

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).

Menurut Henny Hendarti (2011 : 1), menyatakan bahwa sistem adalah sekelompok komponen yang saling berhubungan, bekerja sama untuk mencapai tujuan bersama dengan menerima input serta menghasilkan output dalam proses tranpormasi yang teratur.

Menurut Tata Sutabri (2005 : 8), menyatakan bahwa sistem adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu.

Dari beberapa pengertian diatas diambil kesimpulan bahwa sistem adalah kumpulan atau kelompok yang saling terkait untuk mencapai tujuan.

II.1.1. Klasifikasi Sistem

Menurut Julkifli Amsyah MLS (2005: 25), Sistem dapat diklasifikasikan dari beberapa sudut pandangan, diantaranya sebagai berikut :

1. Sistem abstrak dan sistem fisik.

Sistem abstrak adalah sistem yang berisi gagasan atau konsep. Misalnya, sistem teologi yang berisi gagasan tentang hubungan manusia dan Tuhan.

Sistem fisik merupakan sistem yang secara fisik dapat dilihat. Misalnya sistem komputer, sistem sekolah, sistem akuntansi, dan sistem transportasi.

2. Sistem alamiah dan sistem buatan manusia.

Sistem alamiah adalah sistem yang terjadi karena alam (tidak dibuat manusia).

Misalnya, sistem tata surya. Sistem buatan manusia adalah sistem yang dibuat oleh manusia. Misalnya, sistem komputer dan sistem mobil.

3. Sistem tertentu dan sistem tak tentu.

Sistem tertentu adalah sistem yang operasinya dapat diprediksi secara tepat.

Misalnya, sistem komputer. Sistem tak tentu adalah sistem yang tak dapat diramal dengan pasti karena mengandung unsur probabilitas. Misalnya, sistem arisan dan sistem sediaan.

4. Sistem tertutup dan sistem terbuka

Sistem tertutup adalah sistem yang tidak bertukar materi, informasi, atau energi dengan lingkungan. Misalnya, reaksi kimia dalam tabung terisolasi.

Sistem terbuka adalah sistem yang berhubungan dengan lingkungan dan dipengaruhi oleh lingkungan.

II.1.2. Karakteristik Sistem

Menurut Hanif Al Fatha (2007:5), Suatu sistem mempunyai karakteristik tertentu yaitu

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 berupa 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 bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut

4. Penghubung sistem (*Interface*)

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui perhubungan ini memungkinkan sumber-sumber daya mengalir dari subsistem yang lainnya.

5. Masukan sistem (*Input*)

Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*).

6. Keluaran sistem (*Output*)

Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

7. Pengolah sistem (*Process*)

Suatu sistem yang dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran.

8. Sasaran sistem (*Objective*)

Suatu sistem pasti mempunyai tujuan atau sasaran. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

II.1.3. Elemen sistem

Menurut Hanif Al Fatha (2007:15), Ada beberapa elemen yang membentuk suatu sistem, yaitu:

1. Tujuan, merupakan tujuan dari sistem tersebut yang dapat berupa tujuan usaha, kebutuhan, masalah dan prosedur pencapaian tujuan.
2. Batasan, merupakan batasan-batasan yang ada dalam mencapai tujuan dari sistem, dimana batasan ini berupa peraturan-peraturan, biaya-biaya, dan peralatan.
3. Kontrol, merupakan pengawas dari pencarian tujuan sistem, yang dapat berupa kontrol pemasukan data (*input*), kontrol pengeluaran data (*output*) dan kontrol pengoperasian.

4. Input, merupakan bagian dari sistem yang bertugas untuk menerima data masukan, dimana data dapat berupa masukan, frekuensi pemasukan data, dan jenis pemasukan data.
5. Proses, merupakan bagian yang memproses masukan data menjadi informasi sesuai dengan keinginan penerima, dimana proses dapat klasifikasi, peringkasan dan pencarian.
6. Output, merupakan keluaran atau tujuan akhir dari sistem, output dapat berupa laporan, dan grafik.
7. Umpan balik, merupakan elemen sistem yang mempunyai tugas untuk melihat kembali apakah sistem telah berjalan sesuai dengan yang diinginkan, umpan balik dapat berupa perbaikan, pemeliharaan.

