

BAB II

LANDASAN TEORI

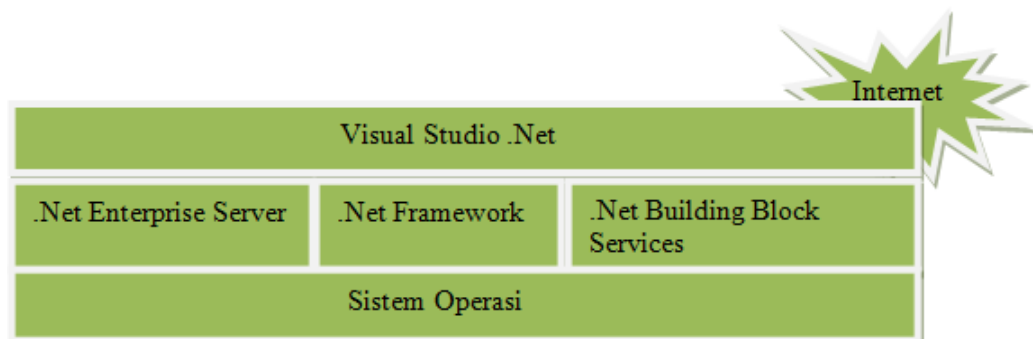
2.1 Teori tentang permasalahan

Dalam pengerjaan tugas akhir ini penulis menggunakan bahasa pemrograman yang berbasis .Net Framework yaitu Visual Basic .Net dan menggunakan SQL server 2005 sebagai database. Penulis akan menjelaskan tentang Microsoft .Net, pengertian .Net Freamwork, Visual Basic.Net dan SQL Server 2005.

2.1.1 Tentang Microsoft .Net

Menurut Ir. Suryanto Thabrani,MM dalam bukunya *Mudah & cepat Menguasai VB .net*, Platform Microsoft .Net merupakan model untuk development dimana platform dan aplikasi bisa dibuat dan dijalankan tanpa bergantung pada alat (device) yang dipakai. Teknologi ini memungkinkan beberapa aplikasi bekerjasama antara lain ; di harddisk user, network lokal, komputer remote, atau di Internet.

Dengan menggunakan Extensible Markup Language (XML) yang sudah terjadi standar industri IT, platform .Net memungkinkan seseorang developer unutk membuat aplikasi windows dan web menggunakan bahasa pemrograman apapun yang mendukung .Net dan aplikasi tersebut akan bisa digunakan oleh semua (aplikasi) client yang mendukung penggunaan .Net



Gambar 2.1 Platform Microsoft .Net

Platform .Net terdiri dari lima komponen utama dalam tiga lapisan (layer) :

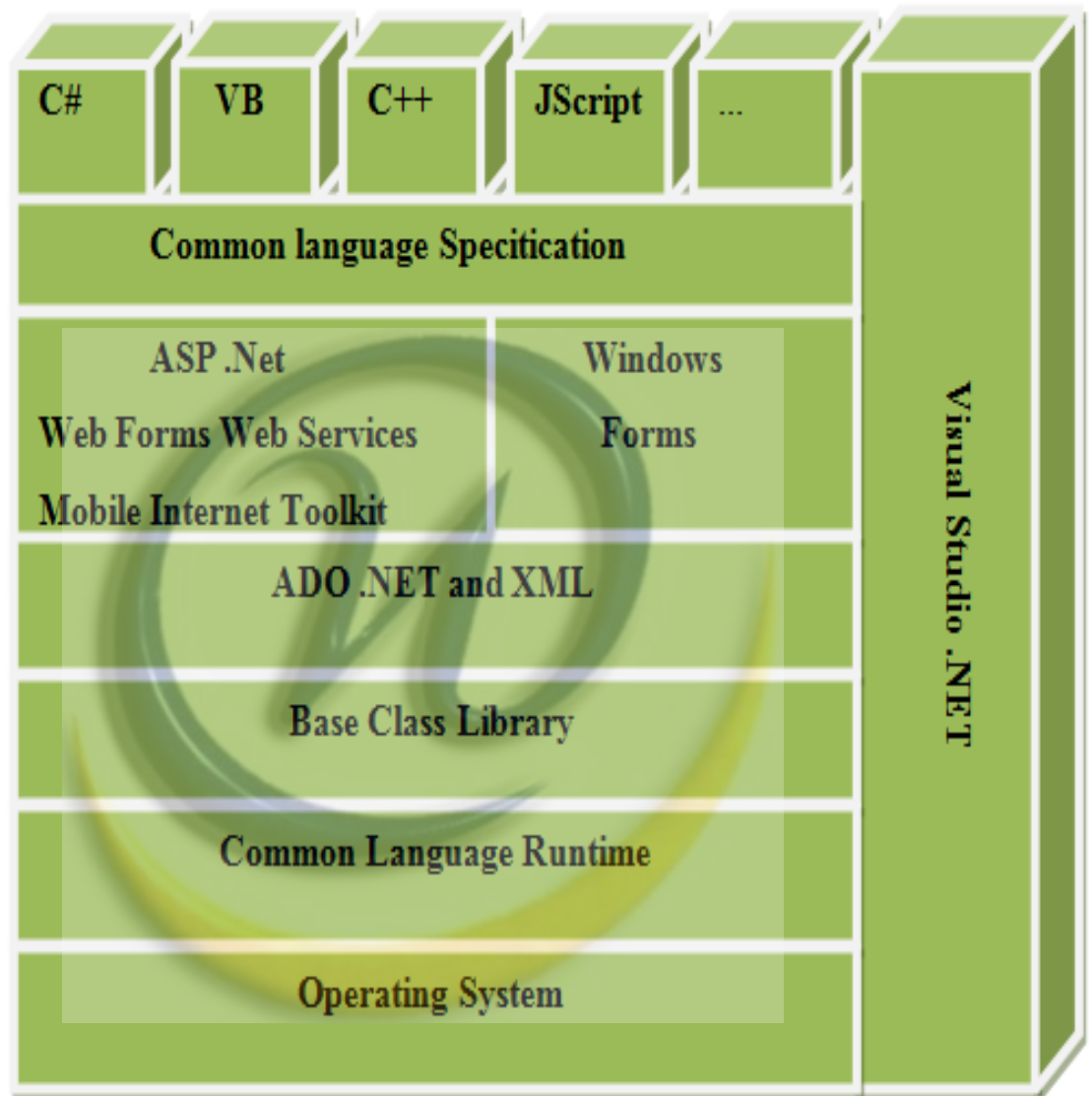
1. Lapisan terbawah adalah sistem operasi yang merupakan sistem operasi Windows 32 bit, termasuk windows XP, Windows 2003, Windows 2000, Windows ME, dan Windows CE.
2. Lapisan kedua terdiri dari tiga produk ;
 - a. .Net Enterprise Server, dipakai untuk waktu yang dibutuhkan untuk membuat sistem bisnis skala besar SQL Server, Application Center, BizTalk Server, Commerce Server, dan lain-lain.
 - b. .Net Building Block Services, merupakan distributed programmable service yang tersedia secara online dan offline. Service bisa dijalankan di komputer stand-alone ataupun diakses menggunakan Internet. .Net Building Block Services bisa digunakan dari platform apa saja yang mendukung SOAP.
 - c. .Net Framework, merupakan pusat dari platform .Net. Termasuk didalamnya adalah Common Language Runtime (CLR).

2.1.2 Pengertian .Net Framework

Microsoft .Net Framework menurut Juninda dalam buku *Panduan Lengkap Menjadi Programmer* adalah sebuah komponen yang dapat ditambahkan ke system operasi Microsoft Windows atau telah terintegritas ke dalam Windows (mulai dari Windows server 2003 dan versi-versi Windows Terbaru). Kerangka kerja ini menyediakan sejumlah besar solusi-solusi program untuk memenuhi kebutuhan-kebutuhan umum suatu program baru, dan mengatur eksekusi program-program yang ditulis secara khusus untuk framework ini. .Net Framework adalah kunci penawaran utama dari Microsoft, dan dimaksudkan untuk digunakan oleh sebagian besar aplikasi-aplikasi baru yang dibuat untuk platform windows.

Defenisi .Net Framework secara formal adalah platform yang memungkinkan kita untuk membangun aplikasi dan library yang disebut dengan “*managed applications*”. .Net Framework menyediakan compiler dan

tools agar kita dapat membangun, debug, dan mengeksekusi managed applications.



Gambar 2.2 *Detail Microsoft .Net*

Jadi .Net adalah platform yang memberikan kita segala sesuatu yang diperlukan untuk membangun dan menjalankan Managed Application yang berjalan di Windows.

