

## **BAB 2**

### **LANDASAN TEORI**

#### **2.1 Pengertian E-Application**

*E-application* adalah sebuah aplikasi komputer berbasis jaringan yang didesain untuk menangani aktivitas-aktivitas yang ada di dalam dunia paralel. Aplikasi ini berjalan di dalam lingkungan dunia jaringan. Untuk dapat menangani aktivitas virtual yang ada, aplikasi ini harus mengerti karakteristik dari dunia yang sedang direpresentasikannya.

#### **2.2 Konsep Analisis dan Perancangan**

##### **2.2.1 Pengertian Analisis**

Menurut Jogiyanto (1995, p129) analisis dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikannya.

##### **2.2.2 Pengertian Perancangan**

Menurut Jogiyanto (1999, p179) perancangan mempunyai 2 maksud, yaitu : untuk memenuhi kebutuhan kepada pemakai sistem dan untuk memberikan gambaran yang jelas kepada pemrogram komputer dan ahli-ahli teknik lainnya yang terlibat.

## **2.3 Internet**

### **2.3.1 Pengertian Internet**

Menurut Turban, Rainner dan Potter (2006,p674) internet adalah jaringan besar yang menghubungkan jaringan komputer dari bisnis, organisasi, lembaga pemerintah, dan sekolah di seluruh dunia, dengan cepat, langsung, dan hemat. Contoh-contoh dari jaringan komputer ini mencakup sistem komputer universitas, sistem komputer perusahaan, atau sistem komputer rumah sakit. Sistem komputer yang berpartisipasi , yang disebut *node*, mencakup PC, LAN (*local area network*), basis data, dan *mainframe*. Di internet, *node* dapat mencakup beberapa jaringan dari suatu organisasi , yang mungkin dihubungkan oleh WAN (*wide area network*). Sebagai satu atau beberapa jaringan, internet memungkinkan orang untuk mengakses data di organisasi lainnya dan untuk berkomunikasi, bekerja sama, dan bertukar informasi di hampir seluruh dunia. Jadi, internet telah menjadi keharusan dalam melaksanakan bisnis modern.

### **2.3.2 Sejarah Internet**

Menurut Janner Simarmata (2006,p282-283), sebelum internet ada, ARPAnet (*US Defense Advanced Research Projects Agency*) atau Departemen Pertahanan Amerika pada tahun 1969 membuat jaringan komputer yang tersebar untuk menghindarkan yang terjadinya informasi terpusat, yang apabila terjadi perang dapat mudah dihancurkan. Jadi, bila satu bagian dari sambungan network terganggu oleh serangan musuh,

jalur yang melalui sambungan itu secara otomatis dipindahkan ke sambungan lainnya. Setelah itu, internet digunakan oleh kalangan akademis (UCLA) untuk keperluan penelitian dan pengembangan teknologi. Baru setelah itu, pemerintah Amerika Serikat memberikan izin ke arah komersil pada awal tahun 1990.

### 2.3.3 World Wide Web (WWW)

Menurut Turban, Rainer dan Potter (2006,p680), *World Wide Web* (Web, WWW, atau W3) adalah sistem standar yang diterima secara universal untuk menyimpan, menelusuri, memformat, dan menampilkan informasi melalui arsitektur *client/server*. Web bisa menerima semua jenis informasi digital, termasuk teks, hipermedia, grafis, dan suara. Web menggunakan antarmuka pengguna grafis, sehingga sangat mudah digunakan. Teknologi *World Wide Web* diciptakan oleh Timothy Berners-Lee, yang pada tahun 1989 mengusulkan jaringan global dari dokumen hiperteks yang akan memungkinkan para peneliti fisika bekerja sama.

Web didasari oleh bahasa hiperteks standar yang disebut ***HyperText Markup Language (HTML)***, yang memformat dokumen dan memadukan link hiperteks dinamis ke dokumen-dokumen lainnya yang disimpan di dalam komputer.

Untuk mengakses situs Web, pengguna harus menentukan ***Uniform Resource Locator (URL)***, yang mengarahkan ke alamat dari sumber tertentu di Web. Misalnya, URL untuk Microsoft adalah <http://www.microsoft.com>. **HTTP** adalah singkatan dari **HyperText**

**Transport Protocol**, yang merupakan standar komunikasi yang digunakan untuk mentransfer halaman di bagian WWW di internet. HTTP mendefinisikan bagaimana pesan diformat dan dikirim.

#### **2.3.4 Transmission Control Protocol/Internet Protocol (TCP/IP)**

TCP/IP adalah standar komunikasi data yang digunakan oleh komunitas internet dalam proses tukar-menukar data dari satu komputer ke komputer lain di dalam jaringan Internet. Protokol ini tidaklah dapat berdiri sendiri, karena memang protokol ini berupa kumpulan protokol (*protocol suite*). Protokol ini juga merupakan protokol yang paling banyak digunakan saat ini. Data tersebut diimplementasikan dalam bentuk perangkat lunak (*software*) di sistem operasi. Istilah yang diberikan kepada perangkat lunak ini adalah TCP/IP stack.

#### **2.3.5 Web Server**

Program komputer yang bertanggung jawab untuk menerima permintaan HTTP dari *client* dan menyediakan respon HTTP beserta data-data tambahannya.

#### **2.3.6 Web Browser**

Menurut Turban, Rainer dan Potter (2006,p681), *Web browser* merupakan aplikasi piranti lunak yang umumnya digunakan oleh pengguna untuk mengakses web. *Browser* menyediakan tampilan grafis yang memungkinkan pengguna untuk menunjuk dan meng-klik bagian yang

diinginkan di web. *Web browser* menjadi sarana akses universal karena mengirimkan antarmuka yang sama pada semua sistem operasi yang dijalankan.

## **2.4 Rekayasa Piranti Lunak**

### **2.4.1 Definisi Piranti Lunak**

Menurut Roger S. Pressman (2005, p10) piranti lunak dapat didefinisikan sebagai berikut :

1. Instruksi-instruksi (program komputer) yang jika dieksekusi akan memberikan hasil yang diharapkan.
2. Struktur data yang memungkinkan program dapat melakukan manipulasi terhadap informasi.
3. Dokumen yang menjelaskan informasi serta penggunaan program.

