

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Sistem merupakan sebuah kata yang luas sekali maknanya. Dalam kehidupan sehari-hari sistem sering kali disamakan dengan ungkapan "cara".

Suatu sistem pada dasarnya adalah sekelompok unsur yang saling terkait, berhubungan, dan berfungsi bersama-sama untuk mencapai suatu tujuan. Menurut James A. Hall, (2001: 5)

"Sebuah sistem adalah sekelompok dua atau lebih komponen-komponen yang saling berkaitan (interrelated) atau subsistem-subsistem yang bersatu untuk mencapai tujuan yang sama (common purpose)."

Menurut A. Aseervatham dan D. Anandarajah (2003: 4)

"A system can be said to be composed of distinguishable parts or elements whose relationship to one another is defined and whose behaviour is mutually supportive towards the achievement of a common goal."

2.2. Konsep Dasar Informasi

Informasi merupakan sesuatu yang dapat mengurangi derajat ketidakpastian tentang suatu keadaan atau kejadian, oleh karena itu informasi merupakan sesuatu yang sangat penting dan vital dalam suatu organisasi.

Informasi adalah istilah dengan banyak arti bergantung pada konteks, tetapi sebagai aturan berhubungan erat dengan konsep seperti arti, pengetahuan, negentropy, komunikasi, kebenaran, representasi, dan rangsangan mental. Sekalipun banyak orang menyatakan munculnya "era informasi", "masyarakat informasi," dan teknologi informasi, dan sungguhpun ilmu informasi dan ilmu komputer sering disorot, kata "informasi" sering dipakai tanpa pertimbangan hati-hati dari berbagai arti yang dimiliki.

Data diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai *input*, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus. Siklus itu disebut dengan siklus informasi (*information cycle*).

2.2.1. Kualitas Informasi

Untuk dapat berguna, maka informasi harus didukung oleh tiga pilar sebagai berikut: tepat nilainya atau akurat (*accurate*), tepat waktu (*timeliness*) dan tepat kepada orangnya atau relevan (*relevance*). Keluaran yang tidak didukung ketiga pilar ini tidak dapat dikatakan sebagai informasi yang berguna, tetapi merupakan sampah (*garbage*) (Jogiyanto, 1999: 10).

Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya.

Tepat pada waktunya, berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi karena informasi merupakan landasan didalam pengambilan keputusan.

Relevan, berarti informasi tersebut harus mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda.

2.3. Konsep Dasar Sistem Informasi

Sistem Informasi adalah Sekumpulan *hardware*, *software*, *brainware*, prosedur dan atau aturan yang diorganisasikan secara integral untuk mengolah data menjadi informasi yang bermanfaat guna memecahkan masalah dan pengambilan keputusan

Sistem Informasi adalah satu kesatuan data olahan yang terintegrasi dan saling melengkapi yang menghasilkan output baik dalam bentuk gambar, suara maupun tulisan(wikipedia.com). **Sistem Informasi** adalah aplikasi komputer untuk mendukung operasi dari suatu organisasi: operasi, instalasi, dan perawatan komputer, perangkat lunak, dan data.

2.3.1. Komponen Sistem Informasi

Sistem informasi mempunyai enam buah komponen, yaitu komponen *input* atau masukan, komponen model, komponen *output* atau keluaran, komponen teknologi, komponen basis data, dan komponen kontrol atau pengendalian. Keenam komponen ini bersama-sama membentuk satu kesatuan. Jika satu atau lebih komponen tersebut tidak ada, maka sistem informasi tidak akan dapat melakukan fungsinya, yaitu pengolahan data dan tidak dapat mencapai tujuannya, yaitu menghasilkan informasi yang relevan, tepat waktu, dan akurat. Komponen-komponen dari sistem ini dapat digambarkan sebagai berikut ini.

1. Komponen *input*

Input merupakan data yang masuk ke dalam sistem informasi. Komponen ini perlu ada karena merupakan bahan dasar dalam pengolahan informasi. Sistem informasi tidak akan dapat menghasilkan informasi jika tidak mempunyai komponen *input*. Jika sistem informasi tidak pernah mendapatkan *input*, tetapi dapat menghasilkan *output*, ini merupakan hal yang ajaib. *Input* yang masuk ke dalam sistem informasi dapat langsung diolah menjadi informasi atau jika belum dibutuhkan sekarang dapat disimpan terlebih dahulu di *storage* dalam bentuk basis data (*database*).