Managed Application adalah eksekusi dari aplikasi tersebut yang diatur oleh .Net Framework yang menyediakan lingkungan runtime yang terkendali dan menyediakan sangat banyak variasi service seperti loading (memuat) aplikasi, mengatur memori, dan monitoring sekuritas dan integritas ketika aplikasi dijalankan sehingga lebih mudah dipelihara dan didebug.

Pada dasarnya .Net terdiri dari dua komponen utama ;

1. Common Language Runtime (CLR)
2. .Net Framework Base Class Library

2.1.3 Visual Basic 2005 atau VB .NET

Visual Basic 2005 memiliki konstruksi dan user interface (antarmuka) yang mirip dengan Visual Basic versi sebelumnya. Seperti bahasa sehari-hari visual basic .Net memiliki sintaks dan beberapa kata-kata yang valid yang biasa digunakan dalam membuat aplikasi. Visual Basic merupakan pilihan yang populer bagi yang mulai belajar pemrograman karena sintaks penulisan kodenya mudah dibandingkan dengan bahasa pemrograman lain.

Visual Basic 2005 memiliki banyak fasilitas baru dan ditingkatkan seperti inheritance, interface, dan overloading yang menjadikannya sebagai bahasa pemrograman berorientasi objek yang tangguh. Object Oriented Programming (OOP) merupakan kumpulan objek yang saling berinteraksi satu dengan lainnya. OOP akan mendekomposisikan masalah dunia nyata dan dinamakan class ataupun tipe.

Yang Baru di VB 2005

Visual Basic 2005 merupakan perbaikan daripada Visual Basic .NET versi 2002 dan 2003.

Berikut adalah fitur baru di Visual Basic 2005 :

Fitur	Penjelasan
Fitur Umum	
Perbaikan IDE (<i>Integrated Development Environment</i>)	Manajer eksepsi (kesalahan program), code snippets dengan pengisian secara otomatis, perbaikan IntelliSense dan AutoCorrect
Kontrol Baru	Form Windows memiliki sejumlah kontrol baru. Beberapa memang baru dan lainnya menggantikan kontrol yang sudah ada sebelumnya dan menawarkan tambahan kemampuan. Contoh kontrol baru yaitu : Kontrol WebBrowser, MaskedTextBox, TableLayoutPanel

	I, dan FlowLayoutPanel
Data Binding	Beberapa kontrol baru untuk data binding dan desainer baru yang mendukung seperti Data Sources untuk drag and drop field ke dalam Form Windows.
ClickOnce	Teknologi deployment baru untuk deploying melalui internet dengan update secara otomatis
Edit dan Lanjutkan	Anda dapat membuat perubahan kode ketika anda menjalankannya dalam IDE dan efek perubahan akan terjadi secara langsung
Fitur Kode Bahasa	
Class – class Partial	Memungkinkan kode class dipecah menjadi beberapa modul kode
Generics	Memungkinkan collection generic menangani tipe

	yang spesifik, yang dideklarasikan ketika collection dibuat
Class – class “My”	Menyediakan akses cepat ke class – class yang sering digunakan dalam Framework .NET
Type Nullable	Memungkinkan tipe data yang mempunyai nilai atau menjadi null, menjadikan tipe .NET cocok dengan tipe database
Operator Overloading	Membolehkan anda membuat operasi di antara tipe yang berubah – ubah
Keyword IsNot	Menyederhanaka n pernyataan if yang mengecek apakah objek <i>nothing</i>
Keyword Using	Secara otomatis mengatur objek yang dibuat dalam seksi kode

Tabel 2.1 Fitur baru pada Visual Basic 2005

Edisi Visual Basic 2005

VB 2005 merupakan rilis ketiga dan terbaru dari perbaikan besar – besaran bahasa VB, secara internal dikenal sebagai VB versi 8.0 dan datang dengan pembaruan paralel di Framework .NET versi 2.0

VB yang termasuk dalam Visual Studio 2005 memiliki beberapa edisi yaitu :

- a. **Edisi Express** merupakan produk *entry-level*/tingkat awal yang ditujukan untuk pelajar dan orang yang melakukan pemrograman sebagai hobi. Paket ini meliputi suatu antarmuka lingkungan pengembangan yang disederhanakan dan beberapa pembatasan fungsionalitas/kemampuan. Disediakan fitur untuk membantu pengembang pemula menjadi lebih produktif
- b. **Edisi Standar** menggunakan lingkungan pengembangan yang disederhanakan seperti edisi Express tetapi ditambahkan beberapa fungsionalitas ekstra. Penambahan meliputi dokumentasi MSDN (*Microsoft Developer Network*) lengkap, desainer class, dukungan penuh untuk membangun aplikasi form – from Windows, fitur XML yang lebih kaya, mendukung pengintegrasian kontrol – kode – sumber, deployment aplikasi yang baru “ClickOnce”, dan kemampuan mengakses ke SQL Server reporting service.
- c. **Edisi Professional** meliputi semua fitur dari edisi standard tetapi ditambahkan fitur yang lebih ditingkatkan untuk pengembang fulltime. Antarmuka pengguna yang disederhanakan digantikan dengan IDE yang penuh. Paket juga meliputi Crystal Reports, mendukung untuk *remote debugging*, kemampuan untuk menghasilkan aplikasi 64-bit, akses yang penuh ke database dan services sistem, serta deployment penuh menggunakan proyek Installer Windows dan ClickOnce. Disertakan juga SQL Server 2005 Express Edition.

- d. **Tool for Microsoft Office System.** Produk ini mirip dengan edisi Professional tetapi ditambahkan tool – tool yang membuat pengembangan dengan MS Office menjadi lebih mudah.
- e. **Team System.** Produk ini secara aktual merupakan tiga target terpisah yang ditargetkan pada :
 1. Arsitek perangkat lunak
 2. Pengembang perangkat lunak
 3. Tester / penguji perangkat lunak

Paket keempat adalah “**suite**” yang mengombinasikan semua dari tiga target dicatas. Semua paket dirancang untuk proyek dengan banyak pengembang dan meliputi tool untuk menguji dan *profiling* aplikasi .NET. kontrol kode sumber dan tool manajemen proyek juga disertakan. Suatu paket terpisah, **Visual Studio 2005 Team Foundation Server**, adalah suatu produk sisi – server yang menyediakan tambahan kolaboratif dan dukungan fitur untuk semua anggota regu.

Lingkungan Visual Basic 2005

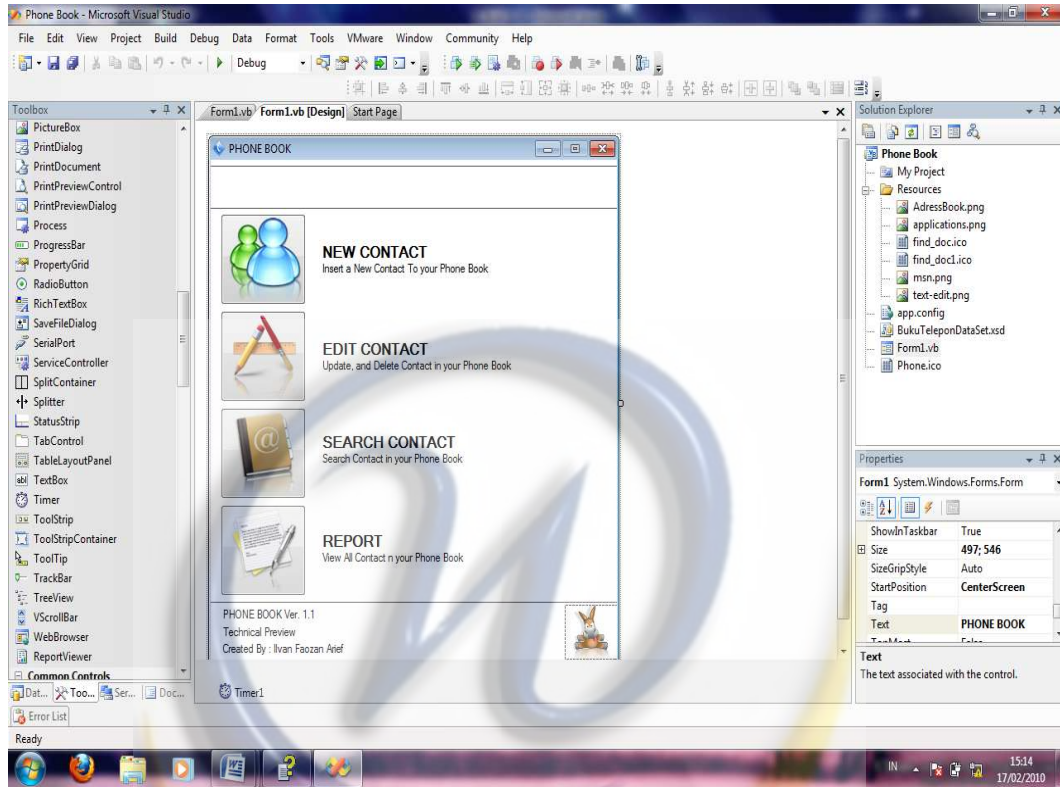
Untuk mengembangkan aplikasi, Visual Studio 2005 menyediakan IDE (*Integrated Development Environment* / lingkungan pengembangan terintegrasi). Lingkungan menyediakan tool – tool untuk mendesain, mengeksekusi dan men-debug (mencari kesalahan program).

