

BAB II

TINJAUAN PUSTAKA

II.1. Pengertian Sistem

Menurut Asbon Hendra (2012;157) Sistem merupakan kumpulan dari unsure atau elemen-elemen yang saling berkaitan/berinteraksi dan saling memengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu.

Contoh :

1. Sistem computer, terdiri dari software, hardware, dan brainware.
2. Sistem akuntansi dan lain-lain.

Menurut Anastasia Diana dan LilisSetiawati (2011:3), Sistem merupakan serangkainya bagian yang saling tergantung dan bekerja sama untuk mencapai tujuan tertentu. Suatu system pasti tersusun dari sub-sub system yang lebih kecil yang juga saling tergantung dan bekerja sama untuk mencapai tujuan.

II.1.1. Karakteristik sistem

Menurut Asbon Hendra (2012;158), Sistem mempunyai beberapa karakteristik atau sifat-sifat tertentu antara lain :

1. Komponen sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama untuk membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupaya suatu subsistem atau bagian

dari sistem. Setiap sistem tidak peduli berapapun kecilnya, selalu mengandung komponen-komponen atau subsistem.

2. Batasan sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

3. Lingkungan luar sistem (*Environment*)

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat ini dapat bersifat menguntungkan atau merugikan.

4. Penghubung sistem (*Interface*)

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya untuk membentuk satu kesatuan sehingga sumber-sumber daya mengalir dari subsistem yang satu ke subsistem yang lainnya.

5. Masukan sistem (*Input*)

Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. masukan sinyal (*signal input*) adalah energi yang diproses untuk didapatkan keluaran.

6. Keluaran sistem (*Output*)

Keluaran adalah hasil dari energi yang diolah oleh system, meliputi output yang berguna, contohnya informasi yang dikeluarkan oleh computer. Dan output yang tidak berguna dikenal sebagai sisa pembuangan, contohnya panas yang dikeluarkan oleh computer.

7. Pengolah sistem (*Process*)

Merupakan bagian yang memproses masukan untuk menjadi keluaran yang diinginkan. Contoh CPU pada computer, bagian produksi yang mengubah bahan baku menjadi barang jadi, serta bagian akuntansi yang mengolah data transaksi menjadi laporan keuangan.

8. Tujuan sistem (*Objective*)

Setiap sistem pasti mempunyai tujuan ataupun sasaran yang memengaruhi input yang dibutuhkan dan output yang dihasilkan.

II.1.2. Klasifikasi sistem

Menurut Asbon Hendra (2012;160), Sistem dapat diklasifikasikan dari beberapa sudut pandangan, diantaranya sebagai berikut :

1. Sistem Abstrak (Abstract System).

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Misalnya, sistem teologia yang merupakan suatu sistem yang menggambarkan hubungan manusia dan Tuhan.

2. Sistem Fisik (Physical System).

Sistem fisik merupakan sistem yang ada secara fisik sehingga setiap makhluk dapat melihatnya. Contohnya sistem komputer, sistem akuntansi, sistem produksi, dan lain-lain.

3. Sistem Alamiah (Natural System).

Sistem alamiah adalah sistem yang terjadi karena proses alam (tidak dibuat manusia). Misalnya, sistem tata surya, sistem galaksi, dan lain-lain

4. Sistem Buatan Manusia (Human Made System).

Sistem buatan manusia adalah sistem yang dirancang oleh manusia. Contohnya, system informasi.

5. Sistem Tertentu (Deterministic System).

Sistem tertentu adalah sistem yang beroperasi dengan tingkah laku yang sudah diprediksi. Contohnya, sistem komputer.

6. Sistem Tak Tentu (Probabilistic System).

Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas. Contohnya, sistem manusia.

7. Sistem tertutup (Closed System).

Sistem tertutup adalah sistem yang tidak berhubungan dan tidak terpengaruh dengan sistem luarnya.

8. Sistem Terbuka (Open System).

Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya.