Data untuk sistem informasi perlu ditangkap dan dicatat di dokumen dasar. Dokumen dasar (*source document*) merupakan formulir yang

digunakan untuk menangkap data yang terjadi. Dokumen dasar sangat penting di dalam arus data sistem informasi. Dokumen dasar ini dapat membantu di dalam penanganan arus data sistem informasi, yaitu:

- a. Dapat menunjukkan macam dari data yang harus dikumpulkan dan ditangkap;
- b. Data dapat dicatat dengan jelas, konsisten, dan akurat;
- c. Dapat mendorong lengkapnya data akuntansi, disebabkan data yang dibutuhkan disebutkan satu-persatu di dalam dokumen dasarnya;
- d. Bertindak sebagai pendistribusi data, karena sejumlah tembusan dari formulir-formulir tersebut dapat diberikan kepada individu-individu atau departemen-departemen yang membutuhkannya;
- e. Dokumen dasar dapat membantu di dalam pembuktian terjadinya suatu transaksi yang sah, sehingga sangat berguna untuk pelacakan pemeriksaan (*audit trail*).
- f. Dokumen dasar dapat digunakan sebagai cadangan atau pelindung (*back-up*) dari file-file data di komputer.

2. Komponen *output*

Produk dari sistem informasi adalah *output* berupa informasi yang berguna bagi para pemakainya. *Output* merupakan komponen yang harus ada di sistem informasi. *Output* dari sistem informasi dibuat dengan menggunakan data

yang ada di basis data dan diproses menggunakan model tertentu.

3. Komponen basis data

Basis data (*database*) adalah kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Dari definisi ini, terdapat tiga hal yang berhubungan dengan basis data, yaitu sebagai berikut ini.

- a. Data itu sendiri yang diorganisasikan dalam bentuk basis data (*database*).
- b. Simpanan permanen (*storage*) untuk menyimpan basis data tersebut. Simpanan ini merupakan bagian dari teknologi perangkat keras yang digunakan di sistem informasi. Simpanan permanen yang umumnya digunakan berupa *hard disk*.
- c. Perangkat lunak untuk memanipulasi basis datanya. Perangkat lunak ini dapat dibuat sendiri dengan menggunakan bahasa pemrograman komputer atau dibeli dalam bentuk suatu paket. Paket perangkat lunak ini disebut dengan DBMS (*Data Base Management Systems*).

4. Komponen model

Informasi yang dihasilkan oleh sistem informasi berasal dari data yang diambil dari basis data yang diolah lewat suatu model-model tertentu. Model-model yang digunakan di sistem informasi dapat berupa model logika yang menunjukkan suatu proses perbandingan logika

atau model matematik yang menunjukkan perhitungan matematika.

5. Komponen teknologi

Teknologi merupakan komponen yang penting di sistem informasi. Tanpa adanya teknologi yang mendukung, maka sistem informasi tidak akan dapat menghasilkan informasi yang tepat waktunya. Komponen teknologi dapat dikelompokkan ke dalam dua macam kategori, yaitu teknologi sistem komputer (perangkat keras dan perangkat lunak) dan teknologi sistem telekomunikasi.

6. Komponen kontrol

Komponen kontrol juga merupakan komponen yang penting dan harus ada di sistem informasi. Komponen kontrol ini digunakan untuk menjamin bahwa sistem informasi yang dihasilkan oleh sistem informasi merupakan informasi yang akurat.

2.4 Aplikasi Mobile Business

Saat ini, perkembangan teknologi piranti *mobile* yang didukung oleh internet, menciptakan akses di setiap waktu dan tempat semakin meluas. Piranti *mobile* menjadi semakin pintar, terciptanya mikroprosesor yang kecil, efektif dan mempunyai kecepatan tinggi.

Banyaknya pemakai peranti *mobile* telah merevolusi kantor tradisional menjadi kantor *mobile*. Peranti *mobile* sekarang tidak hanya berfungsi sebagai pencatat jadwal dan buku alamat. Fungsi peranti *mobile* sudah berkembang

pesat, dan idealnya siap menggantikan dokumen bisnis berbasis kertas.

Vendor-vendor peranti lunak telah menanam teknologi penangkapan data ke dalam produknya. Hal ini menciptakan peningkatan produktivitas, proses pelaporan yang lebih cepat dan pengurangan biaya operasional. Secara dasar aplikasi peranti *mobile* terbagi atas :

1. *Personal Information Management (PIM)* : Menyediakan fungsi seperti kalender, buku alamat, jadwal memo dan tugas yang harus dilakukan.
2. *Dokumen* : Menyediakan fungsi *word processing* dan *spread sheet*. Fungsi-fungsi yang lebih maju antara lain akses web, e-book, multimedia, dan presentasi.
3. *Aplikasi Mobile Business (M-Business)* : Menyediakan fungsi penangkapan, pemrosesan, dan pengiriman data.
4. *Games dan Hiburan* : Menyediakan fungsi permainan, musik video dan sebagainya.