### **2.4.2 Karakteristik Piranti Lunak**

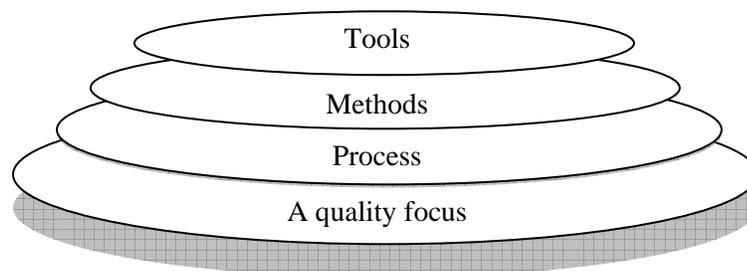
Menurut Pressman (2005, p37) piranti lunak memiliki ciri yang berbeda dari perangkat keras, yaitu :

1. Piranti lunak dibangun dan dikembangkan, tidak dibuat dalam bentuk yang klasik.
2. Piranti lunak tidak pernah usang.
3. Sebagian besar piranti lunak dibuat secara *custom-built*, serta tidak dapat dirakit dari komponen yang telah ada.

### 2.4.3 Definisi Rekayasa Piranti Lunak

Rekayasa piranti lunak (RPL) atau dalam bahasa Inggris *Software Engineering* atau (*SE*) adalah satu bidang profesi yang mendalami cara-cara pengembangan perangkat lunak termasuk pembuatan, pemeliharaan, manajemen organisasi pengembangan perangkat lunak dan sebagainya. Sedangkan menurut Fritz Bauer [NAU69], rekayasa piranti lunak adalah pembentukan dan penggunaan prinsip-prinsip rekayasa piranti lunak guna memperoleh perangkat lunak yang ekonomis, handal serta bekerja efisien pada proses penerapannya.

Rekayasa piranti lunak merupakan sebuah teknologi yang terdiri dari 4 lapisan yaitu alat bantu (*tools*), metode (*methods*), proses (*process*), dan fokus kualitas (*a quality focus*) (Pressman, 2001, p21)



**Gambar 2.1** *Software Engineering Layers*

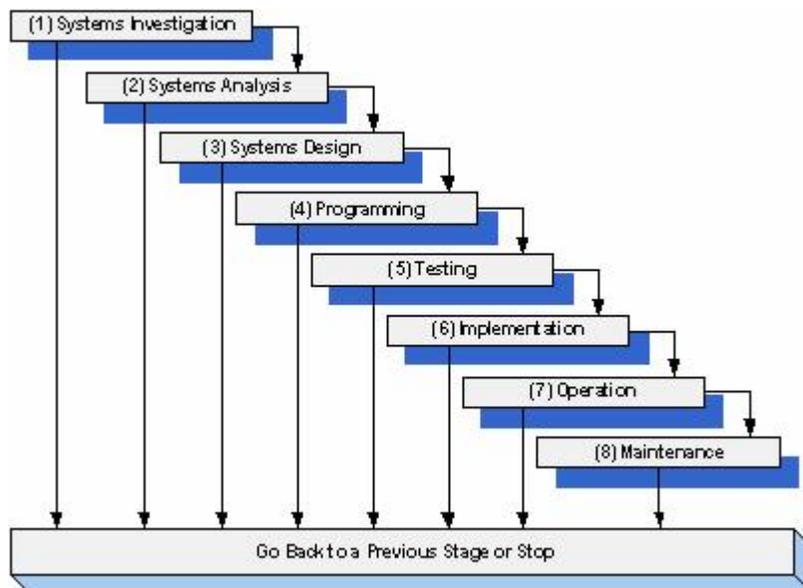
Batu landasan yang menopang rekayasa piranti lunak merupakan berfokus kepada kualitas. Fondasi untuk rekayasa piranti lunak merupakan suatu bentangan proses. Proses – proses ini adalah perekat yang menjaga bentangan – bentangan teknologi secara bersama – sama dan memungkinkan perkembangan perangkat lunak komputer yang tepat

waktu dan rasional. Proses – proses rekayasa piranti lunak membatasi kerangka kerja untuk serangkaian area proses kunci (*key process key*) yang harus dibangun demi keefektifan penyampaian teknologi pengembangan piranti lunak. Area proses kunci ini membentuk dasar bagi kontrol manajemen proyek piranti lunak serta membangun konteks dimana metode teknis diaplikasikan, produk usaha (model, dokumen, data, laporan, *form*) dihasilkan, fondasi dibangun, kualitas dijamin, dan perubahan diatur secara rapi. Metode – metode rekayasa piranti lunak memberikan teknik untuk membangun perangkat lunak. Metode – metode itu menyangkut analisis kebutuhan, konstruksi program, desain, pengujian, dan pemeliharaan. Alat – alat bantu rekayasa piranti lunak memberikan topangan yang bersifat otomatis maupun semi – otomatis pada proses – proses dan metode – metode yang ada. Ketika alat – alat bantu diintegrasikan sehingga informasi yang diciptakan oleh satu alat bantu bisa digunakan oleh yang lain, sistem untuk menopang perkembangan perangkat lunak yang disebut dengan *Computer Aided Software Engineering* (CASE). CASE menggabungkan perangkat lunak, perangkat keras, dan database rekayasa piranti lunak untuk menciptakan lingkungan rekayasa piranti lunak yang analog dengan CAD/CAE (*Computer Aided Design/Engineering*) untuk perangkat keras.

#### **2.4.4 System Development Life Cycle (SDLC)**

Menurut Turban, Rainer dan Potter (2006,p690), *System Development Life Cycle* (SDLC) adalah kerangka kerja tradisional yang

terstruktur dan yang digunakan untuk proyek TI besar, serta terdiri atas berbagai proses berurutan untuk mengembangkan sistem informasi. SDLC terdiri dari delapan tahap, meliputi : penelitian, analisis, desain, pemrograman, pengujian, implementasi, operasi, dan pemeliharaan. Di masa lalu, para pengembang menggunakan **pendekatan air terjun** (*waterfall approach*). *Waterfall approach* merupakan pendekatan SDLC dengan berbagai pekerjaan di suatu tahap diselesaikan terlebih dulu sebelum pekerjaan dilanjutkan ke tahap berikutnya.