II.2. Pengertian Akuntansi

Suwardjono (2013:4), mengatakan kata akuntansi berasal dari kata bahasa Inggris *to account* yang berarti memperhitungkan atau mempertanggungjawabkan. Kata akuntansi sebenarnya diserap dari kata *accountancy* yang berarti hal-hal yang bersangkutan dengan *accountant* (akuntan) atau bersangkutan dengan hal-hal yang dikerjakan oleh akuntan dalam menjalankan profesinya.

Akuntansi adalah seni pencatatan, penggolongan dan peringkasan transaksi dan kejadian yang bersifat keuangan dengan cara yang berdaya guna dan dalam

bentuk satuan uang dan penginterpretasian hasil proses tersebut. Pengertian seni dalam definisi tersebut dimaksudkan untuk menunjukkan bahwa akuntansi bukan merupakan ilmu pengetahuan eksata atau sains karena dalam proses penalaran dan perancangan akuntansi banyak terlibat unsur pertimbangan (*judgment*). Seni dalam definisi diatas lebih mempunyai konotasi sebagai kerajinan dan keterampilan atau pengetahuan terapan yang isi dan strukturnya disesuaikan dengan kebutuhan untuk mencapai suatu tujuan.

Ismail (2010:2), mengatakan akuntansi adalah seni dalam mencatat, menggolongkan dan mengikhtisarkan semua transaksi-transaksi yang terkait dengan keuangan yang telah terjadi dengan suatu cara yang bermakna dan dalam satuan uang.

II.3. Pengertian Sistem Informasi Akuntansi

Menurut Anastasia Diana & Lilis Setiawati (2011;4) Sistem Informasi Akuntansi adalah sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan dengan transaksi keuangan.

II.3.1. Tujuan Sistem Informasi Akuntansi

Lingkup Sistem Informasi Akuntansi dapat dijelaskan dari manfaat yang didapat dari informasi akuntansi. Manfaat atau tujuan Sistem Informasi Akuntansi tersebut adalah sebagai berikut;

1. Mengamankan Harta/Kekayaan Perusahaan.

Harta/kekayaan disini meliputi kas perusahaan, persediaan barang dagangan, termasuk aset tetap perusahaan.

2. Menghasilkan Beragam Informasi untuk Pengambilan Keputusan.
3. Menghasilkan informasi untuk pihak Eksternal.
4. Menghasilkan informasi untuk penilainya kinerja karyawan atau divisi.
5. Menyediakan data masa lalu untuk kepentingan audit.

II.4. *Unified Modeling Language (UML)*

Adi Nugroho (2010:6), *Unified Modeling Language (UML)* adalah bahasa pemodelan untuk system atau perangkat lunak yang berparadigma “berorientasi objek” . Pemodelan (*Modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. Dalam hal ini sasaran model sesungguhnya adalah abstraksi segala sesuatu yang ada diplanet bumi menjadi gambaran-gambaran umum yang lebih mudah dipahami dan dipelajari. Adapun tujuan pemodelan (dalam rangka pengembangan system/perangkat lunak aplikasi) sebagai sarana analisis, pemasahaman visualisasi dan komunikasi antar anggota tim pengembang.

II.4.1. Pengenalan UML

Menurut Julius Hermawan (2010:7), UML (*Unified Modeling Language*) adalah bahasa standard yang digunakan untuk menjelaskan dan

memvisualisasikan artifak dan proses analisis dan desain berorientasi objek. UML Menyediakan standar pada notasi dan diagram yang bias digunakan untuk memodelkan suatu system. UML dikembangkan oleh tiga pendekar “berorientasi objek” yaitu Gradi Booch, Jim Rumbaugh dan Ivar Jacobson. UML menjadi bahasa yang bias digunakan untuk berkomunikasi dalam prespektif objek antara user dengan developer, antara developer analisis dengan developer desain dan antara developer desain dengan developer pemrograman.

UML memungkinkan developer melakukan pemodelan secara visual, yaitu penekanan pada penggambaran, bukan didominasi oleh narasi. Pemodelan visual membantu untuk menangkap struktur dan kelakuan dari si objek, mempermudah penggambaran interaksi antara elemen dalam system dan mempertahankan konsistensi antara desain dan implementasi dalam bahasa pemrograman.

