua aspek pembelajaran yang membutuhkan komunikasi yang intensif. Hal-hal tersebut diwujudkan dengan media-media elektronik seperti forum diskusi, *blog*, *student's workspaces*, *course homepage*, dan lain-lain.

### **2.2.2. *Dynamic E-Learning Content Generator***

*Dynamic E-Learning Content Generator* adalah suatu modul yang berfungsi untuk menghasilkan *content/isi* dari sebuah *E-Learning*. *File* atau *source* yang didapatkan dari pengguna akan diolah dan dihasilkan suatu *file* hasil *generate* yang akan dibaca oleh aplikasi *e-learning* sebagai *data source*-nya. *Data*

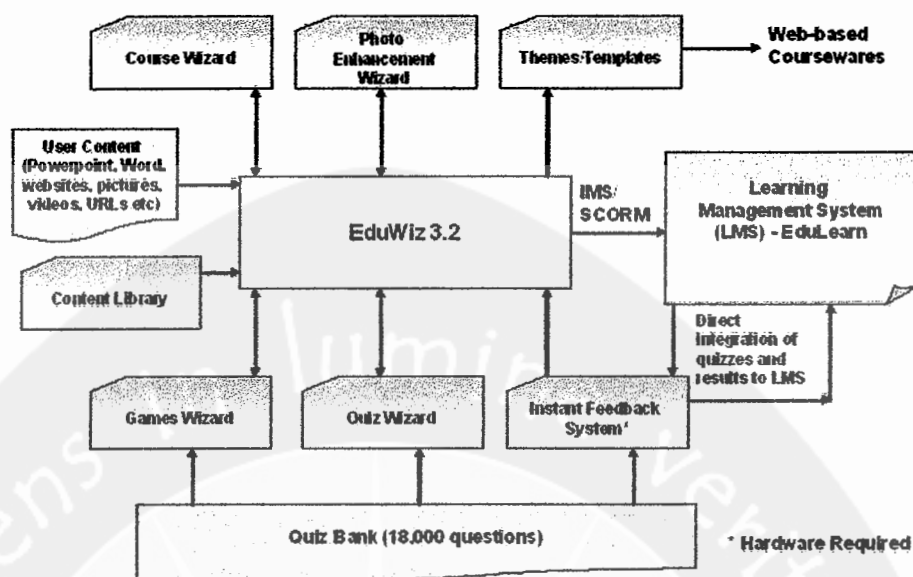
source yang dihasilkan dapat berupa *file* (xml, text, dan lain-lain) atau ditanamkan pada *database*.

### 2.3. Eduwiz

Eduwiz adalah sebuah perangkat lunak yang menghasilkan media *e-learning* berbasis *web* tanpa membutuhkan kemampuan multimedia atau teknis. Eduwiz dibuat oleh perusahaan AskNLearn yang didirikan di Singapura.

Eduwiz memiliki empat buah *tools* yaitu *Course Wizard*, *Games Wizard*, *Quiz Wizard*, dan *Photo Enhancement Wizard*. *Course Wizard* membantu pengguna dalam menciptakan *courseware* dalam langkah mudah. *Game Wizard* membuat permainan berbasis *Flash* dengan cepat dan mudah. *Automatic Image Enhancement Wizard* memperbaiki masalah warna dan *contrast* pada gambar dan sangat diperlukan pada *photo galleries*. *Quiz Wizard* mendukung enam jenis *quiz* dan meng-export *quiz* ke suatu *Learning Management System* (LMS). Setelah membuat dan menyatukan *content* dengan bantuan *content tools*, pengguna memiliki enam *themes* berbasis *flash* untuk menyajikan *courseware*-nya.

Eduwiz terintegrasi dengan sebuah *response system* (*Instant Feedback System*) yang mendapatkan secara cepat *feedback* dari peserta *quiz*, *survey*, atau analisis kecepatan mengajar. *Instant Feedback System* terdiri dari sebuah *infra-red receiver* dan 45 *infra-red controllers*. Terhubung dengan PC atau *Notebook* melalui *USB port*, maka menyediakan kemampuan *portable* dan dapat dengan mudah dipindahkan ke kelas-kelas lainnya.



Gambar 2.1 Arsitekur Eduwiz

(Sumber: [www.asknlearn.com/eduwiz/](http://www.asknlearn.com/eduwiz/))

#### 2.4. Dependency Injection

*Dependency Injection* adalah salah satu teknik pemrograman untuk mengurangi "dependency" (ketergantungan) di antara komponen-komponen penyusun suatu program dan meningkatkan kemampuan "reusability" dari komponen tersebut (Sumber: <http://www.martinfowler.com/>). Sebagai contoh jika sebuah komponen menggunakan *sub-component*, penggunaan kembali komponen tersebut dengan *sub-component* yang berbeda akan sangat sulit. Contoh konkrit adalah jika sebuah program menggunakan suatu *database* dari *DBMS* A sebagai tempat penyimpanan datanya, lalu di kemudian hari program tersebut ingin diubah dengan menggunakan *DBMS* maka yang terjadi adalah mengubah (membongkar) kode program.