**Gambar 2.2 Delapan Step dalam SDLC (Turban, Rainer, Potter, 2005, p490)**

Tahap-tahap SDLC adalah sebagai berikut :

### 1. Penelitian sistem (*System investigation*)

Penelitian sistem dimulai dengan masalah bisnis (atau peluang bisnis). Masalah (dan peluang) sering kali tidak hanya

membutuhkan pemahaman mengenainya dari sudut pandang internal, tetapi juga sebagai sudut Pandang mitra organisasional (pemasok atau pelanggan) yang akan melihatnya. Pekerjaan berikutnya dalam penelitian sisitem adalah studi kelayakan, yakni penelitian yang mengukur probabilitas keberhasilan proyek yang diusulkann dan memberikan penilaian kasar mengenai kelayakan proyek tersebut. Berikutnya dilanjutkan dengan kelayakan teknis, yakni penilaian apakah komponen peranti keras, peranti lunak dan komunikasi dapat dikembangkan dan / atau diperoleh untuk mengatasi suatau masalah bisnis. Selanjutnya dalam tahap ini adalah tahap kelayakan ekonomis, yakni penilaian apakah suatu proyek adalah risiko keuangan yang dapat ditoleransi dan apakah perusahaan dapat menanggung pengeluaran dan waktu yang dibutuhkan untuk menyelesaikannya. Berikutnya adalah tahap kelayakan organisasional, yang berkaitan dengan kemampuan perusahaan untuk menerima proyek yang diusulkan. Yang terakhir pada tahap ini adalah kelayakan berperilaku, yakni penilaian berbagai isu manusia yang dilibatkan dalam proyek yang diusulkan, termasuk penolakan atas perubahan dan kebutuhan keahlian serta pelatihan.

## 2. Analisis sistem (*System analysis*)

Analisis sistem adalah proses mempelajari berbagai masalah bisnis yang direncanakan perusahaan dapat diatasi melalui sistem informasi. Tahap ini menentukan masalah bisnisnya, mengidentifikasi berbagai penyebabnya, menspeksifikasi solusi, Dan mengidentifikasi kebutuhan informasi yang harus dipenuhi oleh solusi tersebut. Pemahaman atas masalah bisnis membutuhkan pemahaman atas berbagai proses yang dilibatkan. Ini sering kali dapat cukup rumit dan saling terkait.

## 3. Desain sistem (*System design*)

Desain sistem menjelaskan bagaimana sistem tersebut akan menjalankan sistem tersebut. Hasil dari tahap desain sistem adalah desain teknis yang menspesifikasikan berbagai hal berikut :

- a. Output dan input sistem serta antarmuka pengguna
- b. Peranti keras, peranti lunak, basis data, telekomunikasi, personel, dan prosedur
- c. Bagaimana berbagai komponen ini diintegrasikan

Desain sistem melintasi dua aspek sistem utama baru : desain logis sistem yang menyatakan apa yang akan dilakukan oleh sistem tersebut, dengan menggunakan spesifikasi abstrak , dan desain fisik sistem yang menyatakan bagaimana sistem baru

tersebut akan melakukan fungsinya, dengan spesifikasi fisik yang sesungguhnya.

#### **4. Pemrograman (*Programming*)**

Pemrograman melibatkan penerjemahan spesifikasi desain ke dalam kode komputer. Dalam proyek tim pemrograman akan dibentuk. Tim ini sering kali meliputi para pengguna berbagai area fungsional untuk membantu programmer fokus pada masalah bisnis yang ditanganani. Dalam usaha untuk menambahkan presisi (atau suatu keseragaman) dalam proses pemrograman, para programmer menggunakan teknik pemrograman terstruktur. Teknik ini memperbaiki arus logis program dengan memecah kode-kode komputer menjadi modul, yang merupakan beberapa bagian dari kode. Struktur modular ini memungkinkan pengujian yang lebih efisien dan efektif, karena tiap modul dapat diuji sendiri-sendiri.

#### **5. Pengujian (*Testing*)**

Pengujian menyeluruh dan berkelanjutan dilakukan di sepanjang tahap pemrograman. Pemeriksaan pengujian melihat apakah kode komputer akan memberi hasil yang diperkirakan dan diinginkan di bawah kondisi tertentu. Pengujian didesain untuk mendeteksi kesalahan (*bug*) dalam kode komputer. Kesalahan-kesalahan ini terdiri atas dua jenis :

kesalahan sintaks (*syntax error*) dan kesalahan logika (*logic error*). Programmer harus mengikuti arus logika dalam program untuk menentukan sumber kesalahan pada hasilnya.

## **6. Implementasi (*Implementation*)**

Implementasi adalah proses konversi dari sistem lama ke sistem baru. Terdapat empat strategi konversi : paralel, langsung, percobaan awal, dan bertahap. Dalam konversi paralel, sistem lama dan sistem baru beroperasi secara bersamaan untuk jangka waktu tertentu. Dalam konversi langsung, sistem yang lama akan dimatikan ketika sistem yang baru dinyalakan, pada waktu tertentu. Dalam konversi percobaan awal, akan memperkenalkan sistem baru di satu bagian perusahaan untuk percobaan; jika sistem baru itu bekerja baik, maka akan diperkenalkan ke berbagai bagian lain perusahaan. Dalam konversi bertahap, memperkenalkan sistem baru dalam berbagai tahap, hingga keseluruhan sistem dapat beroperasi.

## **7. Operasi (*Operation*)**

Sistem baru akan beroperasi dalam jangka waktu tertentu, hingga (seperti sistem lama yang digantikannya) sistem tersebut tidak lagi memenuhi tujuan perusahaan, begitu operasi sistem baru stabil, audit akan dilakukan selama masa

operasi untuk menilai kemampuan sistem dan menentukan apakah sistem tersebut digunakan dengan benar.

## **8. Pemeliharaan (*Maintenance*)**

Sistem membutuhkan pemeliharaan. Jenis pertama pemeliharaannya adalah *debugging program*, yaitu proses yang terus berlanjut sepanjang umur hidup sistem. Jenis yang kedua adalah pembaruan sistem untuk mengakomodasi berbagai perubahan dalam kondisi bisnis. Berbagai pembaruan dan perbaikan ini biasanya tidak menambahkan fungsi baru tetapi dibutuhkan hanya agar sistem tersebut dapat tetap memenuhi tujuannya. Jenis ketiga dari pemeliharaan adalah menambah fungsi baru ke sistem yaitu, penambahan fitur baru ke sistem yang ada tanpa mengganggu operasinya.

## **2.5 Sistem Basis Data**

### **2.5.1 Basis Data**

Basis data merupakan sekelompok logis berbagai file yang saling berhubungan dan yang menyimpan data serta hubungan antar file tersebut, dapat meniadakan berbagai masalah yang berkaitan dengan lingkungan file tradisional.