II.2. Teori Data

Data berasal dari kata “datum” yang berarti fakta, yang mengandung arti dikembangkan dengan kenyataan yang dapat digambarkan dengan simbol, angka, huruf, dan sebagainya.

Menurut Aji Supriyanto (2005 :6), Menyatakan data merupakan objek yang belum dan akan dilakukan pengolahan yang sifatnya masih “mentah”.

Menurut Abdul Kadir (2009:29), menyatakan data adalah deskripsi tentang benda, kejadian, aktivitas, dan transaksi, yang tidak mempunyai makna atau tidak berpengaruh secara langsung kepada pemakai.

Menurut Tata Sutabri (2005:16), menyatakan data merupakan kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata.

Menurut Kenneth C. Laudon (2008:10), menyatakan data adalah sekumpulan baris fakta yang mewakili peristiwa yang terjadi pada organisasi atau pada lingkungan fisik sebelum diolah kedalam format yang bisa dimengerti dan digunakan orang.

II.3. Pengertian Informasi

Informasi adalah hasil dari pengolahan data dalam bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata sehingga dapat digunakan sebagai pengambilan keputusan.

Menurut Kenneth C. Laudon (2008:10), menyatakan bahwa informasi adalah data yang telah terolah dan sifatnya menjadi data yang lain yang bermanfaat.

Menurut Tata Sutabri (2005:23), menyatakan informasi adalah data yang telah diklasifikasi atau diolah atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan.

Menurut Kenneth C. Laudon, (2005:14) menyatakan informasi adalah data yang telah dibentuk kedalam suatu format yang mempunyai arti dan berguna bagi manusia.

Dari pengertian diatas diambil kesimpulan bahwa informasi adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata yang digunakan dalam mengambil keputusan.

II.3.1. Kualitas Informasi

Menurut Kusrini dan Andri kuniyo (2007:9), Kualitas suatu informasi tergantung dari 3 (tiga) hal yaitu, informasi harus akurat (*accurate*), tepat waktu (*timelines*), dan relevan (*relevance*). Untuk penjelasannya akan dipaparkan dibawah ini:

1. Akurat (*accurate*)

Berarti informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan serta harus jelas mencerminkan maksudnya, informasi harus akurat, karena dari sumber informasi sampai kepada si penerima informasi kemungkinan banyak terjadi gangguan yang dapat mengubah informasi tersebut.

2. Tepat waktu (*timelines*)

Berarti informasi yang datang pada sipenerima tidak boleh terlambat. Informasi sudah usang tidak mempunyai nilai lagi, sebab informasi mempunyai landasan di dalam pengambilan keputusan.

3. Relevan (*relevance*)

Berarti informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk orang satu dengan yang lainya berbeda.

II.3.2. Nilai informasi

Menurut Julkifli Amsyah MLS (2005: 314), Nilai dari informasi (*value of information*) ditentukan dari dua hal, manfaat dan biaya mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan

biaya mendapatkannya. Akan tetapi perlu diperhartikan informasi yang dipergunakan didalam suatu sistem informasi umumnya digunakan untuk menghubungkan suatu bagian informasi pada suatu masalah dengan biaya untuk memperolehnya karena sebagian besar informasi dinikmati tidak hanya oleh pihak didalam perusahaan.

Sebagian besar informasi tidak dapat persis ditafsir keuntungannya dengan suatu nilai uang, tetapi dapat ditafsir nilai efektifitasnya. Pengukuran nilai informasi biasanya dihubungkan dengan analisis *cist effectiveness* atau *cost benefit*. Nilai informasi ini didasarkan atas 10 (sepuluh) sifat yaitu :

1. Mudah diperoleh

Sifat ini menunjukkan mudah dan cepatnya informasi dapat diperoleh. Kecepatan untuk memperoleh dapat diukur.

2. Luas dan lengkap

Sifat ini menunjukkan lengkapnya isi informasi. Hal ini tidak berarti hanya mengenai volumenya, tetapi juga keluaran informasinya. Sifat ini sangat kabur dan karena itu sulit untuk mengukurnya.

3. Ketelitian

Sifat ini berhubungan dengan tingkat keberadaan dari kesalahan keluaran informasi dalam hubungannya dengan volume data yang besar biasanya terjadi dua jenis kesalahan pencatatan dan kesalahan perhitungan.