Ide dasar dari *dependency injection* adalah memindahkan sebagian kode yang menginstansi *sub-component* ke sebuah *configuration file* (biasanya *XML*

*file*) dan menggunakan sebuah *framework* untuk mewujudkannya.

## 2.5. **Framework**

*Framework* adalah sebuah kumpulan dari modul-modul pembangun perangkat lunak yang umum dan tak dapat dieksekusi, di mana *programmer* dapat menggunakannya, menambahnya, atau mengkonfigurasi untuk solusi atau program/aplikasi yang spesifik. Dengan *framework*, *developer* tidak harus memulai dari awal setiap aplikasi dibuat. *Framework* terdiri dari kumpulan modul atau objek sehingga desain dan kode dari *framework* dapat digunakan kembali (Sumber: <http://www.wikipedia.org/framework>).

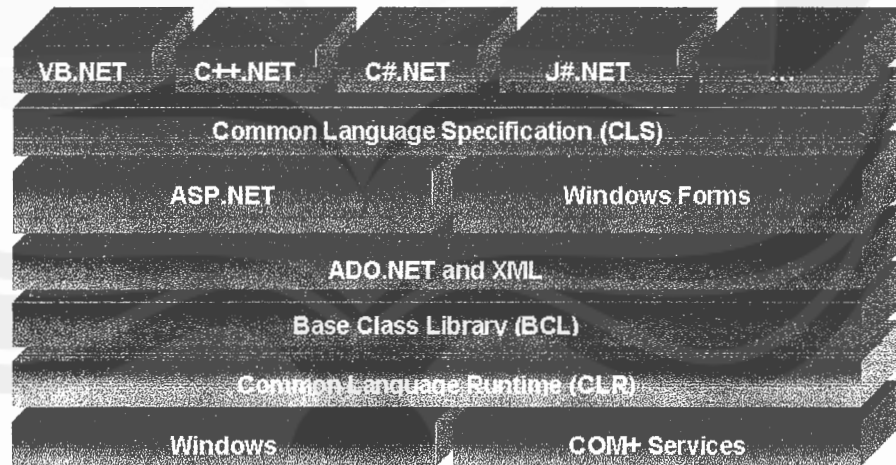
### 2.5.1. **.NET Framework**

*Framework .NET* adalah lingkungan untuk membangun, menyebarkan/*deploying*, dan menjalankan aplikasi dan layanan berbasis web. *.Net Framework* disusun oleh dua komponen utama, yaitu *Common Language Runtime (CLR)* dan *.NET Framework Class Library* termasuk aplikasi *Console*, *Windows GUI*, *ASP.NET*, layanan web *XML* dan layanan *Windows*. (Sumber: <http://www.microsoft.com/>)

*Visual Studio.NET* dibangun menggunakan fondasi *.NET Framework*. *.NET Framework* menyediakan lingkungan yang cerdas, mudah dikembangkan untuk membangun, menyebarkan dan menjalankan aplikasi dan layanan web *XML* yang terdistribusi. Dalam istilah yang mudah, *.NET Framework* memisahkan *platform* sistem operasi menjadi dua lapisan, yaitu lapisan pemrograman dan lapisan eksekusi.

Tujuan dari *.NET Framework* adalah:

1. Menyediakan lingkungan pemrograman berorientasi objek, apakah kode objek disimpan dan dijalankan secara lokal, dijalankan secara lokal tetapi disebarakan melalui *internet* atau dijalankan secara *remote* (dijalankan dari suatu tempat).
2. Menyediakan lingkungan untuk menjalankan suatu kode yang menjamin keamanan saat kode dijalankan.
3. Menyediakan lingkungan untuk menjalankan suatu kode yang dapat mengeliminasi masalah performa dari lingkungan *scripted* dan *interpreted*.
4. Menyediakan lingkungan untuk menjalankan suatu kode yang meminimalkan konflik pada *deployment* dan *versioning* perangkat lunak.
5. Menyatukan model-model pemrograman dengan didukung oleh banyak bahasa dan membuat berbagai tipe aplikasi.