### **2.5.2 Sistem Manajemen Basis Data**

Peranti lunak program (atau sekelompok program) yang menyediakan akses ke sebuah basis data. Sistem manajemen basis data (*Database Management System – DBMS*) memungkinkan penyimpanan data di suatu lokasi yang dapat diperbaharui serta ditari, dan menyediakan akses ke data yang disimpan oleh berbagai program aplikasi. DBMS juga menyediakan mekanisme untuk memelihara integritas informasi yang disimpan, mengelola keamanan dan akses pengguna, serta memulihkan informasi ketika sistem gagal. DBMS memberikan alat kepada para pengguna untuk menambah, menghapus, memelihara, menampilkan, mencetak, mencari, memilih, mengurutkan, serta memperbarui data.

### **2.5.3 Relation Database Management System**

*Relational Database Management System (RDBMS)* adalah sebuah program komputer (atau secara lebih tipikal adalah seperangkat program komputer) yang didisain untuk mengatur/memanajemen sebuah basisdata sebagai sekumpulan data yang disimpan secara terstruktur, dan melakukan operasi-operasi atas data atas permintaan penggunanya. Contoh penggunaan DBMS ada banyak sekali dan dalam berbagai bidang kerja, misalnya akuntansi, manajemen sumber daya manusia, dan lain sebagainya.

#### 2.5.4 Normalisasi

Normalisasi adalah metode untuk menganalisis dan mengurangi basis data relasional ke dalam bentuknya yang paling sederhana untuk meminimalkan redundansi, memaksimalkan integritas data, dan mendapat kinerja pemrosesan terbaik.

#### 2.5.5 Entity Relationship Diagram (ERD)

ERD merupakan suatu model untuk menjelaskan hubungan antardata dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi.

ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol. Pada dasarnya ada tiga simbol yang digunakan, yaitu :

a. Entiti

Entiti merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain (Fathansyah, 1999: 30). Simbol dari entiti ini biasanya digambarkan dengan persegi panjang.

b. Atribut

Setiap entitas pasti mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasi isi elemen satu dengan yang lain. Gambar atribut diwakili oleh simbol elips.

c. Hubungan / Relasi

Hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Relasi dapat digambarkan sebagai berikut :

Relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dalam satu basis data yaitu (Abdul Kadir, 2002: 48) :

1. Satu ke satu (*One to one*)

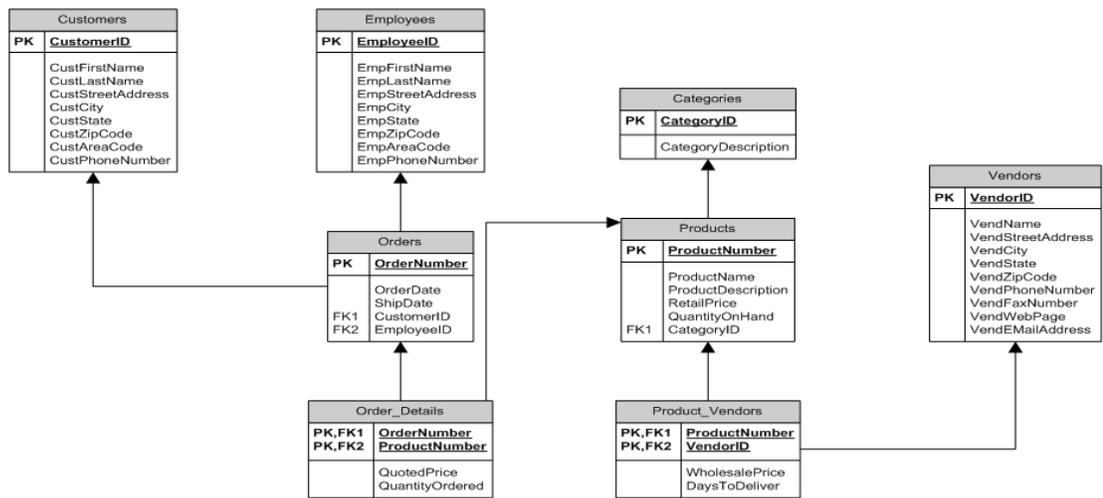
Hubungan relasi satu ke satu yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B.

2. Satu ke banyak (*One to many*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi setiap entitas pada entitas B dapat berhubungan dengan satu entitas pada himpunan entitas A.

3. Banyak ke banyak (*Many to many*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B.



**Gambar 2.3** Contoh *Entity Relationship Diagram* (ERD)

## 2.6 Unified Modelling Language (UML)

### 2.6.1 Pengertian Unified Modelling Language

Menurut Martin Fowler (2004,p1-2), UML adalah kelompok notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO).

UML merupakan standar yang relatif terbuka yang dikontrol oleh *Object Management Group* (OMG), sebuah konsorsium terbuka yang terdiri dari banyak perusahaan. OMG dibentuk untuk membuat standar-standar yang mendukung interoperabilitas, khususnya interoperabilitas

sistem berorientasi objek. OMG mungkin lebih dikenal dengan standar-standar COBRA (*Common Object Request Broker Architecture*).

Kegunaan UML adalah sebagai berikut:

1. UML sebagai bahasa visualisasi digunakan untuk merancang suatu model yang dapat dibaca oleh banyak orang dengan pengertian yang sama.
2. UML bahasa pendefinisian digunakan untuk mendefinisikan dengan rinci seluruh hasil analisis, desain, dan implementasi yang harus dilakukan dalam pengembangan sistem.
3. UML sebagai bahasa dokumentasi digunakan untuk mendokumentasikan arsitektur beserta perinciannya, unsure-unsur yang dibutuhkan dalam pengembangannya. Serta perencanaan dan implementasi proyek secara keseluruhan dengan simbol-simbol yang mudah dimengerti.