4. Kecocokan

Sifat ini menunjukkan betapa baik keluaran informasi dalam hubungannya dengan penerima para pemakai. Isi informasi harus ada hubungannya dengan masalah yang sedang dihadapi.

5. Ketepatan waktu

Sifat ini berhubungan dengan waktu yang dilalui yang lebih pendek, daripada siklus untuk mendapatkan informasi. Masukan, pengolahan dan pelaporan keluaran kepada para pemakai biasanya tepat waktu.

6. Kejelasan

Sifat ini menunjukkan tingkat keluaran yang bebas dari istilah-istilah yang tidak jelas karena membetulkan laporan dapat membutuhkan biaya yang besar.

7. Keluwesan

Sifat ini berhubungan dengan dapat disesuaikan keluaran sistem tidak hanya dengan lebih dari satu keputusan, tetapi juga dengan lebih dari seorang pengambil keputusan.

8. Dapat dibuktikan

Sifat ini menunjukkan kemampuan beberapa pemakai informasi untuk menguji keluaran informasi dan sampai pada kesimpulan yang sama.

9. Tidak ada prasangka

Sifat ini berhubungan dengan tidak adanya keinginan untuk mengubah informasi guna mendapatkan kesimpulan yang telah dipertimbangkan sebelumnya.

10. Dapat diukur

Sifat ini menunjukkan hakekat informasi yang dihasilkan dari sistem informasi formal.

Nilai informasi yang sempurna adalah pengambilan keputusan diijinkan untuk memilih keputusan optimal dalam setiap hal dan bukan keputusan yang “rata-rata” akan menjadi optimal dan untuk menghindari kejadian-kejadian yang akan menyebabkan kerugian.

II.4. Pengertian Sistem Informasi

Sistem informasi dapat didefinisikan sebagai suatu sistem dalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi yang penting dengan tujuan sebagai pengambilan keputusan.

Menurut Kenneth C. Laudon (2005:23), menyatakan sistem informasi sebagai satuan komponen yang saling berhubungan yang mengumpulkan (mendapatkan kembali), memproses, menyimpan, mendistribusikan informasi untuk mendukung pengambilan keputusan dan kendali dalam suatu organisasi.

Menurut Kusriani dan Andri kuniyo (2007:8), menyatakan sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategis dari suatu organisasi untuk dapat menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

II.4.1. Komponen Sistem Informasi

Menurut Kusrini dan Andri kuniyo (2007:9), Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*Building Block*) yaitu Blok Masukan (*Input Block*), Blok Model (*Model Block*), Blok Keluaran (*Output Block*), Blok Teknologi (*Technologi Block*), Blok Dasar Data (*Database Block*) dan Blok Kendali (*Control Block*)

Sebagai sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai suatu sasaran.

1. Blok Masukan (*Input Block*)

Input mewakili data yang masuk kedalam sistem informasi. Input disini termasuk metode-metode dan media untuk menangkap data yang dimasukkan, yang dapat berupa dokumen-dokumen.

2. Blok model (*Model Block*)

Blok ini terdiri dari kombinasi prosedur, logika dan model matematika yang akan memanipulasi data input dan data tersimpan di dasar data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran (*Output Block*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi (*Tecnologi Block*)

Teknologi merupakan “Kotak Tool” dari pekerjaan sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model menyimpan dan mengakses data, menghubungkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan. Teknologi terdiri dari bagian utama yaitu:

- a. Teknisi (*Brainware*)
- b. Perangkat Lunak (*Software*)
- c. Perangkat keras (*Hardware*)

Teknisi dapat berupa orang-orang yang mengetahui teknologi dan membantunya untuk dapat beroperasi.

5. Blok Basis Data (*Database Block*)

Basis data merupakan komponen dari data yang saling terhubung satu dengan yang lainnya, tersimpan di perangkat lunak untuk memanipulasinya. Data di dalam basis data perlu diorganisasikan sedemikian rupa supaya informasi yang dihasilkan berkualitas.

6. Blok Kendali (*Control Block*)

Agar sistem dapat berjalan sesuai dengan yang diinginkan maka perlu diterapkan pengendalian-pengendalian di dalamnya. Banyak hal yang dapat merusak sistem informasi seperti misalnya bencana alam, api, temperatur, air, debu, kecurangan sistem itu sendiri, dan lain-lain.