Aplikasi *mobile business* dikategorikan secara horisontal dan vertikal. Kategori horisontal berdasarkan jenis aplikasinya, sedangkan aplikasi vertikal berdasarkan jenis bisnisnya.

Secara Horisontal antara lain :

1. *Customer Relationship Management (CRM)* : CRM memiliki 5 modul dasar yaitu : *Sales Force Automation, Field Service, Customer Support, Call Center,* dan *Marketing Analysis*. Secara operasional biasanya para pengguna garis depan

CRM ini bekerja di lapangan terutama *salesman* penggunaan aplikasi *mBusiness*, tentunya akan sangat mempercepat eksekusi proses CRM. Contohnya jika ada *komplain* pelanggan terhadap *customer field service* maka dengan cepat *komplain* tersebut bisa diproses langsung di lapangan tanpa harus menunggu di kantor.

2. *Enterprise Resource Planing (ERP)* : ERP bisa dikatakan sebagai inti dari sistem informasi sebuah organisasi. Mengingat ERP adalah suatu *backbone system*, tentunya akan sangat menguntungkan sekali jika koneksinya ditingkatkan ke level *mobile*. Para pekerja bermobilitas tinggi tidak akan kerepotan lagi untuk menyelesaikan laporan dan proses-proses lainnya dimana dan kapan saja melalui peranti *mobile*.

3. *Transportation Management System (TMS)* : Fungsi utama TMS adalah untuk mengatur *utility loading* (Pemuatan), *routing* (Perjalanan) dan estimasi pengiriman. Penggunaan aplikasi *mobile* akan sangat meningkatkan kecepatan proses TMS. Contohnya, sebuah *transport* yang diatur untuk mengantarkan barang, akan dengan mudah mengirimkan status posisi dan kondisinya melalui perangkat *mobile*.

Selain kategori horisontal, aplikasi *mobile* M-Business juga dikategorikan secara vertikal. Beberapa kategori vertikal antara lain :

1. Jasa Keuangan
2. Kesehatan
3. *Customer Care*

4. Manufaktur
5. Distribusi

2.5 Visual Basic.Net (Mobile Application)

VB .NET dibangun diatas fondasi .NET Framework (lingkungan kerja .NET). Microsoft memaparkan pengembangan Microsoft .NET dalam PDC (*Professional Developers Conference*). Setelah melalui pengembangan cukup lama Microsoft merilis VS.NET dimana salah satu bahasa pemrogramannya adalah VB.NET.

Sekarang ini pemakaian *mobile device* atau *smart device* semakin marak. Dengan menggunakan VB .NET 2003, kita dapat membuat aplikasi untuk *smart device*.

PDA (*Personal Digital Asssistants*) merupakan komputer genggam yang awalnya dibuat untuk *organizer* pribadi yang biasanya memiliki fitur dasar: buku tanggal, buku alamat, daftar tugas dan memo pad. Terdapat juga fitur tambahan untuk melakukan sinkronisasi informasi dari PDA ke PC.

Ada tiga sistem operasi yang saling bersaing dalam pasar PDA yaitu *Palm*, *Microsoft Pocket PC* dan *Symbian*.

Microsoft Visual Studio.NET 2003 menyediakan *Smart Device Extention (SDE)*, yang dapat digunakan untuk membuat aplikasi bagi *Pocket PC* dan *Microsoft Windows CE*.

Membuat aplikasi menggunakan SDE dari VB.NET 2003, sama seperti membuat aplikasi Windows umumnya yaitu membuat proyek baru SDE, menambahkan kontrol ke form dan menulis kode. Jika anda tidak memiliki perangkat keras PDA dengan sisitem operasi *pocket PC*, Anda tidak

perlu khawatir tidak bisa mencoba aplikasi ini karena Microsoft sudah menyediakan emulator Pocket PC.

Untuk membuat aplikasi *mobile programmer* dapat menerapkan hampir semua fitur yang ada pada visual basic .net seperti fitur bahasa berorientasi objek yang akan memudahkan progamer VB.net yang sudah terbiasa membuat aplikasi berbasis Windows, untuk membuat aplikasi *mobile* yang berjalan pada sistem operasi *mobile* seperti Windows CE atau Pocket PC.