UML adalah hanya sebuah notasi yang digunakan untuk menggambarkan diagram-diagram. Pada UML juga ditambahkan tambahan fitur-fitur menarik lainnya yaitu:

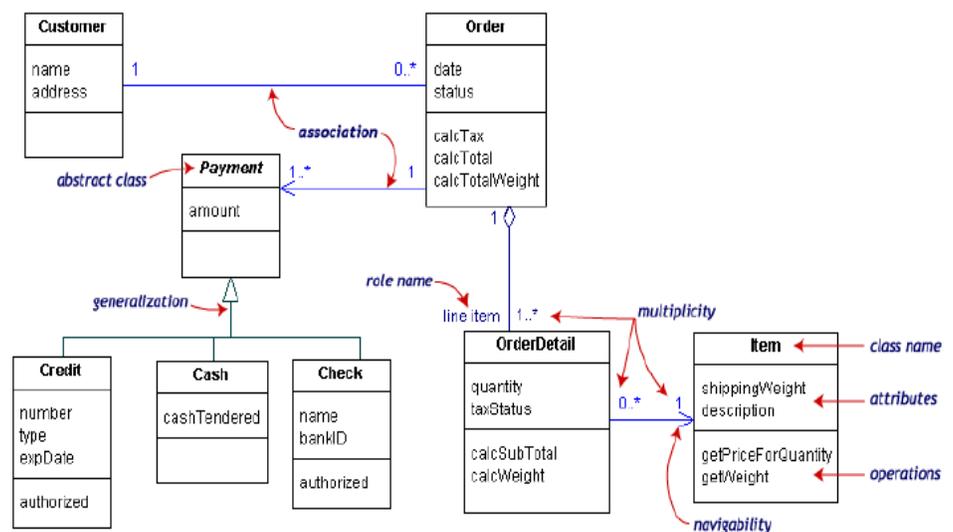
- 1 UML memiliki semantik detil, yang digunakan untuk menjelaskan maksud dari variasi-variasi notasi.
- 2 UML memiliki mekanisme tambahan, yang memungkinkan pendesain *software* untuk mempresentasikan konsep-konsep yang bukan merupakan inti dari UML itu sendiri.

- 3 UML telah diasosiasikan dengan *textual language* yang disebut *Object Constraint Language (OCL)*.

## 2.6.2 Diagram Struktur

### 2.6.2.1 Diagram Kelas

Diagram kelas mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat di antara mereka. Diagram kelas juga menunjukkan properti dan operasi sebuah kelas dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut. UML menggunakan istilah fitur sebagai istilah umum yang meliputi properti dan operasi sebuah kelas.



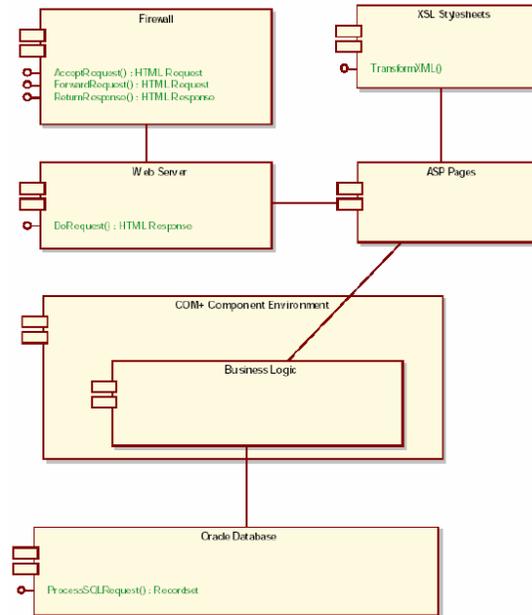
Gambar 2.4 Contoh Diagram Kelas

### **2.6.2.2 Diagram Objek**

Diagram objek merupakan sebuah gambaran tentang objek-objek dalam sebuah sistem pada satu titik waktu. Karena lebih menonjolkan perintah-perintah daripada kelas. Diagram objek lebih sering disebut sebagai sebuah diagram perintah. Diagram objek dapat digunakan untuk menunjukkan sebuah konfigurasi contoh dari objek.

### **2.6.2.3 Diagram Komponen**

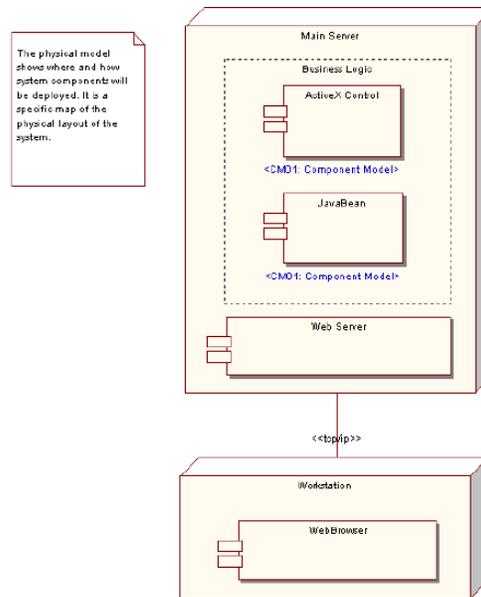
Komponen diagram digunakan pada saat akan memecah sistem menjadi komponen-komponen dan ingin menampilkan hubungan-hubungan mereka dengan antarmuka atau pemecahan komponen menjadi struktur yang lebih rendah. Komponen terhubung melalui antarmuka yang digunakan dan dibutuhkan.



**Gambar 2.5 Contoh Diagram Komponen**

#### 2.6.2.4 Diagram Deployment

Diagram deployment menunjukkan susunan fisik sebuah sistem, menunjukkan bagian perangkat lunak mana yang berjalan pada perangkat keras mana. Diagram deployment sangatlah sederhana.



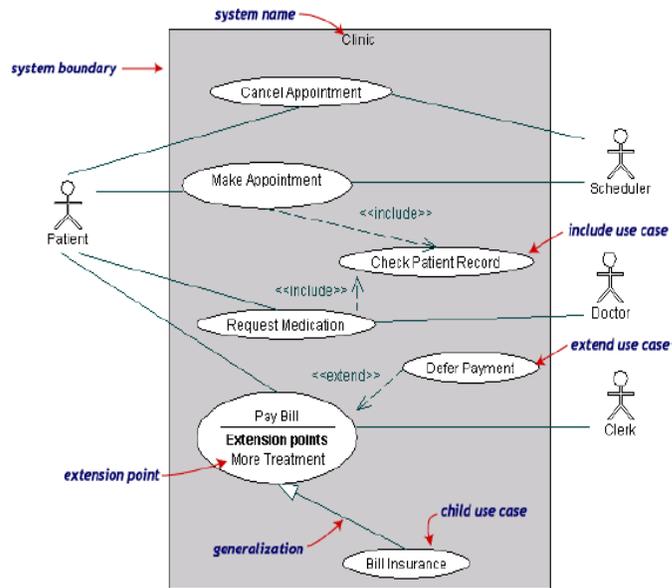
**Gambar 2.6 Contoh Diagram Deployment**

## 2.6.3 Diagram Perilaku

### 2.6.3.1 Diagram Use Case

Use case adalah teknik untuk merekam persyaratan fungsional sebuah sistem. Use case mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan.

