

BAB II

TINJAUAN PUSTAKA

II.1. Pengertian Sistem

Sistem merupakan salah satu yang terpenting dalam sebuah perusahaan yang dapat membentuk kegiatan usaha untuk mencapai kemajuan dan target yang dibutuhkan. Defenisi tentang sistem cukup banyak, untuk mengetahui lebih jelasnya tentang defenisi sistem ini diambil pernyataan dari beberapa ahli berikut :

Sistem merupakan serangkaian bagian yang saling tergantung dan bekerja sama untuk mencapai tujuan tertentu. Suatu sistem pasti tersusun dari sub-sub sistem yang lebih kecil yang juga saling tergantung dan bekerja sama untuk mencapai tujuan (Anastasia Diana dan Lilis Setiawati; 2011 : 3).

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu aturan tertentu. Sistem juga dapat diartikan sebagai urutan operasi kerja (tulis-menulis), yang biasanya melibatkan beberapa orang di dalam satu atau lebih departemen, yang diterapkan untuk menjamin penanganan yang seragam dari transaksi bisnis yang terjadi (Kusrini dan Andri Koniyo; 2007 : 5).

Sehingga kesimpulannya, sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Dengan demikian di dalam suatu sistem, komponen-komponen ini tidak dapat berdiri

sendiri-sendiri, tetapi sebaliknya, saling berhubungan hingga membentuk satu kesatuan sehingga tujuan sistem itu dapat tercapai.

II.2. Pengertian Informasi

Informasi adalah data yang sudah diolah menjadi sebuah bentuk yang berarti bagi pengguna, yang bermanfaat dalam pengambilan keputusan saat ini atau mendukung sumber informasi. Data belum memiliki nilai sedangkan informasi sudah memiliki nilai. Informasi dikatakan bernilai bila manfaatnya lebih besar dibanding biaya untuk mendapatkannya (Kusrini dan Andri Koniyo; 2007 : 7)

Informasi yang berkualitas memiliki tiga kriteria, yaitu :

1. Akurat (*accurate*)

Informasi harus bebas dari kesalahan, tidak bias ataupun menyesatkan. Akurat juga berarti bahwa informasi itu harus dapat dengan jelas mencerminkan maksudnya.

2. Tepat pada waktunya (*timeliness*)

Informasi yang datang pada penerima tidak boleh terlambat. Di dalam pengambilan keputusan, informasi yang sudah usang tidak lagi bernilai. Bila informasi datang terlambat sehingga pengambilan keputusan terlambat dilakukan, hal itu dapat berakibat fatal bagi perusahaan.

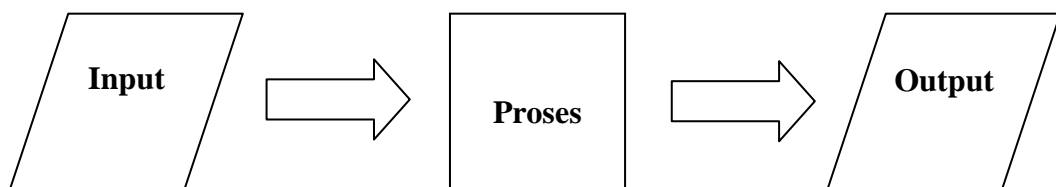
3. Relevan (*relevance*)

Informasi yang disampaikan harus mempunyai keterkaitan dengan masalah yang akan dibahas dengan informasi tersebut. Informasi harus bermanfaat bagi pemakainya.

II.3. Pengertian Sistem Informasi

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Robert A. Laitch dan K. Roscoe Bavis; 2007 : 8)

Sistem informasi, yang kadang kala disebut sebagai sistem pemrosesan data, merupakan sistem buatan manusia yang biasanya terdiri dari sekumpulan komponen (baik manual ataupun berbasis komputer) yang terintegrasi untuk mengumpulkan, menyimpan, dan mengelola data serta menyediakan informasi kepada pihak-pihak yang berkepentingan sebagai pemakai informasi tersebut (Anastasia Diana dan Lilis Setiawati; 2011 : 4).