Perbedaan Solution dengan Proyek

Pada Visual Basic 6, keseluruhan program aplikasi disebut sebagai proyek dan biasanya akan menghasilkan sebuah file .exe atau .dll.

Proyek adalah grup file – file yang akan menghasilkan suatu output yang spesifik. Output mungkin berupa program executable hasil kompilasi, file .dll (dynamic link library) dari class – class untuk digunakan pada proyek lainnya, atau kontrol custom (dibuat oleh pengguna) yang akan digunakan pada form Windows.

Solution adalah group dari satu atau lebih proyek yang akan dikelola bersama.



Gambar 2.3 IDE Visual Studio 2005

Berikut pembahasan bagian – bagian dari IDE Visual Studio 2005.

Mengenal Menu dan Toolbar

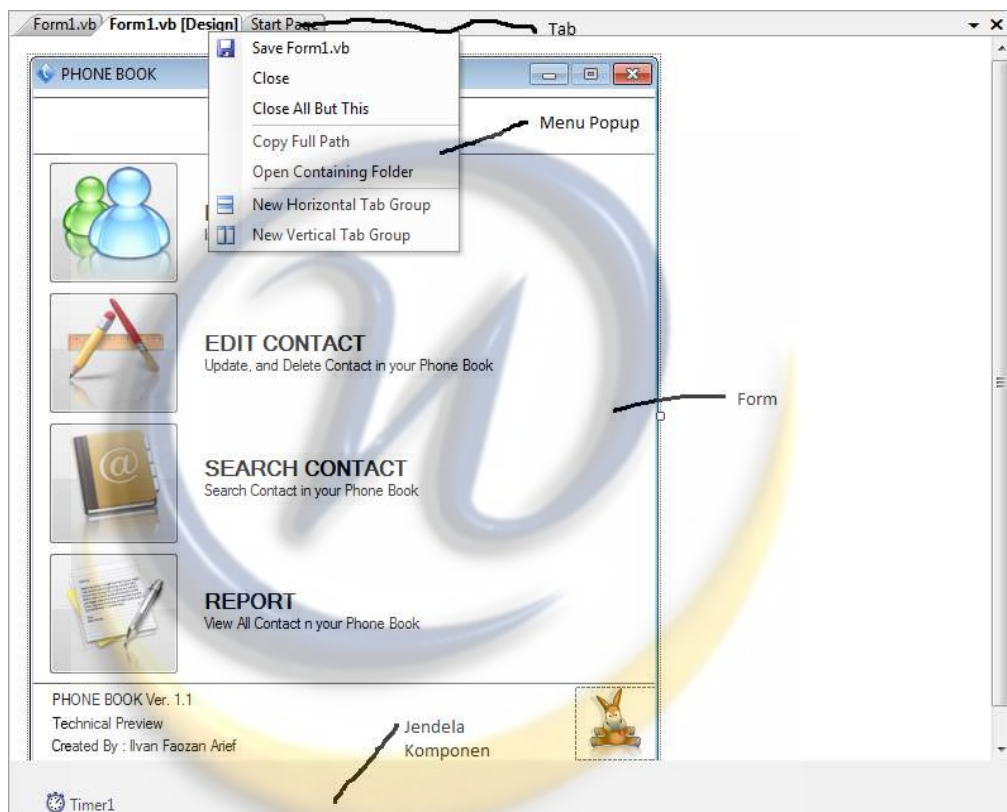


Gambar 2.4 Menu dan Toolbar Visual Studio 2005

Menu berisi perintah standar Visual Studio. Menu bersifat dinamik, berubah tergantung objek apa yang sekarang anda pilih.

Toolbar berupa kumpulan tombol ikon dan label yang dapat digunakan untuk mengakses dengan cepat fitur yang ada di menu. Di bawah menu terdapat toolbar standar.

Menjelajahi Jendela Desainer



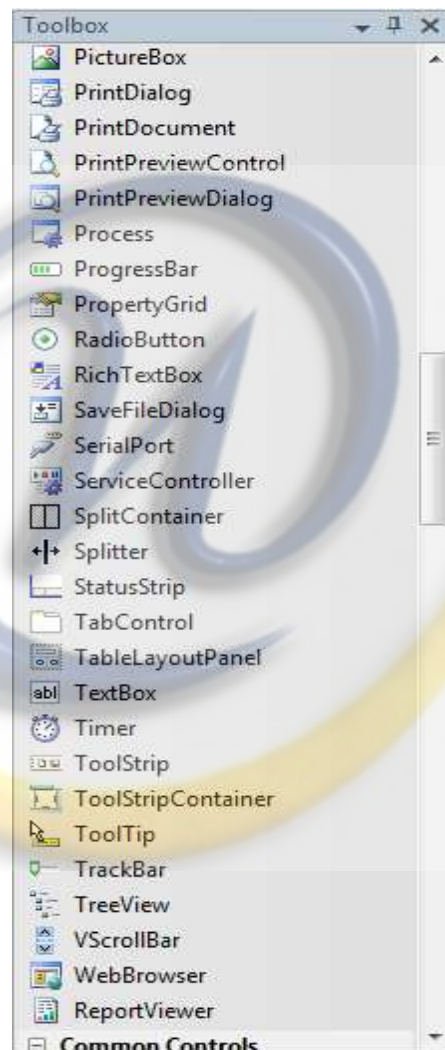
Gambar 2.5 Jendela Desainer

Di tengah lingkungan Visual Studio 2005 terdapat jendela desainer. Jendela ini merupakan pusat pengembangan aplikasi Visual Basic 2005. Di bagian atas jendela terdapat tab berisi form dengan nama NamaForm.vb {Design}. Form adalah tempat pengguna “menggambar” aplikasi. Kode yang berhubungan dengan form terdapat pada tab NamaForm.vb. Tab yang berwarna abu – abu menunjukkan file yang sudah dibuka tapi tidak aktif.

Lambang asterik (*) pada sebelah kanan nama file berarti membuat perubahan tetapi file tersebut belum tersimpan.

Di jendela desainer bagian bawah atau disebut jendela komponen diletakkan komponen yaitu objek Visual Basic 2005 yang tidak memiliki antar muka.

Mengakses Kontrol dan Komponen Menggunakan Toolbox



Gambar 2.6 Jendela Toolbox

Toolbox berisi kontrol dan komponen yang akan digunakan pada aplikasi. Untuk menambah kontrol dan komponen ke form di jendela desainer, hanya perlu menyeretnya dari Toolbox. Toolbox biasanya terlihat di sisi kiri lingkungan Visual Studio dan di sebelah kiri jendela desainer.

Toolbox bersifat proyek sensitif, artinya kontrol untuk halaman Web dan smart devices tidak akan muncul ketika sedang menulis suatu proyek Form Windows. Karena kontrol itu tidak digunakan dalam tipe proyek itu, maka Toolbox tidak akan terlihat.

Komponen dapat ditambahkan dengan cara berikut :

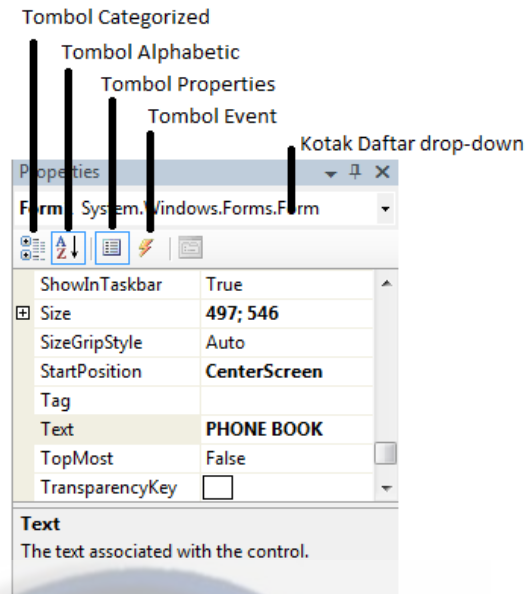
- a. Klik kontrol dan drag/menyeretnya ke form, jatuhkan ke tempat yang diinginkan.
- b. Klik ganda kontrol, dan objek akan muncul di sudut kiri atas dari form aktif.

Jika Toolbox tidak terlihat, dapat membukanya dengan memilih menu **View > Toolbox** atau dengan menekan **Ctrl + Alt + X**.