2.6 Remote Data Access (RDA)

Merupakan salah satu solusi koneksi database pada SQL CE yang menggunakan *authentication, authorization, dan encryption services* dari *Microsoft Internet Information Services (IIS)*. Koneksi dengan RDA dapat dilakukan dengan menggunakan Microsoft ActiveSync dengan kabel data yang memungkinkan terjadinya sinkronisasi data antara SQL CE dengan SQL Server. Dalam hal ini maka akan mengatasi kekurangan dari *mobile device* yang tidak memiliki koneksi seperti GPRS, bluetooth.

RDA dapat digunakan dengan dua cara yaitu *client* dapat mengambil data pada server untuk disimpan pada *database lokal*, kemudian *client* dapat melakukan manipulasi secara *offline* pada *database lokal* di *mobile device*. Setelah itu *client* dapat mengembalikan data pada yang telah dimanipulasi ke server untuk disinkronisasi.

Implementasi untuk RDA dapat dilakukan melalui 2 cara yaitu:

1. *Pull method*

Pull method digunakan untuk menarik data dari server saat terkoneksi melalui ActiveSync dan menyimpannya pada *database* lokal di *mobile device*.

2. *Push Method*

Push method digunakan untuk mengembalikan data ke server dari *database* lokal pada *mobile device* untuk di gabungkan dengan *database* yang ada pada server.

2.7 Tools Yang Digunakan

2.7.1 Windows CE

Windows CE adalah perangkat lunak keluaran Microsoft yang di *design* untuk perangkat *organizer*, *handheld* computer, dan bahkan untuk mesin game (John O'Donnel, 2001).

Untuk menjangkau pasar yang berbeda Microsoft mengambil inti dari system operasi Windows CE dan memproduksi kategori yang berbeda seperti *Handled PC*, *Pocket PC*, dan *Auto PC*. Setiap kategori tersebut memiliki elemen unik tersendiri. Saat ini *Pocket PC* telah menjadi lebih populer dengan *form* dari Windows CE dan tampilan layer 320 x 240. *Handled PC* telah menjadi *Handled PC* 2000 dan telah mendukung *device* dari Hewlett Packard. Sega bahkan telah melisensi Windows CE untuk *console* Sega Dreamcastnya.

Pada saat mengembangkan Windows CE, microsoft memutuskan agar Windows CE memiliki dukungan terhadap

multi processor. Alasannya mungkin disebabkan OEM yang berbeda memerlukan processor yang berbeda pula, yang berarti memungkinkan Windows CE berjalan pada empat mesin yang masing-masing menggunakan processor yang berbeda seperti : ARM, MIPS, SH3, SH4, dan X86. Perlu dicatat bahwa pada *Pocket PC 2002* semua mesin menggunakan processor ARM. Dan saat ini Microsoft telah mengeluarkan *development tools* dengan *cross compiler*, yang artinya kode hanya perlu ditulis sekali dan kemudian dapat *dicompile* di processor yang berbeda.

Beberapa fitur - fitur yang tersedia pada Windows CE pada kegiatan sehari hari seperti :

1. Akses *email* lewat GPRS
2. Akses *internet*, LAN, modem, atau koneksi RAS
3. Koneksi ke alat lainnya melalui *Bluetooth*, *Infrared*, dan *WiFi*

Keunggulan utama dari Windows CE / *Pocket PC* adalah kompatibilitas dengan Microsoft Windows dalam komunikasi data, penanganan *transfer file*, sinkronisasi, dan akses *database* sehingga *Pocket PC* dengan berbasis Windows CE sudah mulai dipakai pada beberapa PDA (*Personal Digital Assistant*) terbaru.

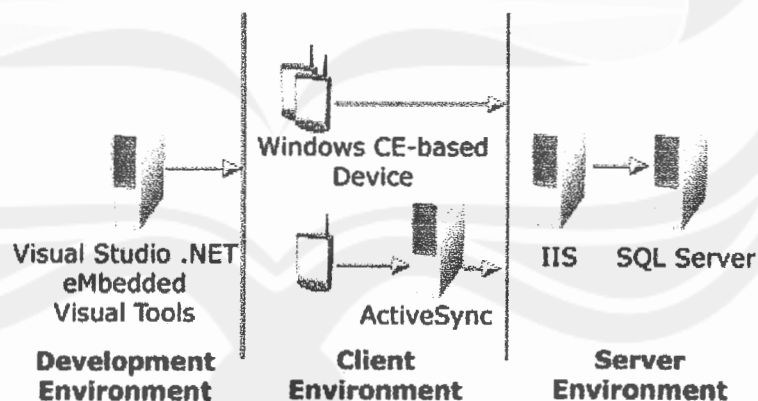
Untuk membuat komunikasi antara Windows CE dan komputer desktop dalam pembuatan *software*, diperlukan suatu *software* dari Microsoft yang bernama *ActiveSync*. Dengan aplikasi tersebut pengguna *Pocket PC* dapat menginstall/menghapus aplikasi, *backup data*, *restore data*, sinkronisasi *email/file/kontak*, *remote data access*, dsb