Gambar II.1. Komponen Sistem Informasi
Sumber : (Anastasia Diana dan Lilis Setiawati ; 2011 : 4)

II.4. Sistem Informasi Akuntansi

Sistem informasi akuntansi merupakan sebuah sistem informasi yang mengubah data transaksi bisnis menjadi informasi keuangan yang berguna bagi pemakainya (Kursini, S.Kom & Andri Koniyo ; 2007 : 10).

Tujuan dari sistem informasi akuntansi adalah :

1. Mendukung operasi sehari-hari.
2. Mendukung pengambilan keputusan manajemen.
3. Memenuhi kewajiban yang berhubungan dengan pertanggungjawaban.

Komponen-komponen yang terdapat dalam sistem informasi akuntansi adalah sebagai berikut :

1. Orang-orang yang mengoperasikan sistem tersebut.
2. Prosedur-prosedur, baik manual maupun yang terotomatisasi, yang dilibatkan dalam pengumpulan, pemrosesan dan penyimpanan data aktivitas-aktivitas organisasi.
3. Data tentang proses-proses bisnis.
4. *Software* yang dipakai untuk memproses data organisasi.
5. Infrastruktur teknologi informasi.

II.5. Administrasi

Administrasi dapat di kelompokkan dalam beberapa kategori, secara makna sempit administrasi di definisikan sebagai sebuah kegiatan mencatat, mengumpulkan, dan menyimpan sesuatu kegiatan untuk penelusuran kembali sehingga dapat digunakan untuk pengambilan keputusan (Angga Sultoni; 2011).

II.6. Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual. Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek (Mulawarman ; 2011 : 1).

Tujuan pemanfaatan UML :

1. Menyediakan bagi pengguna (analisis dan desain sistem) suatu bahasa pemodelan visual yang ekspresif sehingga mereka dapat mengembangkan dan melakukan pertukaran model data yang bermakna.
2. Menyediakan mekanisme yang spesialisasi untuk memperluas konsep inti.
3. Karena merupakan bahasa pemodelan visual dalam proses pembangunannya maka UML bersifat independen terhadap bahasa pemrograman tertentu.
4. Memberikan dasar formal untuk pemahaman bahasa pemodelan.
5. Mendorong pertumbuhan pasar terhadap penggunaan alat desain sistem yang berorientasi objek (OO).
6. Mendukung konsep pembangunan tingkat yang lebih tinggi seperti kolaborasi, kerangka, pola dan komponen terhadap suatu sistem.
7. Memiliki integrasi praktik terbaik.

UML (*Unified Modeling Language*) adalah suatu alat Bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek (Munawar ; 2005 : 17). Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas visi

mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

Meskipun UML sudah banyak menyediakan diagram yang bisa membantu mendefinisikan suatu aplikasi, tidak berarti bahwa semua diagram tersebut akan bisa menjawab persoalan yang ada. Adapun tipe diagram UML yang ada seperti pada Tabel II.1.

Tabel II.1 Tipe Diagram UML

Diagram	Tujuan
Activity	Prilaku prosedural dan paralel
Class	Class, fitur dan relasinya
Communication	Interaksi diantara objek. Lebih menekankan kepada link
Component	Struktur dan koneksi dari komponen
Composite Structure	Dekomposisi sebuah class saat runtime
Deployment	Penyebaran/instalasi ke klien
Interaction Overview	Gabungan dari activity dan sequence diagram
Object	Contoh konfigurasi instance
Package	Struktur hierarki saat kompilasi
Sequence	Interaksi antara objek. Lebih menekankan pada urutan.
State Machine	Bagaimana event mengubah sebuah objek
Timing	Interaksi antar objek. Lebih menekankan pada waktu
Use Case	Bagaimana user berinteraksi dengan sebuah sistem

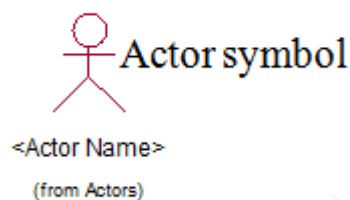
Sumber : (Munawar ; 2009 : 32)