Kontrol dan komponen dalam Toolbox dikelompokkan dalam kategori – kategori. Kategori terletak pada divider / pemisah yang berwarna kelabu dan berisi label nama kategori. Untuk membuka suatu kategori yang tertutup, klik di palang kategori

Mengubah Detail Menggunakan Jendela Properties

Jendela Properties digunakan pada mode desain untuk mengatur suatu nilai pada objek (kontrol dan komponen yang sudah diletakkan di form atau jendela komponen). Kotak drop-down pada bagian atas jendela berisi daftar semua objek pada form sekarang. Disediakan dua tampilan yang dapat dipilih dengan cara mengklik tombol di bawah kotak drop-down yaitu : Alphabetic (huruf abjad) dan Categorized (dikelompokkan berdasar kategori tertentu). Tombol Event dengan gambar petir dapat digunakan untuk menentukan apa yang dilakukan kontrol di bawah kondisi tertentu – suatu pola yang disebut *events*. Di bawahnya terdapat properti yang disediakan untuk objek yang aktif (dipilih). Untuk membuka Jendela Properties, dapat menggunakan **View > Properties** atau menekan tombol fungsi **F4**.

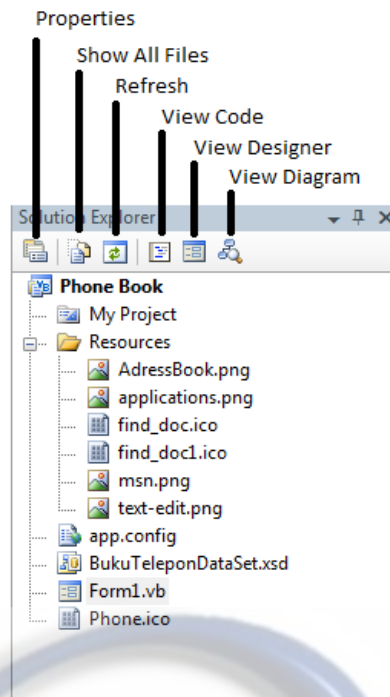


Gambar 2.7 Jendela Properties

Tabel yang berada di dalam Jendela Properties berisi dari kontrol. Properti berada di sebelah kiri dan nilainya terletak di sebelah kanan. Untuk mengubah suatu nilai biasanya dengan mengetik dan menekan Enter. Beberapa properti memiliki tombol elipsis (...) dan jika diklik akan membuka kotak dialog untuk mengubah nilai properti yang detailnya kompleks.

Mengorganisasikan Proyek Menggunakan Solution Explorer

Solution dan proyek menangani form dan komponen seperti file – file dalam suatu folder. Solution Explorer adalah tool Visual Studio yang dapat digunakan dalam mengelola file – file penyusun proyek. Jendela Solution Explorer menampilkan daftar semua form, modul dan file lainnya untuk membuat aplikasi. Untuk menampilkan suatu file di jendela, klik-ganda pada ikon atau nama file. Untuk mengganti nama, menyalin atau menghapus file, klik mouse kanan pada file dan pilih aksi dalam menu popup yang muncul. Dalam Solution Explorer, dapat juga membuat folder baru dan memindahkan file ke dalamnya

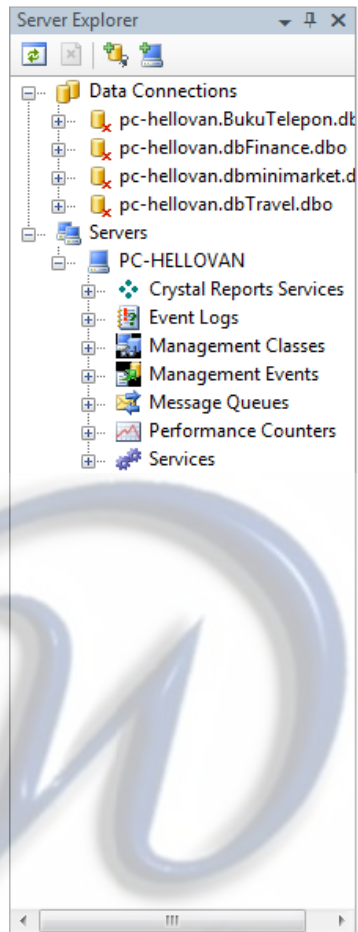


Gambar 2.8 Jendela Solution Explorer

Berikut ini tombol di bagian atas Solution Explorer, dari kiri ke kanan sebagai berikut :

- a. **Properties** : Untuk membuka jendela Properties
- b. **Show All Files** : membuka file – file yang tersembunyi, terutama di proyek form Web
- c. **Refresh** : Memeriksa folder solution untuk file baru yang mungkin telah ditambahkan oleh tool lainnya
- d. **View Code** : Membuka file yang terpilih di jendela kode
- e. **View Desidner** : Membuka file yang terpilih di jendela desainer
- f. **View Diagram** : Membuka file yang terpilih di jendela diagram

Akses Sumber Luar dengan Server Explorer



Gambar 2.9 Jendela Server Explorer

Server Explorer dapat dibuka dengan mengklik menu **View > Server Explorer** atau klik tab Server Explorer di sebelah bawah jendela Toolbox. Server Explorer memberi akses ke sumber lokal atau remote dari perspektif manajemen dan kode. Tree view yang berada dalam Server Explorer dapat menampilkan banyak server dan di bawah tiap server terdapat potongan fungsional yang dapat di akses.

Service / layanan yang ada dalam Server Explorer antara lain :

- a. Crystal Report Services (berkaitan dengan pembuatan laporan)
- b. Event Logs
- c. Message Queues (mengelola sejumlah permintaan ke aplikasi yang besar)

- d. Performance Counters
- e. Services (service dari panel kontrol), dan
- f. Data Connection (SQL Server dan database lainnya yang berjalan di server terdapat disini)

2.2 Metodologi

Dalam penyusunan tugas akhir ini metodologi yang penulis gunakan adalah Metodologi yang berbasis OOP (Object Oriented Programming). Dengan demikian penjelasan mengenai system ini dapat digambarkan dengan flow diagram, use case diagram, ERD dan class diagram.

2.2.1 OOP (Object Oriented Programming)

Objek Oriented Programming (OOP) adalah suatu metode dalam pembuatan program dengan tujuan untuk menyelesaikan kompleks berbagai masalah program yang terus meningkat. OOP tidak seperti pendahulunya (Pemrograman Terstruktur), OOP mencoba melihat permasalahan lewat pengamatan dunia nyata dimana setiap objek adalah entitas tunggal yang memiliki *kombinasi struktur data* dan *fungsi* tertentu. Ini kontras dengan pemrograman terstruktur dimana struktur data dan fungsi didefinisikan secara terpisah dan tidak berhubungan secara erat.

Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat. **Object - oriented programming** terdiri dari objects yang berinteraksi satu sama lainnya untuk menyelesaikan sebuah tugas. Contoh anggap kita memiliki sebuah perusahaan yang memiliki manager, sekretaris, petugas administrasi, dan lain – lain. Misal manager ingin memperoleh data dari bagian administrasi maka manager tidak harus mengambil berkas tersebut

langsung, tetapi dapat menyuruh petugas bagian administrasi untuk mengambilnya. Pada kasus ini seorang manager tidak harus mengetahui bagaimana cara mengambil data tersebut tetapi manager bisa mendapatkan data tersebut melalui objek petugas administrasi.

Dalam pemrograman berbasis objek, kita akan selalu menemui istilah-istilah berikut :

- a. Kelas – kumpulan atas defenisi data dan fungsi-fungsi dalam satu unit untuk suatu tujuan tertentu. Sebagai salah satu contoh '*class of dog*' adalah suatu unit yang terdiri atas defenisi-defenisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah class adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi objek.
- b. Objek – membungkus data dan fungsi bersama menjadi satu unit dalam sebuah program komputer; objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.
- c. Abstraksi – kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.
- d. Enkapsulasi – memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya dan objek lain tidak akan mengetahui kepada representasi dalam objek tersebut.
- e. Polimorfisme – melalui pengiriman pesan dan tidak tergantung pada pemanggilan subrutin, bahasa orientasi objek dapat mengirim

pesan; metode tertentu yang menghubungkan dengan sebuah pengiriman pesan tergantung kepada objek tertentu dimana pesan tersebut dikirim.

- f. Inheritas – mengatur polimorfisme dan enkapsulasi dengan mengizinkan objek didefinisikan dan diciptakan dengan jenis khusus dari objek yang sudah ada – objek-objek ini dapat membagi dan memperluas perilaku tanpa harus mengimplementasikan ulang perilaku tersebut (bahasa berbasis objek tidak selalu memiliki inheritas).