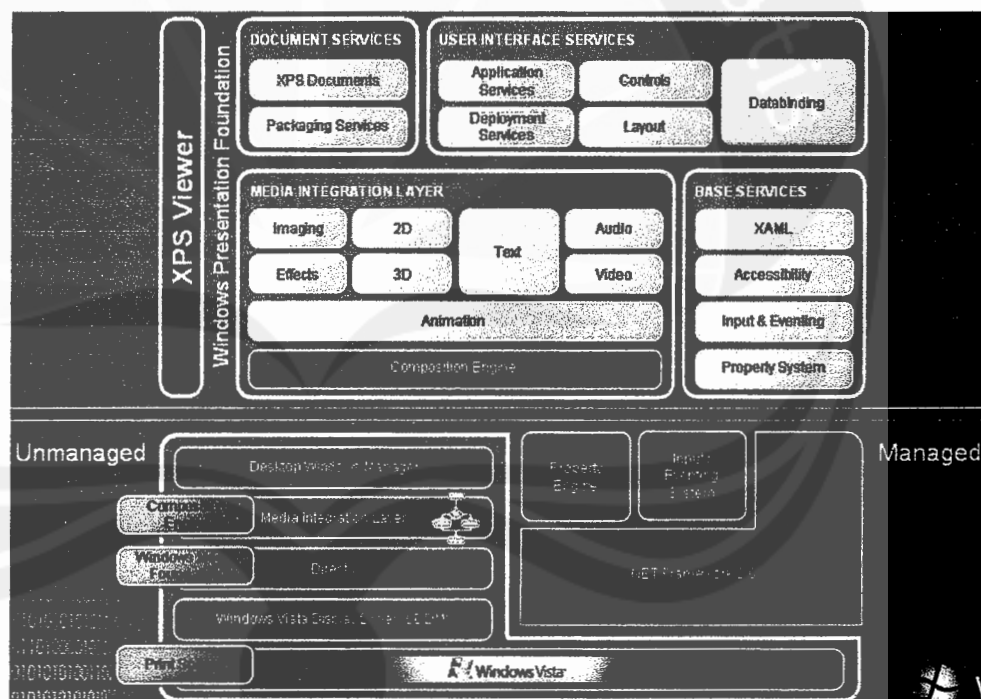
Gambar 2.2 Arsitekur .NET Framework  
(Sumber: [www.microsoft.com/](http://www.microsoft.com/))

### 2.5.2. Windows Presentation Foundation (WPF)

Windows Presentation Foundation (WPF) sebelumnya disebut *Avalon* adalah salah satu bagian atau subsistem



dari .NET Framework 3.0 yang khusus menangani *Graphical User Interface* (GUI). WPF menyebabkan suatu *programming* lebih konsisten dalam membangun suatu aplikasi dengan memisahkan secara jelas antara *User Interface (UI) code* dan *business logic code*. Sebuah aplikasi yang menggunakan WPF dapat dijalankan pada *desktop* maupun *web* karena menggunakan suatu standar yang sama dalam *me-render* komponen UI yaitu dengan menggunakan XAML (*Extensible Application Markup Language*). WPF menyediakan kontrol-kontrol, *layout*, objek 2D, objek 3D, dan *visual effects* yang lebih kaya dan dapat digunakan secara dinamis.



Gambar 2.3 Arsitekur WPF  
(Sumber: [www.microsoft.com/netfx3/](http://www.microsoft.com/netfx3/))

## **2.6. Tools dan Teknologi yang Digunakan**

### **2.6.1. Teknologi .NET**

Microsoft Visual Studio .NET adalah sebuah *platform* untuk membangun, menjalankan, dan meningkatkan generasi lanjut dari aplikasi terdistribusi. Microsoft Visual Studio .NET memperluas *client*, *server*, dan *service-service* yang terdiri atas:

1. Sebuah model pemrograman yang memungkinkan *developer* membangun aplikasi dan layanan *web XML*.
2. Sekumpulan *XML Web services* seperti Microsoft .NET *My Services*, yang membantu *developer* menghasilkan aplikasi yang sederhana dan terpadu.
3. Serangkaian *server* termasuk Microsoft Windows Server 2003, Microsoft SQL Server, dan Microsoft BizTalk Server yang terintegrasi, untuk menjalankan, mengoperasikan, dan mengelola aplikasi dan layanan berbasis *web*.
4. *Tool-tool* pengembang yang menyediakan IDE (*Integrated Development Environment*) untuk memaksimalkan produktivitas pengembangan menggunakan *.NET Framework*.
5. Piranti lunak *client* seperti Windows XP, Windows CE, dan Microsoft Office XP yang membantu pengembang untuk menyebarkan dan mengelola aplikasinya.

### **2.6.2. Visual C#.NET 2005**

C# adalah bahasa pemrograman baru yang diciptakan oleh Microsoft, dikembangkan dibawah kepemimpinan Anders Hejlsberg. Anders dikenal sebagai salah satu pencipta *IDE Pascal*, yakni *Delphi*. Latar belakang

Anders kemudian disatukan dengan aspek teknologi C++ yang telah ada, sehingga hadirlah C# sebuah bahasa yang canggih.

Definisi C# dari Microsoft adalah sebagai berikut:

*"C# is a simple, modern, object oriented and type-safe programming language derived from C and C++. C# is firmly planted in the C and C++ family tree of languages, and will immediately be familiar to C and C++ programmers. C# aim to combine the high productivity of Visual Basic and the raw power of C++"*

Ada dua kata kunci pada definisi di atas, yakni *high productivity* dan *raw power C++*. C# hadir bukan saja sebagai *official language* bagi .NET, dia juga hadir sebagai alternatif bagi para pengembang yang berasal dari C, C++, atau bahkan Java tetapi hadir dengan kemudahan *Visual Basic*. Sehingga pengembang dapat dengan mudah mengembangkan suatu solusi sistem informasi dengan konsep dan paradigma yang modern di bawah naungan teknologi .NET.