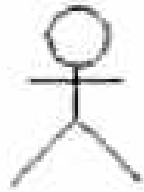
Namun karena UML hanya merupakan bahasa pemodelan maka UML bukanlah rujukan bagaimana melakukan analisis dan desain berorientasi objek. Untuk mengetahui bagaimana melakukan analisis dan desain berorientasi objek secara baik, sudah terdapat beberapa metodologi yang bias diikuti.

II.4.2. Notasi dan Artifak dalam UML

Menurut Julius Hermawan (2010:13), UML menyediakan beberapa notasi dan artifak standard yang bias digunakan sebagai alat komunikasi bagi para proses analisis dan desain. Artifak didalam UML didefenisikan sebagai informasi dalam berbagai bentuk yang digunakan atau dihasilkan dalam proses pengembangan perangkat lunak.

1. Aktor

Aktor adalah segala sesuatu yang berinteraksi dengan system aplikasi computer. Jadi actor ini bisa berupa orang, perangkat keras atau juga objek lain dalam system yang sama. Biasanya yang dilakukan oleh Aktor adalah memberikan informasi pada system dan\atau memerintahkan system untuk melakukan sesuatu.



Gambar II.1. Notasi Aktor

2. Class

Class merupakan pembentuk utama dari system berorientasi objek karena *class* menunjukkan kumpulan objek yang memiliki atribut dan operasi yang sama. *Class* digunakan untuk mengimplementasikan *interface*. *Class* digunakan untuk mengabstraksikan elemen-elemen dari system yang dibangun. *Class* bisa untuk direpresentasikan baik perangkat lunak maupun perangkat keras, baik konsep maupun benda nyata. Atribut digunakan untuk menyimpan informasi. Nama atribut menggunakan kata benda yang bisa dengan jelas direpresentasikan informasi yang disimpan didalamnya. Operasi menunjukkan sesuatu yang bisa dilakukan oleh objek, dan menggunakan kata kerja.

3. Interface

Interface merupakan kumpulan informasi tanpa implementasi dari suatu *class*. Implementasi operasi dari suatu *interface* dijabarkan oleh operasi didalam *class*. Oleh karena itu keberadaan *interface* selalu disertai oleh *class* yang

mengimplementasikan operasinya. *Interface* ini merupakan salah satu cara mewujudkan prinsip enkapsulasi dalam objek.



Gambar II.2. Notasi Interface

4. *Use Case*

Use case menjelaskan urutan kegiatan yang dilakukan Aktor dan system untuk mencapai tujuan tertentu. Walaupun menjelaskan kegiatan namun *use case* hanya menjelaskan apa yang dilakukan oleh actor dan system, bukan bagaimana system melakukan kegiatan tersebut.



Gambar II.3. Notasi Use Case

Didalam *use case* terdapat teks untuk menjelaskan urutan kegiatan yang disebut *use case specification*. *use case specification* terdiri dari :

a. Nama *Use Case*

Mencantumkan nama dari use case yang bersangkutan. Sebaiknya diawali dengan kata kerja untuk menunjukkan suatu aktivitas

b. Deskripsi singkat

Menjelaskan secara singkat dalam 1 atau 2 kalimat tentang tujuan dari use case ini.

c. Aliran Normal (*Basic Flow*)

Ini adalah jantung dari *use case*. Menjelaskan interaksi antara actor dan system dalam kondisi normal, yaitu segala sesuatu berjalan dengan lancar tiada halangan atau hambatan dalam mencapai tujuan dalam *use case*.

d. Aliran Alternatif (*Alternative Flow*)

Merupakan pelengkap dari *basic flow* tidak ada yang sempurna dalam setiap kali *use case* berlangsung. Didalam *Alternative Flow* ini dijelaskan dalam apa yang terjadi bila suatu halangan atau hambatan terjadi sewaktu *use case* berlangsung. Ini terutama berhubungan dengan *error* yang mungkin terjadi terutama karena system kekurangan data untuk diolah.

e. *Special Requirement*

Berisi kebutuhan lain yang belum tercakup dalam kebutuhan normal dan alternatif. Biasanya secara tegas dibedakan bahwa *basic flow* dan *alternate flow* menangani kebutuhan fungsional dari *use case* sementara *Special Requirement* yang tidak berhubungan dengan kebutuhan fungsional, misalnya kecepatan transaksi maksimum artinya berapa cepat dan berapa lama, kapasitas akses yaitu jumlah user yang akan mengakses dalam waktu bersamaan.

f. *Pre-Condition*

Menjelaskan persyaratan yang harus dipenuhi sebelum *use case* bisa dimulai.

g. *Post-Condition*

Menjelaskan kondisi yang berubah atau terjadi saat *use case* selesai dieksekusi.