II.4.2. Jenis Sistem Informasi

Menurut Eko Nugroho (2008:135), Sistem informasi memiliki beberapa jenis yaitu:

1. Sistem informasi menurut level organisasi

Berdasarkan level organisasi, sistem informasi yang dikelompokkan menjadi

- a. Sistem informasi Departemen
- b. Sistem informasi perusahaan
- c. Sistem informasi antarorganisasi.

2. Sistem informasi fungsional

Sistem informasi fungsional adalah sistem informasi yang ditujukan untuk memberikan informasi bagi kelompok orang yang berada pada bagian tertentu dalam perusahaan. Beberapa informasi fungsional yang umum yaitu:

- a. Sistem informasi akuntansi
- b. Sistem informasi keuangan,
- c. Sistem informasi manufaktur
- d. Sistem informasi pemasaran,
- e. Sistem informasi SDM.

3. Sistem informasi berdasarkan dukungan yang tersedia

Berdasarkan dukungan yang diberikan pada pemakai, sistem informasi yang digunakan semua area fungsional dapat diklasifikasikan yaitu sistem pemrosesan transaksi, sistem informasi manajemen, sistem informasi perkantoran, sistem pendukung keputusan, sistem informasi eksekutif, sistem pendukung kelompok, dan sistem pendukung cerdas.

4. Sistem Informasi Geografis

Sistem informasi geografis adalah sistem berbasis komputer yang digunakan untuk menyimpan dan manipulasi informasi geografis. Sistem informasi geografis dapat menggabungkan tugas-tugas pengambilan keputusan seperti mencari jarak terdekat atau tercepat dari posisi A ke posisi B, menentukan/memilih lokasi lain memiliki pola serupa, mengelompokkan daerah perindustrian untuk meminimalkan jarak perjalanan.

II.4.3. Tujuan Sistem Informasi

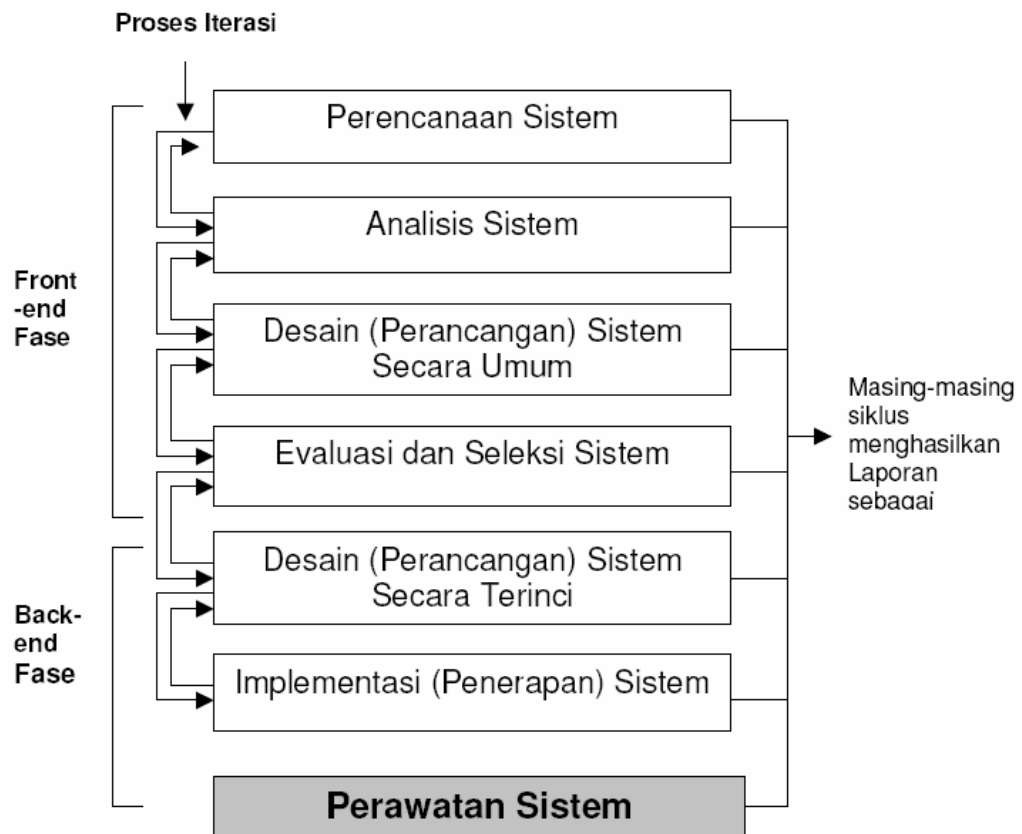
Adapun tujuan sistem informasi adalah untuk menghasilkan informasi yang baik dan lebih berguna untuk mempermudah dan mengefisienkan waktu, biaya dalam mengambil suatu keputusan dan untuk mengurangi ketidakpastian bagi pengambil keputusan. Adapun tujuan utama yang umum dari sistem informasi adalah:

1. Untuk mendukung fungsi kepengurusan manajemen
2. Untuk mendukung pengambilan keputusan manajemen
3. Untuk mendukung kegiatan operasi perusahaan.

II.4.4. Metodologi Pengembangan Sistem

Jogiyanto, H.M. (2008 : 59), Pengembangan sistem dapat berupa menyusun suatu sistem yang baru dan menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang telah ada. Setiap tahap harus diselesaikan terlebih dahulu kemudian diteruskan ketahap berikutnya untuk

menghindari terjadinya pengulangan tahap. Metodologi pengembangan sistem Waterfall dapat dilihat di bawah ini :



Gambar II.1 Struktur Hidup Pengembangan Sistem

*Sumber : “Analisis dan Desain Sistem Informasi: Pendekatan Terstruktur
Teori dan Praktek Aplikasi Bisnis (jogiyanto ; 2008 : 59)”*

Dari gambar diatas dapat dijelaskan sebagai berikut :

1. Perencanaan sistem

Manfaat dari tahapan ini adalah untuk menentukan masalah-masalah atau kebutuhan yang timbul. Hal ini memerlukan pengembangan sistem secara menyeluruh agar ada usaha lain yang dapat di lakukan untuk memecahkan masalah tersebut.

2. Analisa Sistem.

Tahap analisa bertitik tolak pada kegiatan-kegiatan dan tugas-tugas dimana sistem yang berjalan di pelajari lebih mendalam, konsepsi dan usulan dibuat untuk menjadi landasan bagi sistem yang baru yang akan dibangun.

3. Perancangan Sistem.

Pada tahap ini sebagian besar kegiatan yang berorientasi ke komputer dilaksanakan. Spesifikasi perangkat keras dan perangkat lunak yang telah disusun pada tahap sebelumnya ditinjau kembali dan disempurnakan. Rencana pembuatan program dilaksanakan dan juga testing programnya.

4. Implementasi Sistem

Tahap ini prosedur yang dilakukan untuk menyelesaikan desain sistem yang ada dalam dokumen desain sistem yang disetujui dan menguji, menginstal dan memulai penggunaan sistem baru atau sistem yang diperbaiki.

5. Pemeliharaan Sistem

Tujuan tahapan ini adalah untuk melakukan evaluasi sistem secara tepat dan efisien, menyempurnakan proses pemeliharaan sistem dengan selalu menganalisa kebutuhan informasi yang dihasilkan sistem tersebut.

II.5. Teori Akuntansi

Menurut Al. Haryono Jusuf (2005 : 21), Akuntansi adalah suatu proses mengidentifikasi, mengukur dan menyampaikan informasi ekonomi sebagai informasi dalam hal mempertimbangkan sebagai alternatif dalam mengambil kesimpulan oleh para pemakainya.

Sistem informasi akuntansi merupakan sebuah sistem informasi yang mengubah data transaksi bisnis menjadi informasi keuangan yang berguna bagi pemakainya.

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.

Fungsi utama akuntansi adalah menyediakan laporan – laporan periodik untuk manajemen, investor, kreditur, dan pihak- pihak lain diluar perusahaan. Laporan keuangan utama yang dihasilkan dari proses akuntansi adalah neraca dan laporan laba rugi. Neraca dibuat dengan maksud untuk menggambarkan posisi keuangan suatu organisasi, sedangkan laba rugi menggambarkan hasil – hasil usaha yang dicapai dalam suatu periode tertentu.