II.6.1. Notasi Dasar UML

Berikut merupakan beberapa jenis notasi dasar UML yaitu:

1. *Actor*

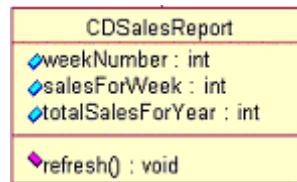
Actor menentukan peran yang dimainkan oleh user atau sistem lain yang berinteraksi dengan subjek. *Actor* adalah segala sesuatu yang berinteraksi langsung dengan sistem aplikasi komputer, seperti orang, benda atau lainnya. Tugas *actor* adalah memberikan informasi kepada sistem dan dapat memerintahkan sistem untuk melakukan sesuatu tugas. Berikut notasi *actor* dalam UML:



Gambar II.2. Notasi *Actor* pada UML
Sumber : (Mulawarman ; 2011 : 6)

2. *Class Diagram*

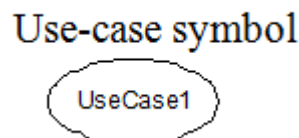
Class diagram yaitu notasi utama dan yang paling mendasar pada diagram UML adalah notasi untuk mempresentasikan suatu *class* beserta dengan atribut dan operasinya. *Class* adalah pembentuk utama dari sistem berorientasi objek.



Gambar II.3. Notasi *Class* pada UML
Sumber : (Mulawarman ; 2011 : 6)

3. Use Case

Use Case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use Case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah – langkah yang menerangkan antara pengguna dan sistem disebut skenario. Notasi *use case* dapat dilihat pada gambar II.4 :



Gambar II.4. Notasi *Use Case* pada UML
Sumber : (Mulawarman ; 2011 : 6)

II.6.2. Diagram UML

Berikut merupakan beberapa macam dari diagram UML yaitu:

1. *Class Diagram*

Class diagram menggambarkan struktur statis dari kelas dalam sistem dan menggambarkan atribut, operasi dan hubungan antara kelas. *Class diagram*

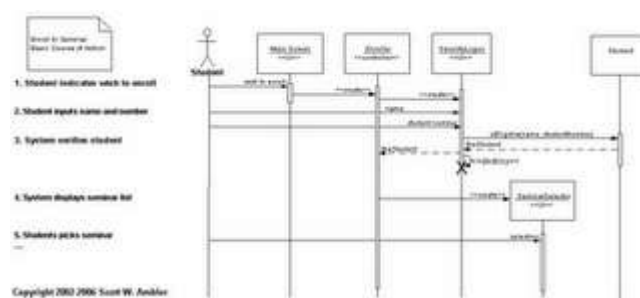
membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat.

Class memiliki tiga area pokok :

- a. Nama (dan *stereotype*)
- b. Atribut
- c. Metoda

2. *Sequence Diagram*

Sequence diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence diagram* adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case diagram*.

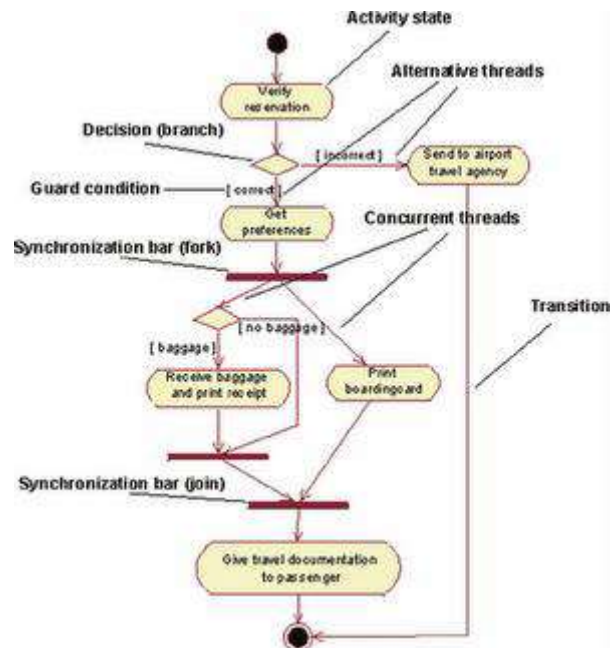


Gambar II.5. *Sequence Diagram*

Sumber : (Mulawarman ; 2011 : 5)

3. Activity Diagram

Menggambarkan aktifitas-aktifitas, objek, *state*, transisi *state* dan *event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas.