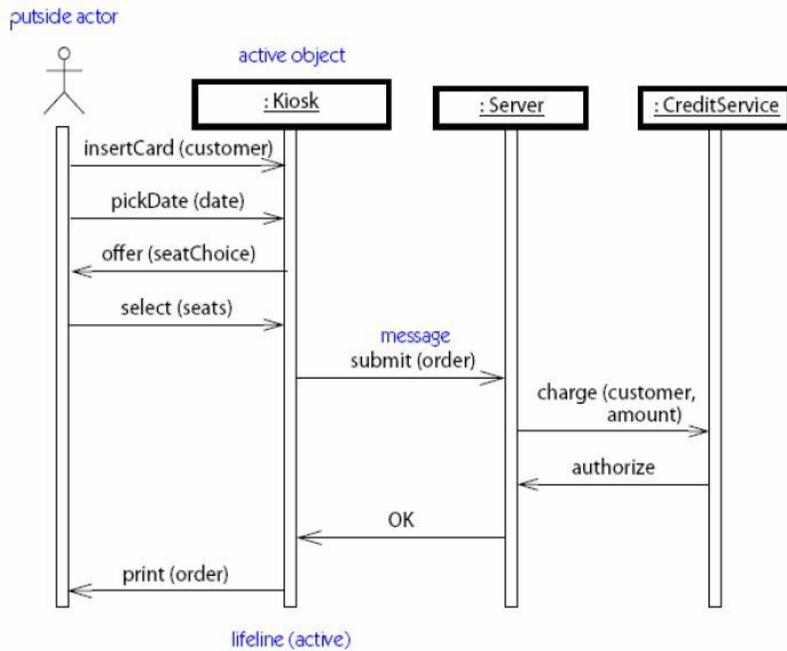
Dalam bahasan use case, para pengguna disebut sebagai aktor. Aktor dapat meliputi pelanggan, petugas layanan konsumen, manajer penjualan, dan analisis produk. Seorang aktor dapat menggunakan banyak use case, sebaliknya, sebuah use case juga dapat digunakan oleh beberapa aktor.



**Gambar 2.7 Contoh Diagram Use Case**

### 2.6.3.2 Diagram Sequence

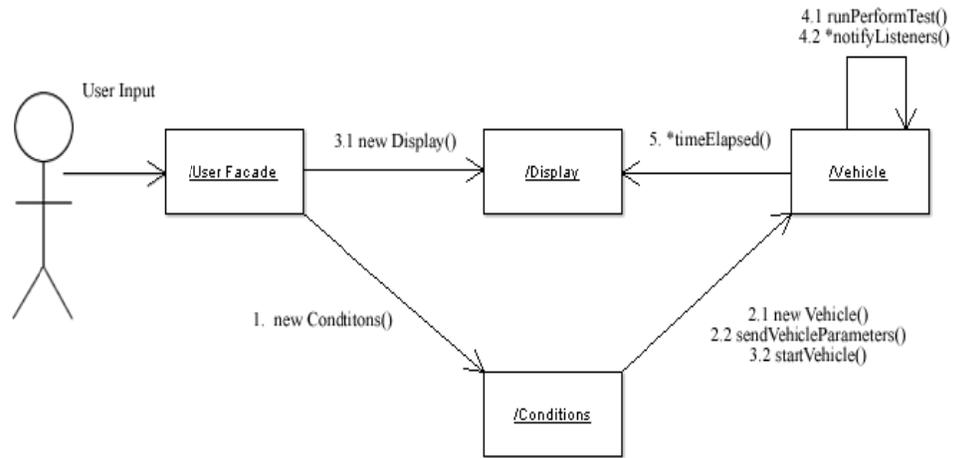
Diagram sequence secara khusus menjabarkan sebuah perilaku tunggal. Diagram tersebut menunjukkan sejumlah objek contoh dan pesan-pesan yang melewati objek-objek ini di dalam use case.



**Gambar 2.8 Contoh Diagram Sequence**

### 2.6.3.3 Diagram Kolaborasi

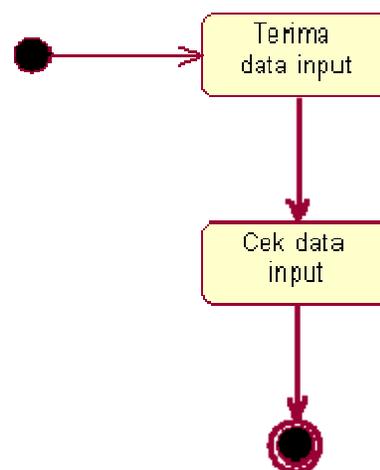
Diagram kolaborasi menggambarkan interaksi (atau kolaborasi) antara objek dalam sebuah format jaringan. Diagram kolaborasi tidak fokus pada *timing* atau “sekuensi” pesan. Diagram kolaborasi merupakan *isomorphic* artinya kita dapat mengubah dari satu diagram ke diagram lain.



**Gambar 2.9 Contoh Diagram Kolaborasi**

#### 2.6.3.4 Diagram Statement

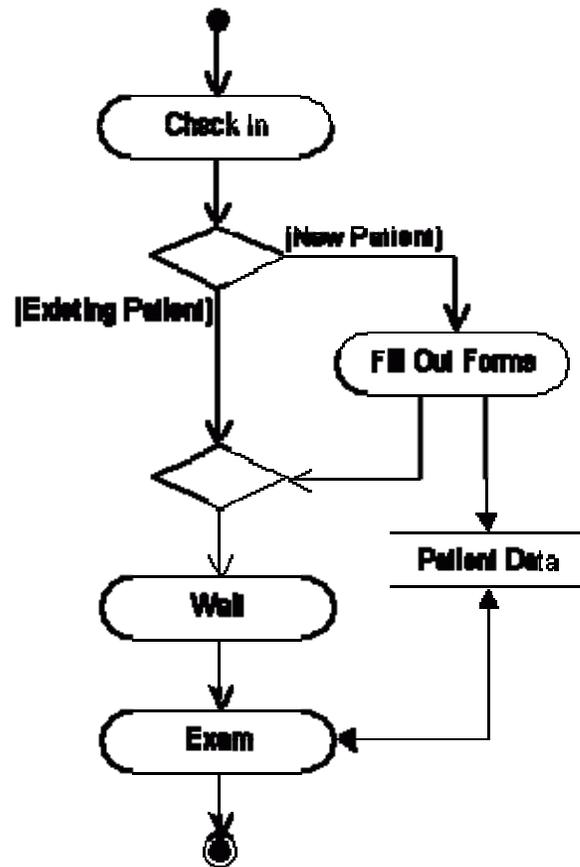
Diagram statement digunakan untuk memodelkan *behavior* objek khusus yang dinamis. Diagram ini mengilustrasikan siklus hidup objek-berbagai keadaan yang dapat diasumsikan oleh objek dan *event-event* (kejadian) yang menyebabkan objek beralih dari satu *state* ke *state* lain.