5. *Interaction*

Digunakan untuk menunjukkan baik aliran pesan maupun informasi antara objek maupun antara hubungan objek. Biasanya *Interaction* dilengkapi juga dengan

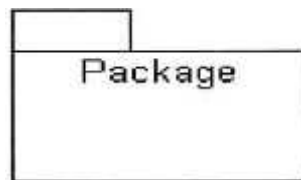
teks bernama *operation signature* yang tersusun dari mana operasi, parameter yang dikirim dan type parameter yang dikembalikan.



Gambar II.4. Notasi *Interaction*

6. *Package*

Package adalah kontainer atau wadah konseptual yang digunakan untuk mengelompokkan elemen-elemen dari system yang sedang dibangun, sehingga bisa dibuat model menjadi lebih sederhana. Tujuannya adalah untuk mempermudah pengelihatian dari model yang sedang dibangun.



Gambar II.5. Notasi *Package*

7. *Note*

Note digunakan untuk memberikan keterangan dan komentar tambahan dari suatu elemen sehingga bisa langsung terlampir dalam model. *Note* ini bisa ditempelkan ke semua elemen notasi yang lain.

8. *Dependency*

Dependency merupakan relasi yang menunjukkan bahwa perubahan pada salah satu elemen member pengaruh pada elemen lain.



Gambar II.6. Notasi *Dependency*

II.5. Pengertian Basis Data (*Database*)

Basis data merupakan kumpulan dari data-data yang saling terkait dan saling berhubungan satu dengan yang lainnya. Basis data adalah kumpulankumpulan *file* yang saling berkaitan :

Menurut Kusri (2010, p2), pengertian Basis Data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai obyek, orang, dan lain-lain. Data dinyatakan dengan nilai (angka, deretan karakter, atau symbol).

Basis data dapat didefinisikan dalam berbagai susut pandang seperti berikut:

1. Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa sehingga kelak dapat dimanfaatkan dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (*redundancy*) yang tidak perlu, untuk memenuhi kebutuhan.
3. Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpan elektronik.

II.6. Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data rasional. Pada dasarnya, Normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk atribut dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari table relasional. Teori normalisasi didasarkan pada konsep

bentuk normal. Sebuah tabel relasional dikatakan berada pada bentuk normal tertentu jika tabel memenuhi himpunan batasan tertentu. Ada lima bentuk normal yang telah ditemukan.

II.6.1. Bentuk-bentuk Normalisasi

1. Bentuk normal pertama (1NF)

Contoh yang kita gunakan di sini adalah sebuah perusahaan yang mendapatkan barang dari sejumlah pemasok. Masing-masing pemasok berada pada satu kota. Sebuah kota dapat mempunyai lebih dari satu pemasok dan masing-masing kota mempunyai kode status tersendiri.

2. Bentuk normal kedua (2NF)

Definisi bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1NF, tetapi tidak pada 2NF. Sebuah tabel relasional berada pada bentuk normal kedua jika dia berada pada bentuk normal kedua jika dia berada pada 1NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada seluruh kolom yang membentuk kunci utama.

3. Bentuk normal ketiga (3NF)

Bentuk normal ketiga mengharuskan semua kolom pada tabel relasional tergantung hanya pada kunci utama. Secara definisi, sebuah tabel berada pada bentuk normal ketiga (3NF) jika tabel sudah berada pada 2NF dan setiap kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya.