Sisitem informasi akuntansi adalah sumber daya manusia dan modal dalam organisasa yang bertanggung jawab untuk persiapan informasi keungan dan perusahaan.

II.6. UML (*unified modelling language*)

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 mendefenisikan 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.

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

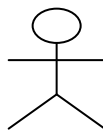
Tabel II.1 Tipe Diagram UML

Sumber : " Pemodelan Visual dengan UML (Munawar ; 2009 : 32)"

II.6.1. Notasi Dasar UML

1. Actor

Sebuah actor mencirikan suatu bagian *outside user* atau susunan yang berkaitan dengan user yang berinteraksi dengan sistem [Rumbaugh, Booch, dan Jacobson 1999]. Dalam model use case, aktor merupakan satu-satunya kesatuan eksternal yang berinteraksi dengan sistem.

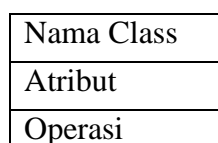


Gambar II.2 : Notasi Actor pada UML

Sumber : "Pemodelan Visual dengan UML (Munawar ; 2009 : 41)"

2. Class

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 yang berorientasi objek. Berikut notasi class dalam UML:

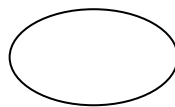


Gambar II.3 : Notasi Class pada UML

Sumber : " Pemodelan Visual dengan UML (Munawar ; 2009 : 41)"

3. Use Case

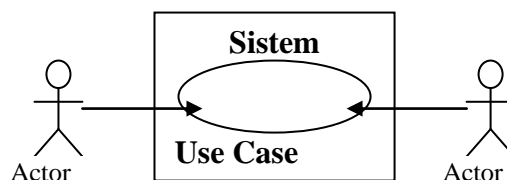
Use Case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use Case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah – langkah yang menerangkan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



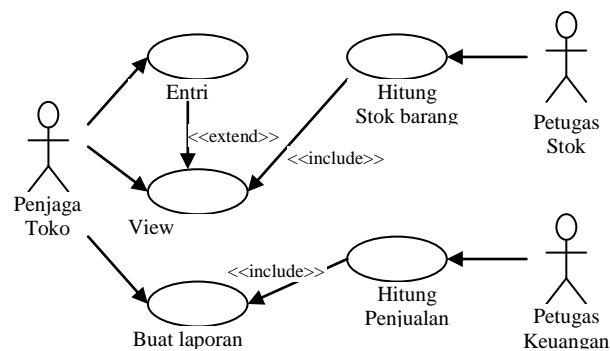
Gambar II.4 : Notasi *Use Case* pada UML
Sumber : "Pemodelan Visual dengan UML (Munawar ; 2009 : 42)"

4. Use Case Diagram

Use Case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* mempresentasikan sebuah interaksi antara actor dengan sistem. *Use Case* menggambarkan kata kerja seperti *Login* ke sistem, *maintenance user* dan sebagainya. Model *use case* seperti gambar 4 dan contoh *use case* diagram ditunjukkan pada Gambar II.5.



Gambar II.5 : Model *Use Case* pada UML
Sumber : "Pemodelan visual dengan UML (Munawar ; 2009 : 43)"



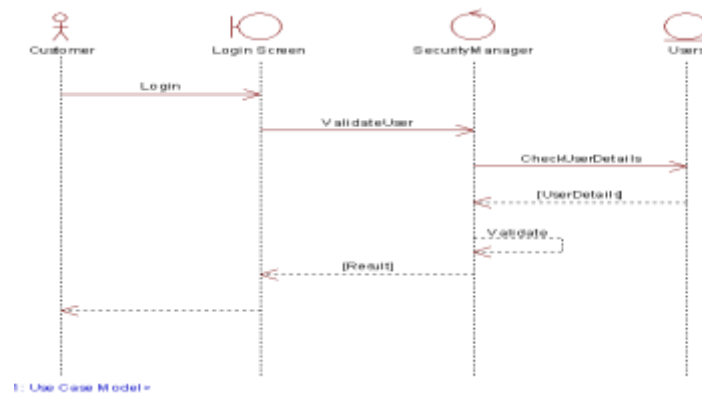
Gambar II.6 : Use Case diagram pada penjualan CD
 Sumber : " *Pemodelan visual dengan UML (Munawar ; 2009 : 43)* "

5. Sequence Diagram

Sequence diagram menjelaskan secara detail urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case*: interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi. Pembuatan *sequence diagram* merupakan aktivitas yang paling kritical dari proses disain karena artifak inilah yang menjadi pedoman dalam proses pemrograman nantinya dan berisi aliran kontrol dari program.