**Gambar 2.10 Contoh Diagram Statement**

### 2.6.3.5 Diagram Aktivitas

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



Gambar 2.11 Contoh Diagram Aktivitas

## 2.7 Interaksi Manusia dan Komputer

### 2.7.1 Pengertian Interaksi Manusia dan Komputer

Interaksi manusia dengan komputer adalah ilmu yang mempelajari tentang bagaimana mendesain, mengevaluasi, dan mengimplementasikan sistem komputer yang interaktif sehingga dapat digunakan oleh manusia. Interaksi manusia dan komputer terfokus pada perancangan antarmuka pemakai (*user interface*).

### 2.7.2 Delapan Aturan Emas Perancangan Antar Muka

Menurut Ben Shneiderman (1998, p78), dalam perancangan *user interface*, digunakan *Eight golden rules interface design* atau yang sering disebut dengan 8 aturan emas perancangan, yaitu:

1. Berusaha untuk konsisten.

Konsisten dalam aksi-aksi dalam situasi tertentu. Konsistensi dalam penggunaan jenis *font*, warna, simbol, tata letak, bentuk tombol.

2. Memungkinkan *user* untuk menggunakan *shortcut*.

Menyediakan tombol-tombol *shortcut* untuk aksi yang sering digunakan. Dengan adanya peningkatan dalam penggunaan *shortcut* maka dapat meningkatkan kecepatan tampilan frekuensi kecepatan interaksi dan mengurangi jumlah interaksi yang diperlukan.

3. Memberikan umpan balik yang informatif.

Memberikan respon yang sesuai dengan aksi yang dilakukan pengguna, sehingga dapat membantu pengguna untuk mengerti sistem yang telah dibuat dalam suatu aplikasi.

4. Merancang dialog yang memiliki akhir atau penutup.

Urutan aksi harus diatur dalam grup dimana ada awal, tengah, dan akhir. Dengan adanya umpan balik dapat memberikan pilihan untuk menyiapkan grup aksi berikutnya.

5. Penanganan kesalahan yang sederhana.

Desain sistem sedemikian rupa sehingga pengguna tidak melakukan kesalahan yang fatal. Apabila pengguna melakukan kesalahan, sistem harus dapat mendeteksi kesalahan serta memberikan instruksi sederhana dan spesifik agar pengguna dapat melakukan perbaikan.

6. Memungkinkan pembalikan aksi yang mudah.

Sedapat mungkin semua aksi dapat dikembalikan. Hal ini dapat mengurangi kegelisahan pengguna, karena kesalahan yang dilakukan oleh pengguna dapat diperbaiki.

7. Mendukung pusat kendali *internal*.

Operator yang berpengalaman menginginkan bahwa mereka bertanggung jawab terhadap sistem dan sistem merespon aksi yang diberikan, karena manusia yang memegang kontrol.

#### 8. Mengurangi ingatan jangka pendek.

Keterbatasan manusia dalam pemrosesan informasi memerlukan tampilan yang sederhana, tampilan banyak halaman digabungkan, frekuensi pergerakan *window* dikurangi, dan waktu pelatihan yang cukup diberikan untuk kode-kode, hafalan dan urutan aksi-aksi.

### 2.7.3 Sepuluh Kesalahan Utama Website

Menurut Ben Shneiderman(1998, p74), dalam perancangan *website*, seringkali melakukan kesalahan dalam pembuatan desain. Berikut ini merupakan 10 kesalahan utama yang sering terjadi dalam pembuatan *website design* :

1. Penggunaan frame.
2. Penggunaan teknologi baru dengan serampangan.
3. Gerakan teks dan animasi yang berjalan terus.
4. URL yang kompleks.
5. Halaman yatim.
6. Navigasi harus tampak dari atas.
7. Kurangnya dukungan navigasi.
8. Warna link yang tidak standart.
9. Informasi yang tidak *up to date*.
10. Waktu *download* yang terlalu lama.

## **2.8 E-learning**

### **2.8.1 Pengertian E-learning**

Menurut Emphy dan Zhuang (2005, p6) *e-learning* adalah kegiatan pelatihan yang menggunakan media elektronik atau teknologi informasi. Berbagai istilah yang digunakan untuk mengartikan *e-learning* antara lain, *web-based learning*, *online learning*, *computer-based training/ learning*, *distance learning*, *computer-aided instruction*, dll.

Menurut Turban (2005, p165) *e-learning* merujuk pada pembelajaran yang didukung melalui web yang dapat digunakan di dalam kelas biasa atau kelas virtual.

Menurut Matt Comerchero (2006, p1) *e-learning* adalah suatu bentuk pendidikan yang menggabungkan motivasi diri sendiri, komunikasi, efisiensi, dan teknologi. Dikarenakan adanya keterbatasan interaksi sosial, pelajar harus menjaga agar mereka tetap termotivasi. *E-learning* cukup efisien dengan menghilangkan jarak dan kendala lain, dimana jarak dihilangkan karena *e-learning* dibuat dengan media yang dapat diakses dengan perangkat yang berhubungan dengan internet.

### **2.8.2 Tipe-tipe E-learning**

Menurut Matt Commerchero (2006, p1) *e-learning* dapat dibedakan berdasarkan 4 hal, yaitu:

## 1. Cara berkomunikasi

Berbagai cara berkomunikasi antar pelajar atau pembimbingnya. Dengan *e-learning* dapat dilakukan melalui aplikasi *online*. Dengan cara itu berkomunikasi tetap dapat dilakukan secara *face to face* melalui *video* dan *audio*.