Seperti halnya sistem operasi windows yang lain, mesin dengan Windows CE dapat diprogram dengan

menggunakan banyak bahasa. Misalnya memprogram dengan bahasa C dengan melakukan pemanggilan standar Win32 API, atau dengan menggunakan bahasa C++ dan visual basic. Bahkan dimungkinkan untuk membuat program di mesin *Pocket PC* dengan menggunakan bahasa Java. Untuk .NET, kode yang ditulis dalam bahasa VB.NET dapat ditransfer ke *Pocket PC* dan kemudian dieksekusi di sana dengan menggunakan .NET Compact Framework.

2.7.2 SQL CE 2.0

SQL CE 2.0 merupakan sebuah solusi untuk *mobile database* yang dikeluarkan oleh Microsoft. Dengan menggunakan SQL server CE edition anda dapat melebarkan akses database pada *mobile device*, yang memungkinkan anda untuk membawa dan melakukan perubahan data perusahaan pada perangkat *mobile* yang bersistem operasi Windows CE atau *Pocket PC*. Berikut adalah gambar *environment* dimana SQL CE digunakan



Gambar 2.1 SQL CE Environment

1. Development Environment

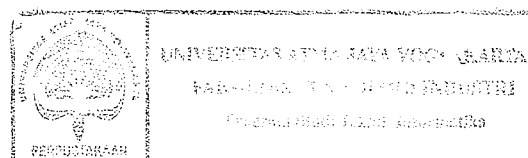
Ini adalah komputer dimana aplikasi *mobile* dibuat dengan menggunakan *tools* Visual Studio.Net.

2. Client Environment

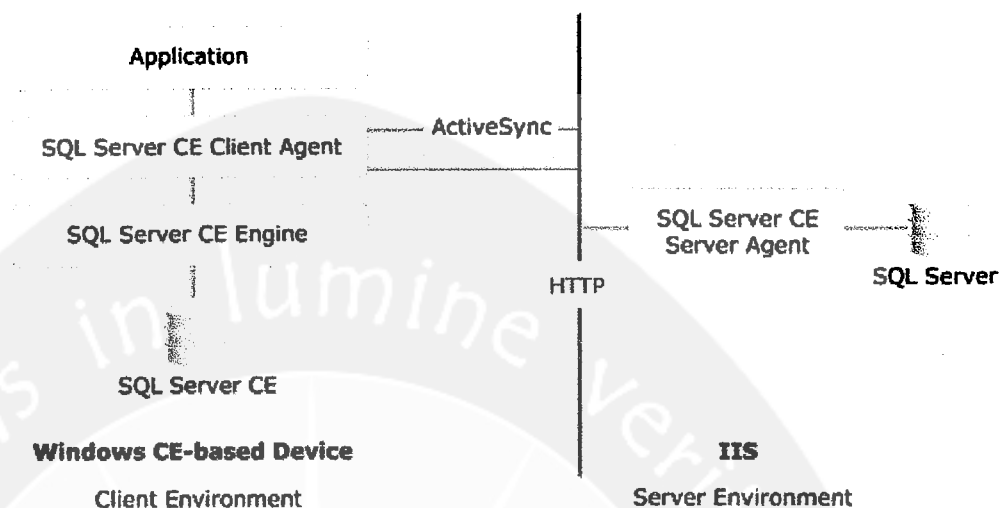
Ini adalah perangkat dimana aplikasi *mobile* terinstall. Pada perangkat dimana tidak terdapat konektivitas seperti GPRS, Bluetooth maka digunakan Microsoft ActiveSync untuk melakukan koneksi dengan kabel data yang memungkinkan terjadi sinkronisasi data antara *server* dan *client*.

3. Server Environment

Ini adalah bagian dari sistem dimana pada perangkat ini terinstall Microsoft Internet Information Services (IIS) dan SQL server 2000 yang merupakan *database* induk dari sistem. IIS digunakan untuk berkomunikasi antara *database* pada *mobile device* dan *database* pada *server*. Untuk berkomunikasi diperlukan metode *remote data access (RDA)* atau *merge replication* yang keduanya berkomunikasi melalui IIS.