Sequence diagram biasanya tersusun dari elemen objek, *Interaction* dan *Message*. *Interaction* menghubungkan 2 Obyek dengan pesannya. Diagram ini menjelaskan aspek dinamis dari sistem yang sedang dibangun. Di dalam *sequence diagram*, terdapat kelas *boundary*, *control* dan *entity*.

Contoh *sequence diagram* :



Gambar II.7 : Sequence Diagram

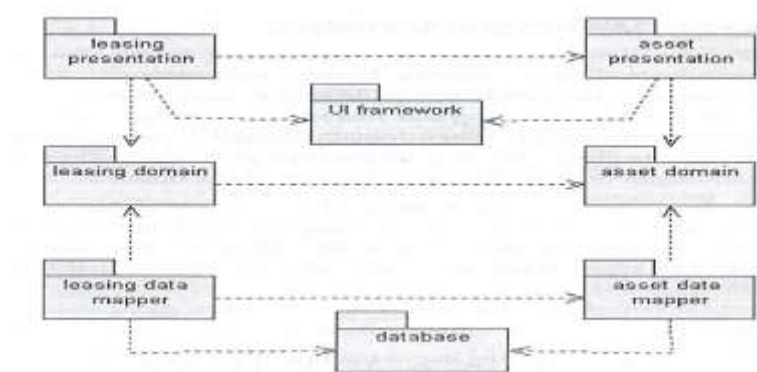
Sumber : " *Pemodelan visual dengan UML (Munawar ; 2009 : 44)* "

6. Class Diagram

Sama seperti *class*, maka class diagram merupakan diagram yang selalu ada di pemodelan sistem berorientasi objek. *Class diagram* menunjukkan hubungan antar *class* yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan.

Class diagram umumnya tersusun dari elemen *class*, *interface*, *dependency*, *Generalization* dan *Association*. *Relasi dependency* menunjukkan bagaimana terjadi ketergantungan antar *class* yang ada. *Relasi Generalization* menunjukkan bagaimana suatu *class* menjadi *superclass* dari *class* lainnya dan *class* tersebut menjadi *subclass* dari *class* tersebut. *Relasi Association* menggambarkan navigasi antar *class*, berapa banyak objek lain bisa berhubungan dengan satu objek (*multiplicity* antar *class*), dan apakah satu *class* menjadi bagian dari *class* lainnya (*agregation*). *Class diagram* digunakan untuk menggambarkan disain statis dari sistem yang sedang dibangun.

Contoh *class diagram* :



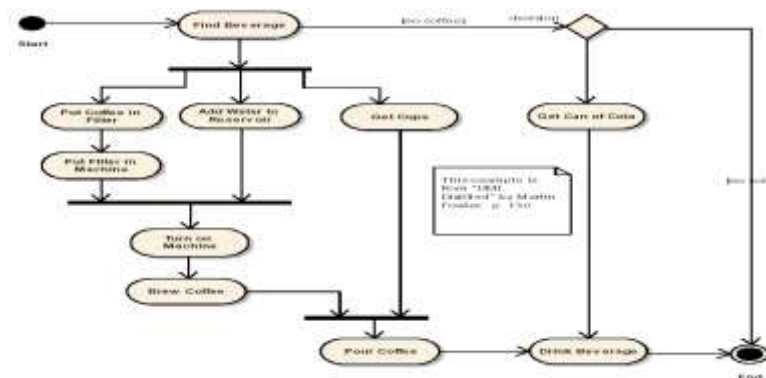
Gambar II.8 : Class Diagram

Sumber : " *Pemodelan visual dengan UML (Munawar ; 2009 : 45)* "

7. Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour internal* sebuah sistem dan interaksi antar subsistem secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan

aktivitas. Sama seperti *state*, standart UML menggunakan segi empat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan *behaviour* pada kondisi tertentu, digambarkan dengan simbol belah ketupat. Untuk mengilustrasikan proses-proses paralel (*fork and join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu. Adapun contoh dari Activity Diagram dapat di lihat pada Gambar II.8.