4. Boyce Code Normal Form (BCNF)

Setelah 3NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Banyak praktisi

berpendapat bahwa menempatkan entitas pada 3NF sudah cukup karena sangat jarang entitas yang berada pada 3NF bukan merupakan 4NF dan 5NF.

5. Bentuk Normal Tahap Keempat dan Kelima

Sebuah tabel relasional berada pada bentuk normal keempat (4NF) jika dia dalam BCNF dan semua ketergantungan multivalued merupakan ketergantungan fungsional. Bentuk normal keempat (4NF) didasarkan pada konsep ketergantungan multivalued (MVD).

Sebuah tabel berada pada bentuk normal kelima (5NF) jika ia tidak dapat mempunyai dekomposisi lossless menjadi sejumlah tabel lebih kecil. Empat bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima berdasarkan pada konsep ketergantungan gabungan (*join dependence*) (Janner Simarmata ; 2010 : 76).

II.7. Visual Basic 2010

Menurut Edi Winarno ST, M.Eng dkk (2010:1), Visual Basic adalah bahasa pemrograman klasik, legendaris yang paling banyak dipakai oleh programmer di dunia. Pemrograman ini dipakai oleh jutaan programmer dan tercatat sebagai program yang paling disukai oleh mayoritas orang.

Visual Studio 2010 pada dasarnya adalah sebuah bahasa pemrograman komputer. Dimana pengertian dari bahasa pemrograman itu adalah perintah-perintah atau instruksi yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. Visual Studio 2010 selain disebut dengan bahasa pemrograman, juga sering disebut sebagai sarana (tool) untuk menghasilkan program-program aplikasi berbasis windows.

Beberapa kemampuan atau manfaat dari Visual Studio 2010 diantaranya seperti :

1. Untuk membuat program aplikasi berbasis windows.
2. Untuk membuat objek-objek pembantu program seperti, misalnya : kontrol ActiveX, file Help, aplikasi Internet dan sebagainya.
3. Menguji program (debugging) dan menghasilkan program berakhiran EXE yang bersifat executable atau dapat langsung dijalankan.