Berikut ini ilustrasi hubungan antara SQL Server CE (*Client*) dengan SQL server (*Server*)



Gambar 2.2 Arsitektur SQL Server CE dan hubungannya dengan SQL server

Pada sisi *client mobile device*, SQL server CE engine bertugas mengelola penyimpanan data pada *database* lokal SQL Server CE termasuk *track record* yaitu penambahan, perubahan dan penghapusan data ketika data disinkronisasi dengan data yang ada pada server. SQL Server Client Agent adalah komponen yang digunakan untuk berkoneksi dengan server. Pada SQL CE ini di implementasikan dengan menggunakan SQL CE objek seperti SQL CE engine dan *Remote Data Acces (RDA)* objek. Pada sisi juga digunakan oleh aplikasi *mobile* untuk berkomunikasi dengan SQL CE. Pada sisi server, SQL Server CE Server Agent berperan sebagai perantara untuk SQL Server dengan SQL CE untuk berkomunikasi. Komunikasi dilakukan dengan HTTP melalui *web server*.

2.7.3 Microsoft SQL Server 2000

Microsoft SQL Server 2000 adalah sistem manajemen basis data yang memakai perintah-perintah Transact-SQL untuk mengirim perintah dari komputer *client* ke komputer *server*. Transact-SQL adalah bahasa SQL yang dikembangkan oleh Microsoft dengan menambahkan dialek-dialek tertentu. Microsoft SQL Server 2000 berisi *database*, mesin *database*, dan aplikasi yang diperlukan untuk mengelola data dan komponen-komponennya. (www.kompas.com, 18 April 2006)

MS SQL Server adalah salah satu produk *Relational Database Management System* (RDBMS) populer saat ini. Fungsi utamanya adalah sebagai *database server* yang mengatur semua proses penyimpanan data dan transaksi suatu aplikasi. Popularitas SQL Server akhir-akhir ini mulai menanjak dan setara dengan pesaing terdekatnya yaitu Oracle 9i dan Oracle 10g.

Versi 2000 memiliki *feature-feature* lengkap untuk membangun aplikasi mulai skala kecil sampai dengan tingkat *enterprise*. (ilmu computer.com)

Pada pembuatan *database client-server* tidak boleh diabaikan masalah *security*. *Security database* di SQL Server cukup handal untuk *database* menengah ke atas. *Security system* ini berhubungan erat dengan otoritas sebuah *client* dalam berkomunikasi dengan *database*. Ada banyak teknik *security* pada aplikasi. *Security* pada aplikasi dapat kita ciptakan sendiri dengan cara

membuat sebuah tabel *user* beserta otoritasnya untuk sebuah aplikasi tertentu. Akan tetapi hal ini kurang aman bilamana tabel yang kita buat sampai diketahui oleh orang yang tidak berkepentingan.

2.7.4 .NET Framework

.NET Platform merupakan satu set kumpulan teknologi yang memungkinkan teknologi *Internet* ditransformasikan ke dalam platform *distributed computing* dengan skalabilitas dan kompatibilitas tinggi. Secara teknis, .NET Platform menyediakan konsep pemrograman dengan library dan modul-modul baru yang konsisten, terlepas dari jenis bahasa pemrograman yang digunakan.

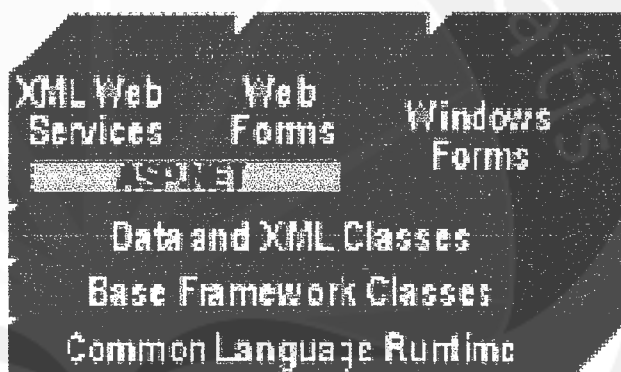
.NET Platform menyediakan hal-hal berikut bagi para developer :

- 1) *Language independent*, dengan *programming* model yang konsisten di semua tier aplikasi yang dibangun.
- 2) *Interoperability* dan *kompatibilitas* antar aplikasi.
- 3) Kemudahan migrasi dari teknologi yang ada saat ini.
- 4) Dukungan penuh terhadap berbagai teknologi standar yang digunakan dalam *platform internet*, antara lain HTTP, XML, SOAP dan HTML.