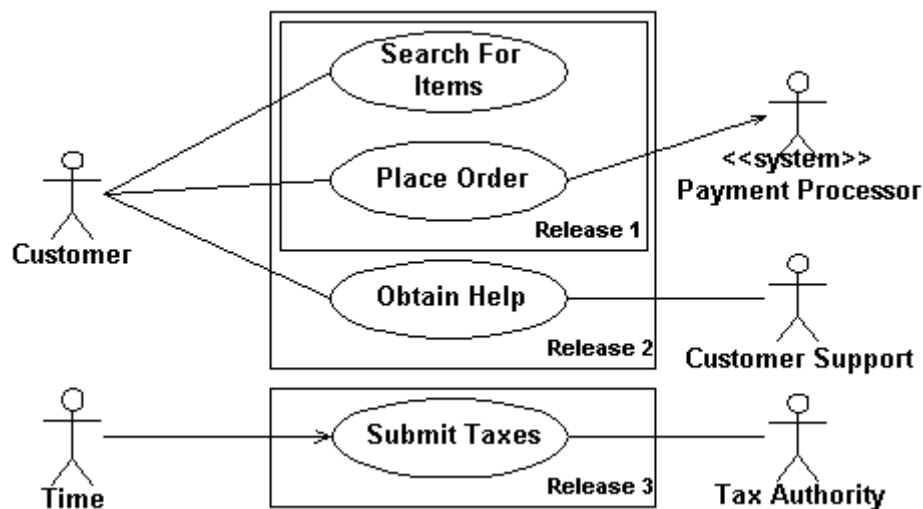
Gambar II.6. Activity Diagram
Sumber : (Mulawarman ; 2011 : 4)

4. Use Case Diagram

Diagram yang menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case* digambarkan sebagai elips horizontal dalam suatu diagram UML *use case*.

Use Case memiliki dua istilah :

- a. *System use case*; interaksi dengan sistem.
- b. *Business use case*; interaksi bisnis dengan konsumen atau kejadian nyata.



Gambar II.7. Notasi Actor pada UML
Sumber : (Mulawarman ; 2011 : 4)

II.7. Normalisasi

Normalisasi merupakan sebuah teknik dalam desain logika sebuah *database*, teknik pengelompokan atribut dari suatu relasi sehingga membentuk struktur relasi yang baik (tanpa redundansi). (Kusrini dan Andri Koniyo; 2007 : 99).

Kegunaan normalisasi :

1. Meminimalisasi pengulangan informasi.
2. Memudahkan identifikasi entitas/objek.

Langkah-langkah normalisasi :

1. Normal Pertama (1st *Normal Form*)

Aturan :

- a. Mendefinisikan atribut kunci.
- b. Tidak ada grup berulang.
- c. Semua atribut bukan kunci tergantung pada atribut kunci.

2. Normalisasi Kedua (2nd Normal Form)

Aturan :

- a. Sudah memenuhi bentuk normal pertama.
- b. Sudah tidak ada ketergantungan parsial dimana seluruh *field* hanya tergantung pada sebagian *field* kunci.

3. Normalisasi Ketiga (3rd Normal Form)

Aturan :

- a. Sudah berada dalam bentuk normal kedua.
- b. Tidak ada ketegantungan transitif (dimana *field* bukan kunci tergantung pada *field* bukan kunci lainnya).