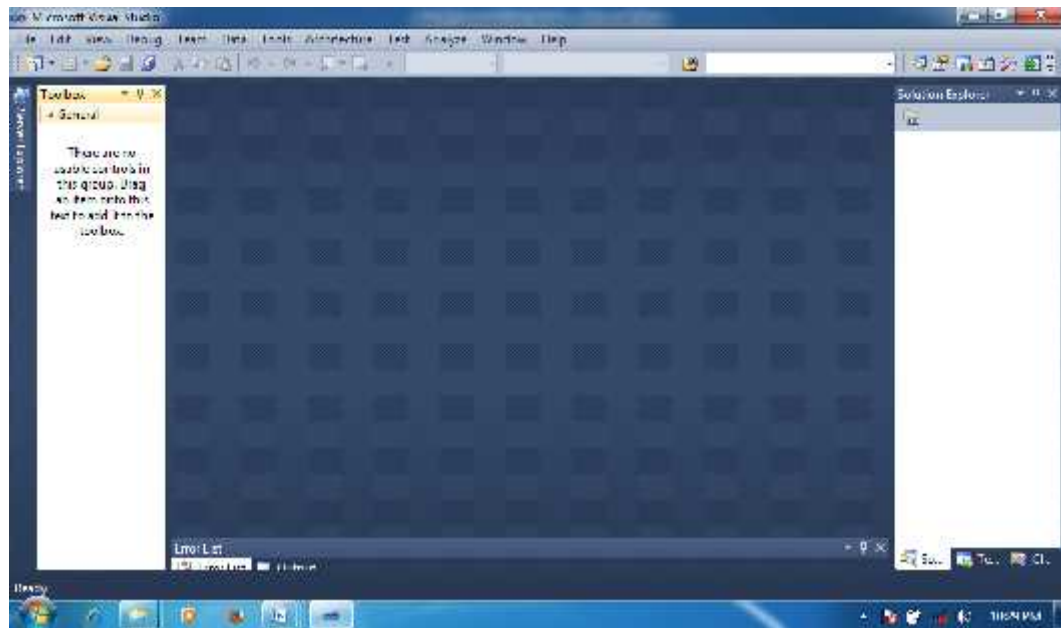
II.7.1. Antar Muka Visual Basic 2010

Saat menjalankan Visual Basic 2010 pertama kali muncul jendela *chosedefault environment settings*. Disini bisa memilih apakah ingin memilih antar muka di Visual Studio. Untuk *programmer* Visual Basic lebih baik memilih *Visual Basic Development Centre*. Dibagian awal visual basic, bisa memilih *Start Page*. *Start Page* adalah halaman yang mencantumkan informasi-informasi seputar program dan juga informasi RSS dari sumber tertentu. Jika tidak ingin menampilkan hal ini hilangkan tanda centang pada *Show Page On Startup*.



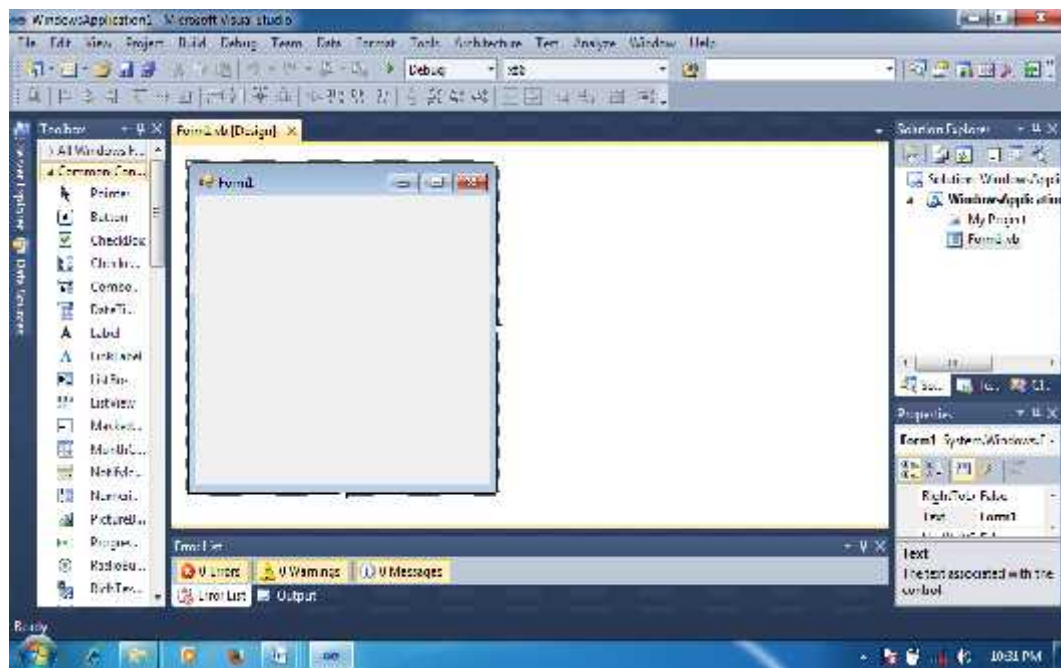
Gambar II.7. Start Page Visual Basic 2010

Jika *start page* ditutup terlihat tampilan sebagai berikut :



Gambar II.8. Tampilan IDE (*Integrated Development Environment*) setelah *Start Page* ditutup

Jika ada sebuah form yang terlihat, tampilan lengkap IDE seperti gambar berikut ini.



Gambar II.9. Tampilan lengkap IDE

Komponen-komponen dari IDE adalah :

1. Dibagian kiri terdapat toolbox yang menampilkan semua objek tool yang bisa dimasukkan kedalam form untuk membuat program.
2. Dibagian tengah terdapat tempat meletakkan form dan kode, baik disaat desain ataupun pada saat program dijalankan.
3. Dibagian kanan terdapat solution explorer yang merupakan explorer untuk melihat file-file disebuah objek.
4. Dikanan bawah terdapat propertis untuk melihat properti dari nilai-nilai pada objek yang dipilih dibagian tengah. (Edi Winarno ST, M.Eng dkk, 2010:1)

II.8. SQL Server 2008 *Express Edition*

Menurut Wahana Komputer (2010:2), SQL Server 2008 *Express Edition* sebuah terobosan baru dalam bidang database, SQL Server adalah sebuah DBMS (*Database Management System*) yang dibuat oleh Microsoft untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan Oracle. SQL Server 2008 *Express Edition* dibuat pada saat kemajuan dalam bidang hardware semakin pesat. Oleh karena itu sudah dapat dipastikan bahwa SQL Server 2008 *Express Edition* membawa terobosan dalam bidang pengolahan dan penyimpanan data.