Teknologi inti .NET secara umum terdiri dari 4 area pokok :

1) .NET Framework

.NET Framework adalah teknologi inti yang menyediakan berbagai *library* untuk digunakan oleh aplikasi di atasnya. Komponen inti .NET Framework adalah *Common Language Runtime (CLR)* yang menyediakan *run time environment* untuk aplikasi yang dibangun menggunakan Visual Studio .NET, terlepas dari jenis bahasa pemrogramannya.



Gambar 2.3 Arsitektur .Net Framework

Dengan adanya CLR tersebut, programmer dapat menikmati *consistent object model* dalam mengakses berbagai komponen *library*. Dengan demikian penggunaan bahasa pemrograman dalam dunia .NET adalah lebih ke masalah selera atau *taste*, dan bukan pada kelebihan maupun kekurangan masing-masing bahasa. Mengapa ? Karena semua bahasa pemrograman yang mensupport .NET mengakses *library*

yang sama di dalam .NET Framework, dengan *object* model yang konsisten, dengan *run time file* yang sama. Bagi seorang *.Net Developer*, pemahaman terhadap konsep dan *object* model .NET Framework adalah jauh lebih penting daripada bahasa pemrograman itu sendiri.

Bagi programmer VB6, tentu mengetahui bahwa diperlukan distribusi *run time library* khusus ketika menginstall aplikasi. Demikian pula ketika anda menginstal aplikasi yang dibangun dengan Visual C++ maupun Delphi. Dalam dunia .NET, hal tersebut sudah tidak diperlukan lagi, selama .NET Framework telah terinstal di komputer sasaran.

2) .NET Building Block Services

Building block merupakan sekumpulan *services* yang bersifat *programmable*, yang dapat diakses secara *offline* maupun *online*. *Service* tersebut merupakan modul-modul yang terdapat di suatu komputer, server dalam jaringan, maupun di suatu server di internet.

Service ini merupakan suatu idealisasi di masa depan, dimana sebuah aplikasi bersifat terdistribusi dengan modul-modul yang tersimpan di berbagai tempat, tetapi dapat diintegrasikan membentuk suatu aplikasi. Konsep ini merupakan arah pengembangan *subscription based software*, yang saat ini mulai banyak berkembang dan dikenal sebagai *Application Service Provider*.

Service tersebut dapat diakses oleh berbagai *platform*, asalkan *platform* tersebut mensupport

protokol SOAP, yang merupakan protokol standar dalam mengakses *web service*. Peranan XML sebagai media definisi data menjadi sangat penting dalam hal ini, dan XML juga menjadi pusat perubahan besar dalam *platform .NET*. Dalam pelajaran selanjutnya, anda akan melihat bahwa semua data dalam *.NET* selalu direpresentasikan dalam bentuk XML.

3) Visual Studio .NET

Visual Studio .NET menyediakan *tools* bagi para developer untuk membangun aplikasi yang berjalan di *.Net Framework*. VS.Net membawa perubahan besar dalam gaya pemrograman, karena setiap programmer dituntut untuk memahami *.NET object model* dan *Object Oriented Programming* dengan baik, jika tidak ingin menghasilkan aplikasi dengan performa rendah.

VS.Net juga semakin mempertipis jarak antara *Windows Programmer* dengan *Web Programmer*. Dunia *scripting* yang akrab bagi *programmer web* akan sulit ditemukan dalam *.NET*, karena pemrograman web sudah bersifat *full object oriented*, dengan fasilitas *event driven programming* sebagaimana layaknya *windows programming*. Pemrograman web menjadi lebih mudah dan menyenangkan bagi para *programmer windows*.

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.

4) .Net Enterprise Server

Bagian ini merupakan sekumpulan server based *technology* yang digunakan untuk mendukung teknologi *.NET*, yang mencakup sistem operasi, *database*, *messaging*, maupun manajemen *e-commerce*. Teknologi yang disediakan antara lain adalah *Windows 2000 Server*, *SQL Server*, *Exchange*, *ISA Server* dan *BizTalk Server*.