2.2.2 Konsep Dalam Pemrograman Objek

Jantung pemrograman berbasis objek adalah konsep objek, sebuah konstruksi pemrograman yang merupakan kombinasi spesifik dan seperangkat kelakuan yang saling berhubungan. Dalam OOP, skenario dimulai dari objek yang bersangkutan dibandingkan dengan langkah demi langkah yang dilakukan oleh bahasa pemrograman terstruktur.

Konsep dasar OOP dalam pemrograman visual basic :

- a. Class, Objek dan Instance

Istilah yang banyak digunakan dalam pemrograman berbasis objek adalah class, objek, instance. Sebuah class merupakan template bagi objek. Analogi yang menggambarkan hubungan antara class, objek, instance adalah ketika kita memikirkan istilah rumah dan bangunan rumah. Pada analogi tersebut, class adalah cetak biru dari rumah dan rumah itu sendiri adalah objek. Banyak rumah yang dapat dibangun dengan cetak biru yang sama dan banyak objek yang dapat dibuat dari class yang sama, dan setiap objek yang dibuat dari class disebut instance dari class.

Objek dibentuk oleh anggota-anggotanya dan anggota dari objek dapat berupa properti, field, methods dan events dan disajikan

dalam bentuk fungsionalitas yang membentuk objek. Sedangkan method merupakan aksi yang dilakukan oleh sebuah objek dan event merupakan notifikasi sebuah objek untuk melakukan tindakan tertentu. Class dapat mendefinisikan properti yang dapat dimiliki oleh sebuah instance. Properti merupakan sebuah nilai yang ada sebagai bagian dari objek dan kita dapat memanggilnya atau mengaturnya melalui sebuah objek. Atribut merupakan status objek dan *behavior* merupakan tingkah laku dari objek tersebut.

Class merupakan prototipe, blue print, ataupun template dengan kata lain kelas adalah representasi abstrak dari suatu objek, sedangkan objek adalah representasi nyata dari Kelas ataupun perwujudan (instance) dari suatu kelas.

b. Inheritance

Mekasnisme inheritance memungkinkan kita untuk memanfaatkan fungsionalitas yang terdapat dikelas yang telah didefenisikan kedalam kelas baru dan mengimplementasikan anggota yang berbeda jika diperlukan. Sebuah class dapat mengindukkan langsung kesatu kelas yang disebut base class. Penggunaan inheritance dalam OOP untuk mengklasifikasikan objects dalam program sesuai karakteristik umum dan fungsinya. Hal ini juga membuat programming lebih mudah karena memungkinkan kita untuk mengkombinasikan karakteristik umum kedalam objec parent dan mewariskan ini ke child object.

c. Override

Diperlukan untuk memodifikasi atau menambah metode terhadap metode yang diwariskan oleh kelas induk. Hal tersebut dapat dilakukan, dan mekanisme tersebut dikenal dengan istilah override.

d. Constructor

Constructor merupakan sebuah routine (prosedur) yang dipanggil pada saat suatu kelas instance dibuat. Di Visual Basic .NET, constructor disajikan dalam bentuk prosedur NEW ketika mendefinisikan class. Constructor dapat dioverride. Constructor berguna untuk menginisialisasi sebuah objek.

e. Overload

Overload digunakan untuk membuat beberapa metode dengan nama yang sama tetapi memiliki implementasi yang berbeda. Metode-metode yang overload harus memiliki ciri khas yang membedakan diantaranya metode-metode tersebut tetapi metode-metode tersebut dapat memiliki return type dan akses level yang sama.

f. Encapsulasi

Encapsulasi merupakan proses penyembunyian atau pembungkusan tentang detail dari bagaimana sebuah objek melakukan kegiatannya ketika diminta untuk melakukan tugas-tugas sebuah client. Keuntungan dari mekanisme ini adalah :

1. Client dapat menggunakan fungsi dari sebuah objek tanpa harus mengetahui bagaimana objek tersebut bekerja.
2. Setiap perubahan yang terjadi didalam sebuah objek tidak diketahui oleh client.

Sebagai contoh, Komputer, pada saat menyalakan, mematikan, pembacaan data, pemrosesan data maka pemakai/user tidak perlu tahu detail dari bagaimana proses menyalakan dan mematikan, itu dilakukan oleh komputer, cukup tahu bagaimana cara menyalakan/mematikan komputer, keyboard, mouse, tombol power merupakan interface (antar muka) pemakai dengan sistem komputer tersebut.

g. Polimorfisme

Polimorfisme sebuah kata dari bahasa Yunani yang mempunyai arti banyak bentuk. konsep ini dimungkinkan untuk menggunakan suatu interface yang sama agar suatu objek melakukan aksi atau tindakan yang mungkin secara prinsip sama tapi secara proses bisa berbeda-beda. Pada umumnya konsep ini sering kali disebut dalam istilah satu interface banyak aksi. Polimorfisme sering dinamakan dengan dynamic binding, date binding, maupun runtime binding.

Contoh yang nyata adalah komputer yang ada dipasaran terdiri atas berbagai tipe dan merk, namun semuanya memiliki interface yang sama, seperti: tombol power, keyboard, mouse, monitor dan sebagainya. Jika seseorang dapat mengoperasikan komputer satu jenis saja dari merk tertentu, maka orang itu akan dapat mengoperasikan hampir seluruh jenis komputer yang ada, karena semua komputer tersebut menggunakan interface yang sama. Perlu diketahui disini bahwa interface yang sama tidak berarti cara kerjanya juga sama. Misal kecepatan/speed komputer, kapasitas memory penyimpanan, jika dibandingkan komputer.

2.3 Microsoft SQL Server 2005

Microsoft SQL Server diperkenalkan pada tahun 1990 untuk platform Microsoft OS/2 dalam kerjasamanya dengan Sybase. Produk ini berasal dari Sybase SQL Server 4.x untuk platform Unix. Dengan adanya Windows NT, muncul inisiatif untuk membangun SQL Server versi Windows NT sehingga dihasilkan Microsoft SQL Server versi 4.2 untuk platform Windows NT. Kerjasama dengan Sybase masih berlanjut dan diluncurkan SQL Server 6.0 pada tahun 1995 dan setahun kemudian SQL Server versi 6.5 diluncurkan.

SQL Server 6.5 memperbarui kemampuan transaksi dan menjadi produk database client/server yang banyak dipakai pada platform Windows NT. Untuk memenuhi kebutuhan pengguna yang makin meningkat, maka SQL Server perlu didisain ulang dan kerjasama dengan Sybase dihentikan.

Kemudian Microsoft mengembangkan SQL Server 7.0 yang difokuskan pada tiga area yaitu : easy to use, scalability dan data warehousing. Pada tahun 2000, kemudian Microsoft meluncurkan SQL Server 2000. Di tahun 2005 ini, Microsoft mengeluarkan produk SQL Server versi terbarunya yaitu Microsoft SQL Server 2005 seiring dengan diluncingnya Microsoft Visual Studio 2005 beta 2.

SQL Server 2005 adalah RDMBS (*Relational Database Management System*) dengan arsitektur client server yang disertai dengan berbagai komponen dan service/layanan yang menjadikannya platform yang komprehensif (memiliki cakupan luas) untuk aplikasi enterprise.

Komponen SQL Server 2005

SQL Server 2005 disusun oleh komponen dan services berikut :

Komponen	Penjelasan
Engine (mesin) database relational	<i>Engine</i> database merupakan komponen inti dari SQL Server yang menyediakan lingkungan untuk menyimpan, menampilkan, dan memodifikasi data dalam format relational dan XML
Analysis Services	Memberikan dasar dari solusi intelegensi bisnis yang didukung oleh aplikasi OLAP

	<p>(<i>Online Analyst Processing</i>) yaitu kemampuan untuk menangani wuery yang kompleks melalui akses data secara multidimensi dan <i>data mining</i> yaitu menganalisa data untuk menemukan pola – pola yang tersembunyi</p>
<p>SSIS (<i>SQL Server Integration Services</i>)</p>	<p>Engine untuk membuat solusi impor dan solusi expor data</p>
<p>Notificati on Services</p>	<p><i>Framework</i> (kerangka kerja) untuk solusi dimana <i>subscribers</i> (pelanggan) menerima notifikasi ketika server yang spesifik terjadi. Notifikasi dapat dikirim ke berbagai tipe perangkat</p>
<p>Reporting Services</p>	<p>Digunakan untuk mengekstrak data dari SQL Server dan membuat laporan</p>
<p>Service</p>	<p>Mekanisme antrian untuk komunikasi</p>