II.8.1 Kebutuhan *Hardware*

Adapun *hardware* yang diperlukan untuk instalasi SQL Server 2008 *Express Edition* minimal adalah sebagai berikut :

1. Prosescor minimal 1 GHz
2. Memori minimal 512 MB
3. Sistem Operasi Windows

Biar dapat diinstal pada system computer dengan memoti 512 MB, tetapi disarankan menggunakan memori 1 GB. Sedangkan untuk jaringannya diperlukan adalah :

1. *Sharer Memory*
2. TCP/IP
3. *Named Pipes*
4. *Virtual Interface Adapter (VIA)*(Wahana Komputer, 2010:2).

II.8.2 Versi SQL Server 2008 *Express Edition*

Microsoft merilis SQL Server 2008 *Express Edition* dalam beberapa versi yang disesuaikan dengan segmen-segmen pasar yang dituju. Versi-versi tersebut adalag sebagai berikut :

1. Menurut cara pemrosesan data pada prosesor makan Microsoft mengelompokkan produk ini berdasarkan dua jenis yaitu :
 - a. Versi 32 Bit (x86), yang biasanya digunakan untuk komputer *single processor* (Pentium 4) atau lebih tepatnya processor 32 bit atau Windows XP
 - b. Versi 64 Bit (x64), yang biasanya digunakan oleh computer yang lebih sari satu processor (Misalnya *Core 2 duo*) dan system operasi 64 bit, Vista dan Windows 7.

2. Sedangkan secara keseluruhan terdapat versi-versi seperti berikut :
 - a. Versi *Compact* ini adalah versi “tipis” dari semua versi yang ada
 - b. Versi *Express* ini adalah versi “ringan”

II.8.3 Instalasi SQL Server 2008 *Express Edition*

Proses instalasi SQL Server 2008 *Express Edition* tidak sama dengan instalasi versi-versi sebelumnya. Proses SQL Server 2008 *Express Edition* agak panjang melalui beberapa tahapan. Tahapan yang dilakukan akan membawa beberapa pilihan yang akan diisi dalam *setting* sebuah *server database*. Berikut ini adalah pilihan-pilihan yang akan dijumpai dalam proses instalasi SQL Server 2008 *Express Edition*.

1. Tempat direktori utama dan penyimpanan file database

Direktori utama adalah direktori dimana semua file program akan ditempatkan dan file-file tersebut tidak akan berubah selama anda menjalankan SQL server. Direktori utama secara standard akan berada dalam direktori “C:\Program Files\Microsoft SQL Server”.

2. Penggunaan *Multiple instance*

Instance adalah sebuah turunan dari server database SQL Server. Karena sebuah tiruan maka sebuah *Instance* memiliki fungsi yang sama dengan database server aslinya. Arti sebenarnya *Instance SQL Server* adalah sebuah server database yang tidak men-*sharing* sistemnya dan database user dengan database server lainnya yang ada dalam komputer yang sama.

3. Jasa *Autentification User* (Menggunakan Windows atau mixed)

Autentification User diperlukan supaya server tidak dapat dipergunakan oleh orang yang tidak bertanggungjawab dan tidak berhak. Dalam SQL server ada dua *Autentification User* yang dapat digunakan yaitu :

- a. Mode Windows, Pada mode ini SQL Server akan melakukan autentifikasi dengan menggunakan level login pada system operasi.
- b. Mode Mixel atau campuran, mode ini menginjinkan *user* untuk masuk kedalam system SQL server dengan menggunakan *Account* yang dibuat di system operasi windows atau juga menggunakan *account* yang di *set up* pada SQL Server (Wahana Komputer, 2010:2)