## 2. Jadwal

Berdasarkan jadwalnya, *e-learning* dapat dibedakan menjadi 2, yaitu:

- Synchronous

*Synchronous* berarti terjadi pada waktu yang sama, di mana proses belajar dan mengajar terjadi pada waktu yang bersamaan. Digunakan dalam *video conference*, *teleconference*, dll. *Synchronous* mengharuskan pelajar dan guru untuk mengakses internet bersamaan.

- Asynchronous

*Asynchronous* berarti terjadi tidak pada waktu yang sama. Proses belajar tidak terjadi secara langsung, pelajar dapat mengambil pelatihan pada waktu yang berbeda dengan pengajar.. Contoh dari penggunaan *Asynchronous* adalah *email*, *online forum*, dan *thread discussion*.

### 3. Struktur kelas *e-learning*

Struktur kelas menandakan bagaimana pelajaran diberikan. Dapat berupa belajar mandiri, belajar dengan bimbingan, atau belajar mandiri dengan seorang ahli.

### 4. Teknologi yang dipakai

Teknologi yang digunakan untuk mengimplementasikannya tidak terbatas pada informasi yang berbasis web, tetapi *e-learning* dapat dicapai dengan memanfaatkan segala bentuk teknologi informasi yang menghasilkan media yang mendukung. Contohnya adalah rekaman *Video* dan *Audio*.

## 2.8.3 Keuntungan E-learning

Menurut Empy dan Zhuang (2005, p9), ada beberapa keuntungan *e-learning*, yaitu:

### 1. Mengurangi biaya

Dengan menggunakan *e-learning* kita dapat menghemat waktu dan uang untuk mencapai suatu tempat pembelajaran. Dengan *e-learning* kita dapat mengakses dari berbagai lokasi dan tempat.

### 2. Fleksibilitas waktu, tempat, dan kecepatan pembelajaran

Dengan menggunakan *e-learning*, pengajar dapat menentukan waktu untuk belajar di manapun. Dengan *e-learning* juga

pelajar dapat belajar sesuai dengan kemampuan masing-masing. Berbeda dengan belajar di kelas, di mana semua pelajar belajar mulai dan berhenti pada waktu yang sama.

### 3. Standarisasi dan efektivitas pengajaran

*E-learning* selalu memiliki kualitas sama setiap kali diakses dan tidak tergantung suasana hati pengajar. *E-learning* dirancang agar pelajar dapat lebih mengerti pelajaran dengan menggunakan simulasi ataupun teknologi animasi.

#### 2.8.4 Kerugian E-learning

Disamping kelebihanannya, *e-learning* juga memiliki kekurangan, yaitu:

1. Pelajar harus memiliki komputer dan akses internet.
2. Dengan tidak adanya rutinitas yang ada di kelas tradisional maka pelajar mungkin akan berhenti belajar dikarenakan bingung atau tidak mengetahui kegiatan belajar dan tenggat waktu tugas.
3. Pelajar harus memotivasi diri sendiri, dikarenakan pelajar belajar sendiri tanpa ada bantuan pengajar setiap saat. Sehingga pelajar harus disiplin dan mengerjakan tugas secara mandiri tanpa bantuan pengajar.

4. Pelajar dan pengajar harus memiliki kemampuan menulis dan kemampuan komunikasi yang baik. Karena tidak saling bertatap muka sehingga mungkin terjadi salah pengertian dalam beberapa hal.

## 2.9 ASP.Net

ASP.Net merupakan salah satu bahasa pemrograman yang sering digunakan oleh *programmer* dalam membuat sebuah *website*. ASP.Net adalah sebuah *web application framework* yang dikembangkan oleh Microsoft supaya *programmer* dapat membuat *web site*, web aplikasi, atau *web service* yang dinamis.

Setelah merilis *Internet Information Services* 4,0 pada tahun 1997, Microsoft mulai mencari kemungkinan baru untuk aplikasi web model yang akan memecahkan masalah ASP, terutama dalam hal pemisahan dan konten presentasi. Mark Anders, seorang manajer tim di IIS, dan Scott Guthrie yang telah bergabung dengan Microsoft pada tahun 1997. Desain awal *prototype* yang telah dikembangkan oleh mereka berdua selama dua bulan disebut dengan XSP. Awalnya XSP dikerjakan dengan menggunakan Java, akan tetapi mereka memutuskan untuk membangun suatu *platform* baru di atas *Common Language Runtime* (CLR).

Dengan berpindah ke CLR, XSP dapat kembali diterapkan di C# dan diubah namanya menjadi LEU+, oleh karena itu *platform* yang baru dilihat sebagai penerus aktif ke *Server Page*, dan bertujuan untuk menyediakan jalur migrasi yang mudah bagi pengembang ASP.

Kemudian Microsoft mengeluarkan ASP. ASP.Net sendiri merupakan pengembangan dari ASP 3.0, kemudian pada ASP.Net muncul pertama kali pada tahun 2000 yaitu ASP.Net 1.0 yang di *release* pada 5 Januari 2002 dengan .Net Framework versi 1.0 pada saat itu. Perbedaan ASP dengan ASP.Net untuk mengakses *library* dan mendukung bahasa pemrograman yang ada pada Visual Studio seperti VB.Net, C#, dan ASP.Net, dengan adanya *framework* ini diharapkan dapat membuat menjadi lebih ringkas dan cepat dikarenakan penggunaan *framework* dapat mengurangi jumlah *script* yang dibuat.

## 2.10 SQL Server

SQL Server atau yang biasa disebut Microsoft SQL Server adalah sebuah sistem manajemen basis data relasional (RDBMS) produk Microsoft. Bahasa *query* utamanya adalah *Transact-SQL* yang merupakan implementasi dari SQL standar ANSI/ISO yang digunakan oleh Microsoft dan *Sybase*. Umumnya SQL Server digunakan di dunia bisnis yang memiliki basis data berskala kecil sampai dengan menengah, tetapi kemudian berkembang dengan digunakannya SQL Server pada basis data besar.

Microsoft SQL Server dan *Sybase/ASE* dapat berkomunikasi lewat jaringan dengan menggunakan protokol TDS (*Tabular Data Stream*). Selain dari itu, Microsoft SQL Server juga mendukung ODBC (*Open Database Connectivity*), dan mempunyai *driver* JDBC untuk bahasa pemrograman Java. Fitur yang lain dari SQL Server ini adalah kemampuannya untuk membuat basis data *mirroring*

*dan clustering*. Pada versi sebelumnya, MS SQL Server 2000 terserang oleh *worm* komputer SQL Slammer yang mengakibatkan kelambatan akses Internet pada tanggal 25 Januari 2003.