Broker	transaksi berbasis-pesan antar-software services
.NET CLR (<i>Common Language Runtime</i>)	Memungkinkan diimplementasikan solusi database menggunakan <i>managed code</i> (kode terkelola) yang ditulis menggunakan bahasa pemrograman .NET seperti C# dan Visual Basic
Native HTTP Support	Memungkinkan aplikasi klien dikoneksikan ke endpoints HTTP dalam SQL Server tanpa menggunakan IIS (<i>Internet Information Services</i>)
<i>Replication</i>	Serangkaian teknologi untuk menyalin dan mendistribusikan data dan objek database dari suatu database atau server ke lainnya dan melakukan sinkronisasi antara database untuk memastikan konsistennya
<i>Full Text</i>	Memungkinkan

<i>Search</i>	melakukan pengindeksan secara cepat dan fleksibel berdasarkan kunci query dari data teks yang disimpan dalam database SQL Server
---------------	--

Tabel 2.2 Daftar Komponen Penyusun SQL Server 2005

Engine Database

SQL Server 2005 memiliki sejumlah perbaikan pada engine database seperti dijelaskan pada tabel berikut :

Fitur / Perbaikan	Penjelasan
Partitioning	Tabel dan indeks dapat dipartisi melalui <i>multiple group file</i> . Ini akan meningkatkan kinerja dan <i>maintainability</i> (kemampuan untuk dikelola)
Trigger DDL (<i>Data Definition Language</i>) dan notifikasi	Trigger DDL dapat digunakan untuk menjalankan <i>stored procedures</i> ketika pernyataan

event	DDL (seperti CREATE) dieksekusi. Notifikasi event dapat digunakan untuk memicu event dari service / layanan Service Broker ketika database dimodifikasi
Tipe data baru	Sejumlah tipe data baru seperti xml dan max , perubahan pada tipe nvarchar , varchar , varbinary
MARS (<i>Multiple Action Result Sets</i>)	MARS memperbolehkan driver client untuk memiliki lebih dari sebuah request yang di-pending (ditunda) per koneksi
Transact-SQL	Transact-SQL sekarang memiliki kata kunci untuk penanganan eksepsi / kesalahan terstruktur (try dan catch). Mendukung <i>Common Table Expressions</i> dan

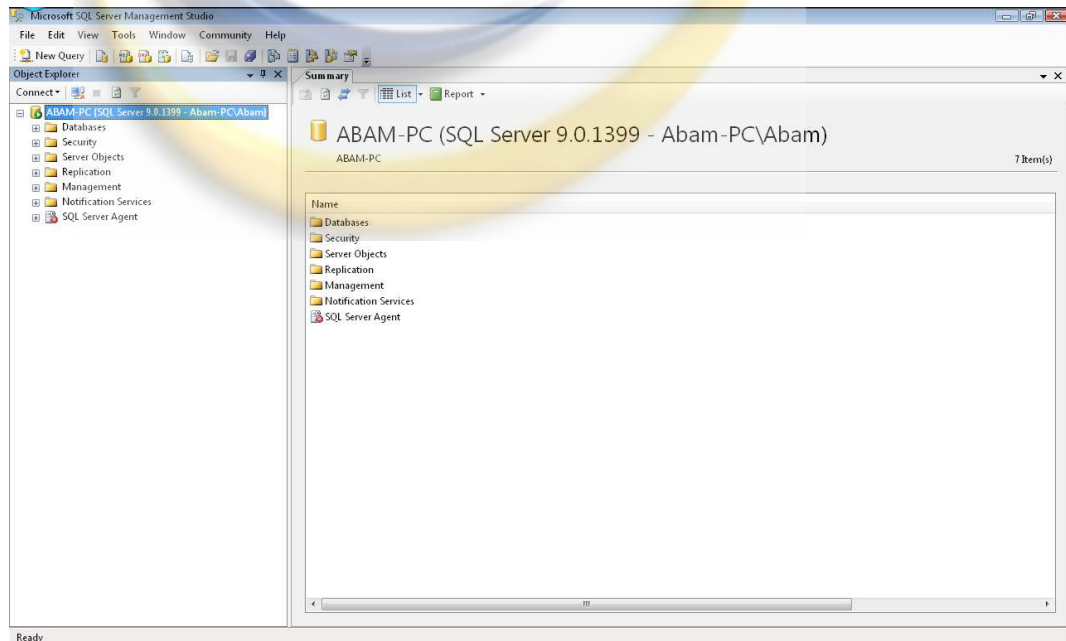
	fungsi ranking baru serta operator relational
Sekuriti	Arsitektur sekuriti baru menyediakan perbaikan proteksi database melalui <i>principals</i> , <i>securables</i> , dan <i>permissions</i>
XML	Engine database sekarang menggunakan tipe data XML untuk menampilkan dan memodifikasi data XML menggunakan ekspresi Xquery
Framework .NET	CLR (<i>Common Language Runtime</i>) Framework .NET diintegrasikan ke dalam engine database SQL Server, memungkinkan membangun solusi database dengan <i>managed code</i> / kode terkelola
Replication	Mendukung replication transaksional <i>peer-to-peer</i> ,

	replication over HTTP, dan perbaikan pada replication <i>heterogeneous</i>
Pengelolaan Database	<i>Maintainability</i> diperbaiki dengan operasi <i>online restore</i> dan <i>online index</i>

Tabel 2.3 Daftar Perbaikan Engine Database SQL Server 2005

SQL Server Management Studio

SQL Server Management Studio adalah lingkungan terintegrasi untuk mengakses, melakukan konfigurasi, mengelola, dan melakukan administrasi semua komponen SQL Server 2005. Tool ini merupakan gabungan kemampuan dari Enterprise Manager, Query Analyzer, dan Analysis Manager yang ada pada SQL Server sebelumnya.



Gambar 2.10 Jendela SQL Server Management Studio

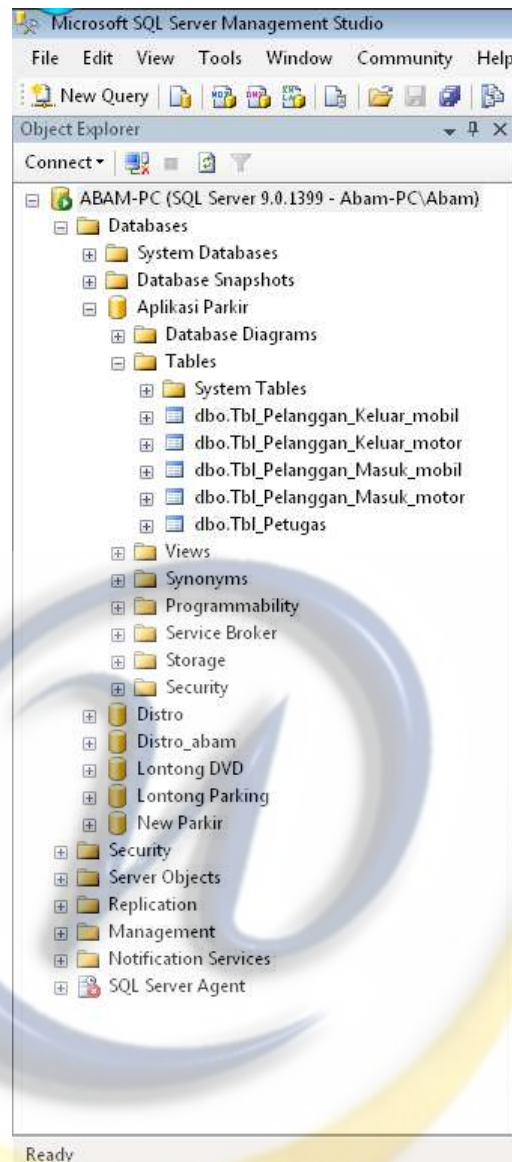
SQL Server Management Studio menyediakan fitur berikut untuk administrator database :

- a. Mengelola secara penuh database relational, database analysis services, reporting services dan database SQL Server CE
- b. Tool visual untuk membuat pernyataan : Transact-SQL (T-SQL), XMLA, MDX, dan Xquery

Ketika membuat query atau scripts, SQL Server Management Studio menggunakan framework Microsoft Visual Studio dan memiliki fungsionalitas Visual Studio. Kontrol sumber mendukung untuk memasukkan dan mengelola salinan script yang berkembang setiap saat, dan mengakses sistem bantu interaktif. Di dalamnya terdapat Object Explorer suatu panel untuk melakukan navigasi server dan database. Dan Solution Explorer suatu panel untuk mengelola solution SQL Server.

Object Explorer

Object Explorer digunakan untuk mengelola object dalam database SQL Server. Object Explorer menampilkan object dalam struktur hierarkial menggunakan suatu *tree view* yang dikelompokkan berdasarkan tipe objek.