II.8. *Entity Relationship Diagram (ERD)*

ERD merupakan notasi grafis dalam pemodelan data konseptual yang mendeskripsikan hubungan antar penyimpanan. ERD digunakan untuk memodelkan struktur data dan hubungan antar data, karena hal ini *relative kompleks*. Dengan ERD kita dapat menguji model dengan mengabaikan proses yang harus dilakukan. ERD dapat digambarkan secara lengkap dengan atribut-atributnya dan dapat pula digambar tanpa atribut. (Kusrini dan Andri Koniyo; 2007 : 98).

ERD menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antar data. Pada dasarnya ada 3 macam simbol yang digunakan, yaitu :

1. *Entity*

Suatu objek yang dapat diidentifikasi dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang akan dibuat. Entitas digambarkan dalam bentuk persegi empat.

2. Atribut

Entitas mempunyai elemen yang disebut dan berfungsi mendeskripsikan karakter entitas, misalnya atribut nama barang dari entitas barang. Setiap ERD bisa berisi lebih dari satu atribut. Atribut digambarkan dalam bentuk elips.

3. Hubungan/*Relationship*

Sebagaimana halnya entitas, hubungan pun harus dibedakan antara hubungan atau bentuk hubungan antar entitas dengan isi dari hubungan itu sendiri. *Relationship* digambarkan dalam bentuk intan (*diamonds*).

Jenis-jenis hubungan :

a. Satu ke Satu

Misalnya suatu perusahaan mempunyai aturan satu sopir hanya boleh menangani satu kendaraan karena alasan tertentu.

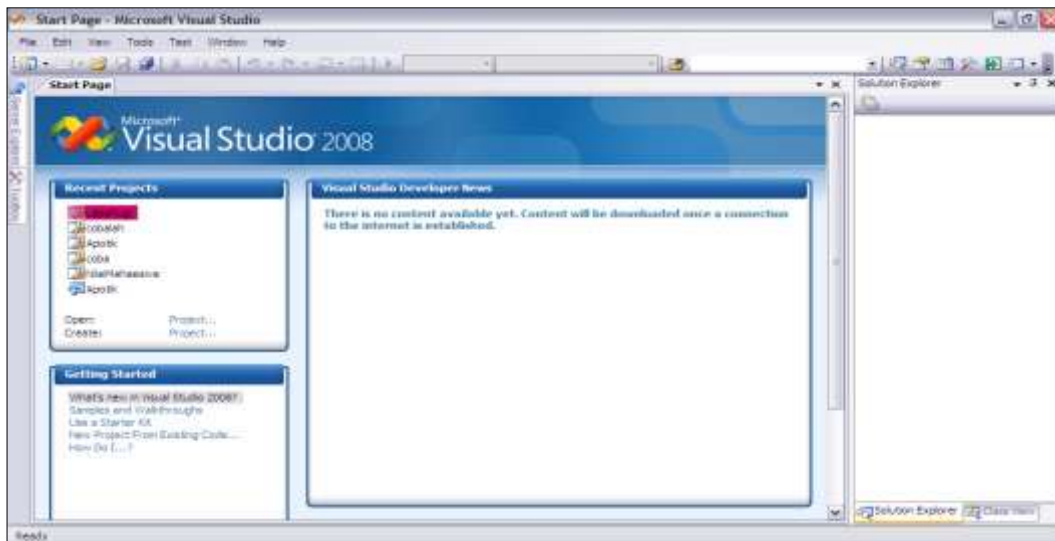
b. Satu ke Banyak atau Banyak ke Satu

Misalnya suatu perusahaan selalu berasumsi bahwa satu pelanggan dapat membeli banyak barang.