Keuntungan menggunakan .NET:

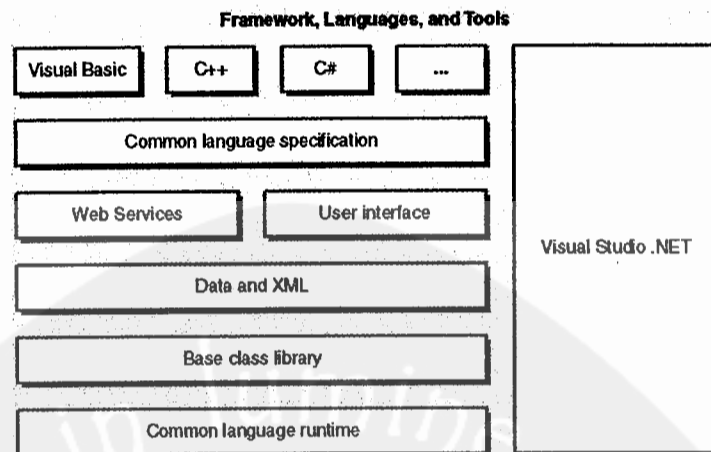
1. Cross Platform Possibility

.Net menyimpan dan mengirim data dalam bentuk XML yang merupakan format data universal di internet. Dengan demikian integrasi data antar platform lebih mudah dilakukan, selama platform tersebut mendukung XML. Representasi konsep ini adalah dataset, suatu cache data yang berbentuk XML dan dapat diakses dengan mudah. Sebuah data dapat diparsing antar tier aplikasi, baik dari database, middle tier, maupun aplikasi klien dalam format XML. Manipulasi format data dalam bentuk XML, .txt, maupun .rtf merupakan sesuatu yang menantang para programmer untuk membuat aplikasi lintas platform.

2. Multi Language

Arsitektur .NET bersifat terbuka, sehingga memungkinkan berbagai bahasa pemrograman mengakses CLR dengan mulus. Banyak kalangan menyebut .NET sebagai "open source" versi Microsoft. Saat ini .NET dapat diprogram menggunakan Visual Basic.NET, C++.NET, Visual C#, Javascript, dan J#. Berbagai third Party yang dapat digunakan adalah COBOL, Eiffel, Smalltalk, Perl, Python, ML, Pascal, dan Delphi. Para veteran Pascal mungkin tidak pernah bermimpi membuat aplikasi web dengan bahasa "kuno" tersebut, tetapi kehadiran .Net Framework merealisasikan hal tersebut.

Bagan berikut menjelaskan kedudukan bahasa pemrograman terhadap .NET Framework dengan CLR sebagai intinya.



Gambar 2.4 .Net Framework dan CLR sebagai inti

Sekali lagi, semua bahasa tersebut mengakses *object* model yang sama dalam .NET, sehingga bagi para petualang tersedia kemungkinan untuk berpindah dari satu bahasa ke bahasa lain dalam satu proyek aplikasi yang sama. .NET menyediakan integrasi bahasa pemrograman dalam satu peroyek aplikasi. Hal ini membuka kemungkinan kerjasama tim yang lebih baik, walaupun dalam tim tersebut masing-masing memiliki keahlian bahasa pemrograman yang berbeda.

3. kemampuan integrasinya kuat

kemampuan integrasinya yang kuat dengan semua lini infrastruktur dan aplikasi korporat dari Microsoft yang memudahkan pengembangan, implementasi, dan manajemen yang berujung efisiensi waktu dan biaya untuk korporat. Teknologi .NET Framework 2.0 yang terbaru bahkan memungkinkan aplikasi yang terpasang di ribuan user *ter-update* secara otomatis apabila ada perubahan yang dilakukan oleh tim pengembang

melalui fitur *ClickOnce*. Hal ini jelas memangkas biaya implementasi dan meningkatkan produktifitas tim secara keseluruhan.

Tujuan dari NET framework adalah :

1. Menyediakan lingkungan pemrograman berorientasi obyek , baik dijalankan secara local ataupun dijalankan secara terkontrol dari suatu tempat.
2. Menyediakan lingkungan untuk menjalankan suatu kode yang meminimalkan konflik saat *software* disebarkan dan *versioning*.
3. Menyediakan lingkungan untuk menjalankan suatu kode yang menjamin saat kode dijalankan, termasuk kode yang dibuat oleh pihak yang tidak diketahui atau pihak ketiga yang setengah dipercaya.
4. Menyediakan lingkungan untuk menjalankan suatu kode yang dapat mengeliminasi masalah performa dari lingkungan *scripted* atau *interpreted*.
5. Membuat pengembang memiliki pengalaman yang konsisten dalam berbagai tipe aplikasi seperti aplikasi berbasis Windows dan aplikasi berbasis web.

.NET Framework juga diartikan sebagai serangkaian teknologi yang:

1. mempersatukan aplikasi web yang sampai sekarang masih terisolasi.
2. Membuat informasi tersedia disetiap saat dan disetiap tempat.
3. Menyederhanakan pengembangan dan penyebaran aplikasi.