Gambar 2.11 Jendela Object Explorer

Object Explorer adalah fitur dalam SQL Server Management Studio yang berbentuk jendela dan dapat menggunakan untuk mem-browse dan mengelola objek. Menggunakan Object Explorer, dapat melakukan tugas sebagai berikut:

1. Membuat dan mengelola database dan objek
2. Menampilkan dan memodifikasi properti database
3. Men-*generate scripts* untuk membangun kembali isi database
4. Membuat sumber data

5. Mengontrol *priviledes*/hak akses dan *permissions*/izin
6. Mengonfigurasi replication
7. Membuat paket DTS (*Data ransformation Services*)
8. Menampilkan file log SQL Server
9. Mengelola agent SQL Server

Object Explorer juga dapat digunakan untuk mengelola item – item sebagai berikut :

1. **Database.** Menangani databse sistem dan pengguna dalam SQL Server yang dikoneksikan.
2. **Security.** Menyediakan akses ke *server login, server roles, linked – serve,r* dan *remote server*.
3. **Server Object.** Menampilkan detail dari objek seperti device backup, dan menampilkan daftar server yang di – link ketika suatu server dikoneksikan ke server remote lainnya.
4. **Replication.** Digunakan untuk menampilkan detail yang berkaitan dengan replikasi data dari database pada server ke database lainnya atau kebalikannya.
5. **Management.** Menyediakan akses ke file log SQL Server, devices backup, perencanaan pemeliharaan database, dan monitor aktivitas.
6. **Notification Services.** Mengirim notifikasi yang dipicu oleh perubahan data atau objek melalui media seperti email atau sms.
7. **SQL Server Agents.** Folder SQL Server Agents digunakan untuk mengelola *job/kerja, alert/pesan, operator, proxi, dan log error* dari SQL Server Agents.

Di sebelah atas Object Explorer terdapat tombol yang dapat digunakan untuk mengontrol operasi yaitu :

- a. **Connect**, tombol ini akan menampilkan kotak dialog **Connect to Server**, yang memungkinkan melakukan koneksi ke *intstance* SQL Server.
- b. **Disconnect**, tombol ini akan menutup koneksi pada server sekarang dipilih

- c. **Stop**, kontrol tree view secara dinamik akan memperluas folder yang dipilih. Suatu folder yang berisi sejumlah besar objek akan memerlukan waktu yang lama untuk ditampilkan. Tombol ini digunakan untuk menghentikan Object Explorer dari melakukan operasi yang membutuhkan waktu lama.
- d. **Refresh**, tombol akan me-refresh informasi yang ditampilkan di Object Explorer.
- e. **Filter**, tombol ini memungkinkan untuk memilih filter objek yang akan ditampilkan oleh Objec Explorer.

Perintah –Perintah T-SQL

Transact-SQL atau T-SQL, adalah implementasi Microsoft dari SQL (*Structure Query Language*), yang dirancang untuk mendapat kembali, memanipulasi, dan menambahkan data ke RDBMS (*Relational Database Management System*).

SQL adalah subbbahasa untuk membuat dan memanipulasi data dalam database.

Pernyataan T-SQL dikelompokkan menjadi tiga yaitu :

- a. **DDL** (*Data Definition Language*) – Perintah yang digunakan untuk membuat dan mengelola objek dalam suatu database. Pernyataan DDL dapat digunakan untuk membuat, memodifikasi, dan mendrop/menghapus database, tabel, indeks, *view*, *stored procedures*, dan objek yang lain.
- b. **DCL** (*Data Control Language*) – Pernyataan untuk mengendalikan izin keamanan untuk objek database dan para pengguna. Beberapa objek diatur mempunyai izin yang berbeda.
- c. **DML** (*Data Manipulation Language*) – Berisi pernyataan yang digunakan untuk bekerja dengan data. Ini meliputi pernyataan untuk mendapatkan kembali data, menyisipkan baris ke dalam suatu tabel, memodifikasi nilai – nilai, menghapus baris, dan melakukan perhitungan terhadap data.

2.4 Pengertian UML (*Unified Modeling Language*)

UML merupakan bahasa pemodelan yang paling sukses dari tiga metode OO yang telah ada sebelumnya, yaitu Booch, OMT dan OOSE. UML merupakan kesatuan dari ketiga metode pemodelan tersebut dan ditambah kemampuan lebih untuk mengatasi masalah pemodelan yang tidak bisa ditangani ketiga metode tersebut. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. Menurut Mathilda Ostling dalam bukunya '*A Sequence Diagram for BlueJ*' menuliskan bahwa :

UML adalah bahasa pemodelan *visual* yang terdiri dari sembilan diagram. Suatu pemodelan dari sistem *software* bisa dijelaskan sebagai sebuah gambaran abstrak dari sistem tersebut. Kesembilan diagram dalam *UML* ini digunakan untuk menggambarkan, mengspesifikasikan, mengartikan, mendokumentasikan perbedaan komponen-komponen dari sistem sebelum diimplementasikan. *UML* digunakan untuk membuat "*blueprints*" dari sistem *software*, hal ini diperlukan dalam proyek-proyek besar untuk memperoleh sebuah tinjauan sistem. (5:3)

UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan system yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.

2.5 Bagian-bagian UML

Bagian-bagian utama dari UML adalah *view*, *diagram*, *model element*, dan *general mechanism*.

2.5.1 View

View digunakan untuk melihat system yang dimodelkan dari beberapa aspek yang berbeda. *View* bukan melihat grafik, tapi merupakan suatu abstraksi yang berisi sejumlah diagram.

Beberapa jenis *view* dala UML antara lain : *use case viewm logical view*, *component view*, *concurrency view*, dan *deployment view*.

2.6.1.1 Use case view

Mendeskripsikan fungsionalitas sistem yang seharusnya dilakukan sesuai yang diinginkan *external actors*. *Actor* yang berinteraksi dengan sistem dapat berupa user atau sistem lainnya.

View ini digambarkan dalam *use case diagrams* dan kadang-kadang dengan *activity diagrams*. *View* ini digunakan terutama untuk pelanggan, perancang (*designer*), pengembang (*developer*), dan penguji sistem (*tester*).

2.6.1.2 Logical view

Mendeskripsikan bagaimana fungsionalitas dari sistem, struktur statis (*class, object, dan relationship*) dan kolaborasi dinamis yang terjadi ketika *object* mengirim pesan ke *object* lain dalam suatu fungsi tertentu.

View ini digambarkan dalam *class diagrams* untuk struktur statis dan dalam *state, sequence, collaboration, dan activity diagram* untuk model dinamisnya. *View* ini digunakan untuk perancang (*designer*) dan pengembang (*developer*).

2.6.1.3 Component view

Mendeskripsikan implementasi dan ketergantungan modul. Komponen yang merupakan tipe lainnya dari *code module* diperlihatkan dengan struktur dan ketergantungannya juga alokasi sumber daya komponen dan informasi administrative lainnya.

View ini digambarkan dalam *component view* dan digunakan untuk pengembang (developer).

2.6.1.4 Concurrency view

Membagi sistem ke dalam proses dan prosesor. *View* ini digambarkan dalam diagram dinamis (*state*, *sequence*, *collaboration*, dan *activity diagrams*) dan diagram implementasi (*component* dan *deployment diagrams*) serta digunakan untuk pengembang (developer), pengintegrasian (integrator), dan pengujian (tester).

2.6.1.5 Deployment view

Mendeskripsikan fisik dari sistem seperti komputer dan perangkat (*nodes*) dan bagaimana hubungannya dengan lainnya.

View ini digambarkan dalam *deployment diagrams* dan digunakan untuk pengembang (developer), pengintegrasian (integrator), dan pengujian (tester).

2.5.2 Diagram

Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu *view* tertentu dan ketika digambarkan biasanya dialokasikan untuk *view* tertentu. Adapun jenis diagram antara lain :

2.5.2.1 Use Case Diagram

Use case adalah abstraksi dari interaksi antara system dan actor. Use case bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah system dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah system dipakai. *Use case* merupakan konstruksi untuk mendeskripsikan bagaimana system akan terlihat di mata user. Sedangkan use case diagram memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan client.

2.5.2.2 Class Diagram

Class adalah dekripsi kelompok obyek-obyek dengan property, perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya class diagram dapat memberikan pandangan global atas sebuah system. Hal tersebut tercermin dari class- class yang ada dan relasinya satu dengan yang lainnya. Sebuah sistem biasanya mempunyai beberapa *class diagram*. Class diagram sangat membantu dalam visualisasi struktur kelas dari suatu system.

2.5.2.3 Component Diagram

Component software merupakan bagian fisik dari sebuah system, karena menetap di komputer tidak berada di benak para analis. Komponen merupakan implementasi software dari sebuah atau lebih class. Komponen dapat berupa *source code*, komponen biner, atau *executable component*. Sebuah komponen berisi informasi tentang logic class atau class yang diimplementasikan sehingga membuat pemetaan dari *logical view* ke *component view*. Sehingga component diagram merepresentasikan dunia riil yaitu component software yang mengandung component, interface dan relationship.