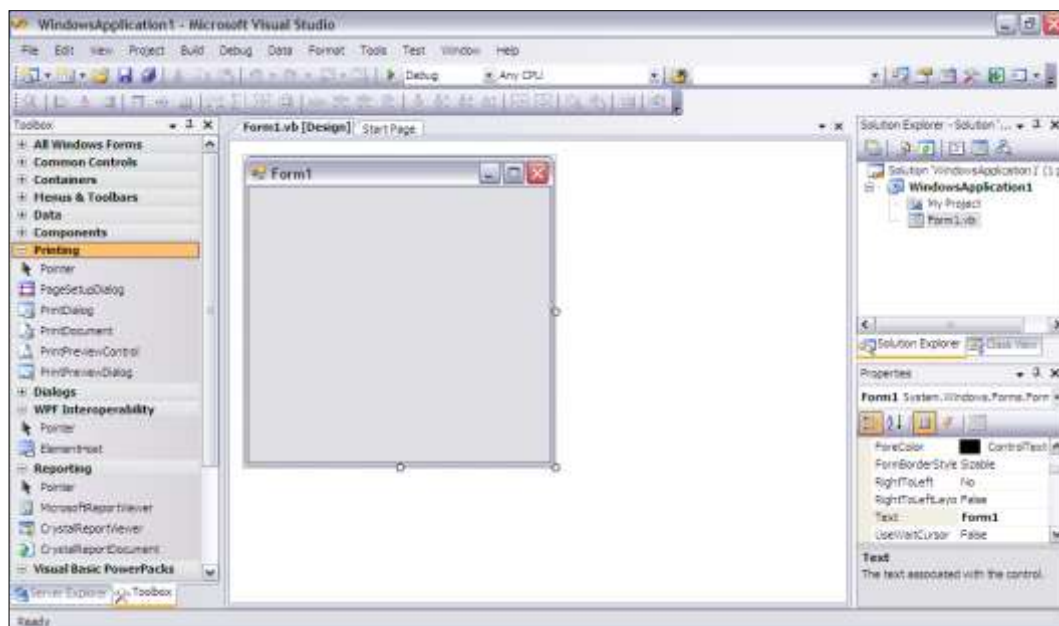
II.9. *Visual Basic.Net 2008*

Visual Basic.Net 2008 adalah suatu lingkungan (*environment*) terintegrasi untuk membangun dan melakukan uji coba (*testing and debugging*) berbagai macam aplikasi, contohnya aplikasi-aplikasi *Windows* (Wahana Komputer; 2007 : 1).

Pada dasarnya *Visual Basic.Net 2008* didesain untuk menampung berbagai macam bahasa pemrograman dan terlingkup dalam *Visual Basic.Net 2008*. Dengan demikian dapat dibangun aplikasi-aplikasi *Windows* di dalamnya. *Visual Basic.Net* juga merupakan salah satu bahasa pemrograman yang dapat digunakan untuk membuat program aplikasi. Program ini sangat mudah digunakan dan merupakan bahasa terpopuler untuk saat ini. Bagi para pengguna *Visual Basic 6* akan lebih mudah untuk berpindah ke *Visual Basic.Net 2008* daripada harus memilih bahasa pemrograman lainnya. Bahasa pemrograman ini menyediakan beberapa *tool* untuk otomatisasi proses pengembangan, yaitu *visual tool* yang digunakan untuk melakukan beberapa operasi pemrograman dan desain umum, dan juga fasilitas-fasilitas lain yang dapat menunjang dalam pemrograman.



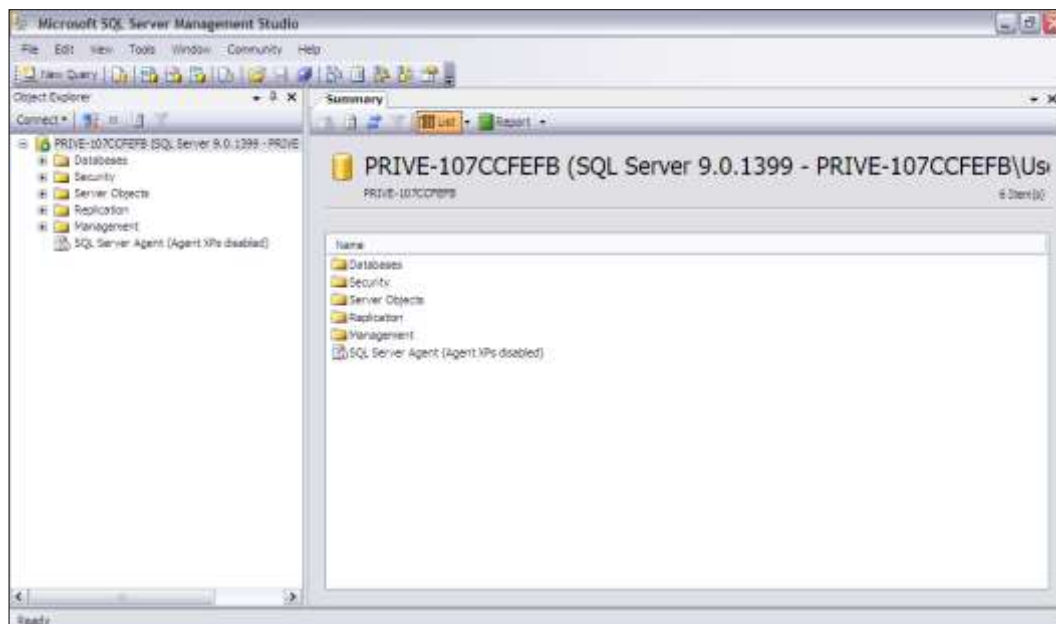
Gambar II.8. Tampilan Awal *Visual Basic.Net 2008*