2.5.2.4 Deployment Diagram

Menggambarkan tata letak sebuah system secara fisik, menampakan bagian-bagian software yang berjalan pada bagian-bagian hardware, menunjukkan hubungan komputer dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya. Di dalam *nodes*, *executeable component* dan *object* yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh *node* tertentu dan ketergantungan komponen.

2.5.2.5 State Diagram

Menggambarkan semua *state* (kondisi) yang dimiliki oleh suatu *object* dari suatu *class* dan keadaan yang menyebabkan *state* berubah. Kejadian dapat berupa *object* lain yang mengirim pesan. *State class* tidak digambarkan untuk semua *class*, hanya yang mempunyai sejumlah *state* yang terdefinisi dengan baik dan kondisi *class* berubah oleh *state* yang berbeda.

2.5.2.6 Sequence Diagram

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah scenario. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

2.5.2.7 Collaboration Diagram

Menggambarkan kolaborasi dinamis seperti *sequence diagrams*. Dalam menunjukkan pertukaran pesan, *collaboration diagrams* menggambarkan *object* dan hubungannya (mengacu ke konteks). Jika penekannya pada waktu atau urutan gunakan *sequencediagrams*, tapi jika penekanannya pada konteks gunakan *collaboration diagram*.

2.5.2.8 Activity Diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.

2.6 Tujuan Penggunaan UML

- a. Memberikan bahasa pemodelan yang bebas dari berbagai bahas pemrograman dan proses rekayasa.
- b. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.
- c. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
- d. UML bisa juga berfungsi sebagai sebuah (blue print) cetak biru karena sangat lengkap dan detail. Dengan cetak biru ini maka akan bisa diketahui informasi secara detail tentang koding program atau bahkan membaca program dan menginterpretasikan kembali ke dalam diagram (reverse engineering).

2.7 Pengertian Hotel [22]

Menurut beberapa pengertian, Hotel didefinisikan sebagai berikut:

Hotel berasal dari kata hostel, konon diambil dari bahasa Perancis kuno. Hotel adalah suatu bidang usaha yang menggunakan suatu bangunan atau sebagian bangunan yang disediakan secara khusus, untuk setiap orang yang menginap, makan, memperoleh pelayanan dan menggunakan fasilitas lainnya dengan pembayaran. (Wikipedia : <http://id.wikipedia.org/wiki/Hotel> 25/06/09 14:20)*Menurut Dirjen Pariwisata:*

Depparpostel Hotel adalah suatu jenis akomodasi yang mempergunakan sebagian atau seluruh bangunan, untuk menyediakan jasa penginapan, makan dan minum, serta jasa lainnya bagi umum, yang dikelola secara komersial.

Menurut Surat Keputusan Menteri Perhubungan R.I No. PM 10/PW – 301/Phb. 77, tanggal 12 Desember 1977:

Hotel adalah suatu bentuk akomodasi yang dikelola secara komersial, disediakan bagi setiap orang untuk memperoleh pelayanan penginapan, berikut makan dan minum. Menurut Webster Hotel adalah suatu bangunan atau suatu lembaga yang menyediakan kamar untuk menginap, makan dan minum serta pelayanan lainnya untuk umum.

Seiring perkembangan zaman dan bertambahnya pemakai jasa, layanan inap-makan ini mulai meninggalkan misi sosialnya. Guest pun dipungut bayaran. Sementara bangunan dan kamar-kamarnya mulai ditata sedemikian rupa agar membuat guest betah. Meskipun demikian, bertahun-tahun standar layanan hotel tak banyak berubah.

Sampai pada tahun 1793, saat City Hotel dibangun di cikal bakal wilayah kota New York. City Hotel itulah pelopor pembangunan penginapan gaya baru yang lebih fashionable. Sebab, dasar

pembangunannya tak hanya mementingkan letak yang strategis. Tapi juga pemikiran bahwa hotel juga tempat istirahat yang mumpuni. Jadi, tak ada salahnya didirikan di pinggir kota.

Setelah itu, muncul hotel-hotel legendaris seperti Tremont House (Boston, 1829) yang selama puluhan tahun dianggap sebagai salah satu tempat paling top di Amerika Serikat (AS). Tremont bersaing ketat dengan Astor House, yang dibangun di New York, 1836. Saat itu, hotel modern identik dengan perkembangan lalu lintas dan tempat beristirahat. Saat pembangunan jaringan kereta api sedang gencar-gencarnya, hampir di tiap perhentian (stasiun) ada hotel.

Maksudnya jelas, untuk mengakomodasi orang-orang yang baru saja bepergian dengan kereta api. Karena masa itu naik kereta api sangat melelahkan, hotel-hotel pun “dipersenjatai” berbagai hiburan pelepas penat. Hotel jenis ini, diembeli-embeli dengan kata “transit”, karena memang ditujukan buat para musafir.

Seiring dengan berkembangnya teknologi dan makin luasnya jangkauan angkutan darat (terlebih setelah ditemukannya kendaraan bermotor), kawasan sekitar rel kereta api tak lagi menarik minat para investor. Orang kemudian lebih suka jalan-jalan pakai mobil ketimbang kereta. Kepopuleran hotel transit pun tersaingi oleh kehadiran “motel”, gabungan kata “motor hotel” alias tempat istirahat para pengendara kendaraan bermotor.

Kejayaan motel tak berlangsung lama. Seiring makin pesatnya perkembangan kota, berakhir pula era motel. Terutama karena letaknya yang agak di pinggir kota dan fasilitasnya yang kalah bagus dengan hotel di pusat kota. Kalaupun terpaksa bermalam di kawasan pinggiran, motel harus bersaing dengan hotel resort, yang banyak tumbuh di tempat-tempat peristirahatan.

Selain hotel, resort, anak-anak kandung hotel yang lahir di era 1990-an tak kalah hebatnya. Sebut saja berbagai extended-stay hotel, khusus buat guest yang membutuhkan tempat menginap minimal lima malam. Sedangkan pelaku bisnis yang harus bernegosiasi di kampung atau negeri orang, bisa mencari hotel apartment. Di Amerika, dua jenis hotel ini berkembang sangat pesat.

Di Indonesia, kata hotel selalu dikonotasikan sebagai bangunan penginapan yang cukup mahal. Umumnya di Indonesia dikenal hotel berbintang, hotel melati yang tarifnya cukup terjangkau namun hanya menyediakan tempat menginap dan sarapan pagi, serta guest house baik yang dikelola sebagai usaha swasta (seperti halnya hotel melati) ataupun mess yang dikelola oleh perusahaan-perusahaan sebagai tempat menginap bagi para guest yang ada kaitannya dengan kegiatan atau urusan perusahaan.

Biaya dan kualitas hotel biasanya berdasarkan jangkauan dan tipe pelayanan yang tersedia. Dikarenakan kenaikan besar-besaran dalam turisme di seluruh dunia, selama dekade terakhir abad 20, pendirian hotel terutama yang kecil telah meningkat secara drastis. Untuk perbandingan, sistem penilaian telah diperkenalkan satu sampai lima bintang dan juga tingkat melati di Indonesia yang lebih murah.

Secara harfiah, kata Hotel dulunya berasal dari kata HOSPITIUM (bahasa Latin), artinya ruang guest. Dalam jangka waktu lama kata hospitium mengalami proses perubahan pengertian dan untuk membedakan antara Guest House dengan Mansion House (rumah besar) yang berkembang pada saat itu, maka rumah-rumah besar disebut dengan HOSTEL. Rumah-rumah besar atau hostel ini disewakan kepada masyarakat umum untuk menginap dan beristirahat sementara waktu, yang selama menginap para penginap dikoordinir oleh seorang host, dan semua guest-guest yang (selama) menginap harus tunduk kepada peraturan yang

dibuat atau ditentukan oleh host (HOST HOTEL). Sesuai dengan perkembangan dan tuntutan orang-orang yang ingin mendapatkan kepuasan, tidak suka dengan aturan atau peraturan yang terlalu banyak sebagaimana dalam hostel, dan kata hostel lambat laun mengalami perubahan. Huruf “s” pada kata hostel tersebut menghilang atau dihilangkan orang, sehingga kemudian kata hostel berubah menjadi Hotel seperti apa yang kita kenal sekarang. [21]

Beberapa Jenis Hotel

Losmen	Destination hotel
Extended Stay Hotel	Flophouse
Hotel Bar	Garden hotels
Resort	Grand Hotel
All-inclusive resort	Holiday cottage
Apartment hotel	Hostal
Bed and breakfast	Hostel
Boutique hotel	Ice hotel
Capsule hotel	Inn
Caravanserai	Motel
Casa particular	Resort
Condo-hotel	