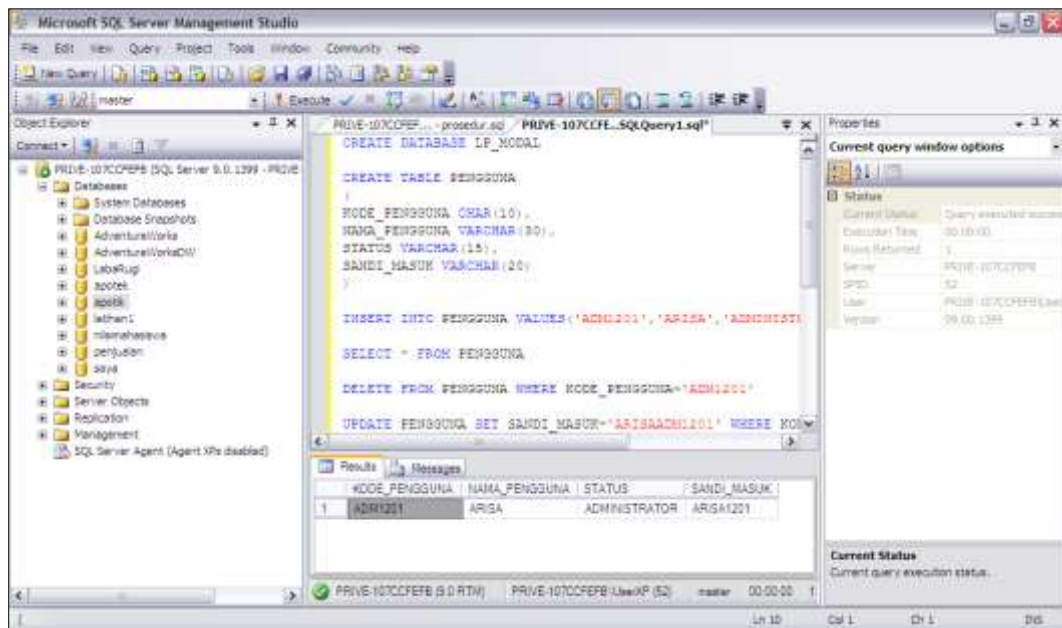
Gambar II.9. Jendela *Visual Basic.Net 2008*

II.10. *SQL Server 2005*

SQL Server 2005 adalah perangkat lunak *relational database management system* (RDBMS) yang didesain untuk melakukan proses manipulasi *database* berukuran besar dengan berbagai fasilitas. *Database* ini sangat *powerful* dan telah terbukti kekuatannya dalam mengolah data. *SQL Server 2005* juga memiliki banyak fitur yang bisa diandalkan untuk meningkatkan performa *database* (Aryo Nugroho, dkk; 2009 : 1).



Gambar II. 10. Tampilan Awal *SQL Server 2005*



Gambar II. 11. Lembar Kerja *SQL Server 2005*