Gambar II.9 : Activity Diagram

Sumber : ” Pemodelan visual dengan UML (Munawar ; 2009 : 47)”

II.7. Pengertian Basis Data (*Database*)

Basis data (*database*) adalah suatu data yang terintegrasi, diorganisasikan, dan disimpan dalam suatu cara yang memudahkan untuk pengambilan kembali.

Menurut Eko Nugroho (2008:27), basis data adalah suatu susunan kumpulan data operasional lengkap dari suatu organisasi /perusahaan yang

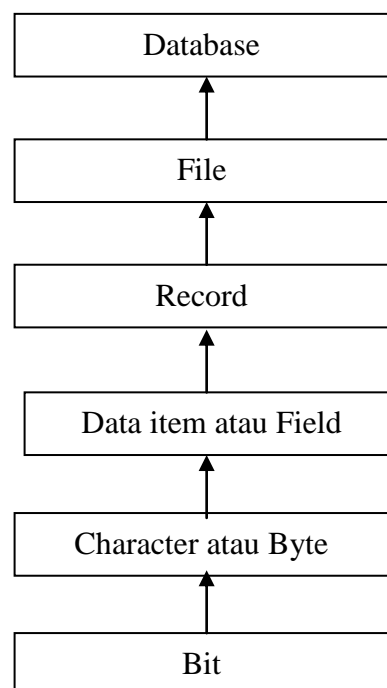
terorganisir/dikelola dan disimpan secara terintegrasi dengan menggunakan metode tertentu menggunakan komputer sehingga mampu menyediakan informasi optimal yang diperlukan pemakainya

Menurut Tata Sutabri (2005:161), menyatakan *database* adalah suatu koleksi data komputer yang terintegrasi, diorganisasikan, dan disimpan dalam suatu cara yang harus digunakan.

Abdul kadir (2005:254), menyatakan *database* adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi.

II.7.1. Struktur Data

Database merupakan salah satu komponen yang penting dalam sistem informasi, karena merupakan basis (dasar) dalam menyediakan informasi bagi para pemakai. Dimana dalam membentuk *database* yang terstruktur harus memperhatikan jenjang-jenjang dari data yaitu :



Gambar II.10 Hierarki data

1. *Bit*

Bit merupakan bagian data yang kecil, dapat berupa karakter numerik, huruf maupun karakter-karakter khusus yang membentuk suatu item data, dimana kumpulan karakter membentuk satu *field*.

2. *Byte*

Byte adalah sistem biner yang terdiri atas dua macam nilai, yaitu 0 dan 1. sistem *biner* merupakan dasar yang dapat digunakan untuk komunikasi antara manusia dan mesin, yang merupakan serangkaian komponen elektronik dan hanya dapat membedakan 2 macam keadaan, yaitu ada tegangan dan tidak ada tegangan yang masuk ke rangkaian tersebut.

3. *Field*

Suatu *field* menggambarkan suatu atribut *record* yang menunjukkan suatu item dari data seperti nama, alamat, dimana kumpulan *field* membentuk suatu *record*.

4. *Record*

Suatu *record* menggambarkan satu kesatuan data yang sejenis, dimana kumpulan dari file-file membentuk *database*.

5. *File*

Suatu file menggambarkan kumpulan dari beberapa *record* yang dapat menampung data-data.

6. *Database*

Suatu database menggambarkan data yang saling berhubungan antara satu dengan yang lainnya.

II.8. Normalisasi

Normalisasi merupakan sebuah teknik dalam desain logika sebuah *database*, teknik pengelompokan atribut dari suatu relasi sehingga membentuk struktur relasi yang baik (tanpa redundansi).

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.9. 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.

II.10. Pengenalan Bahasa Pemrograman Visual Basic Net 2008

Menurut Wahana Komputer (2008, 2010) Visual Basic.Net adalah suatu lingkungan (*environment*) terintegrasi untuk membangun dan melakukan uji coba (*testing and debugging*) berbagai macam aplikasi, contohnya aplikasi-aplikasi Windows.

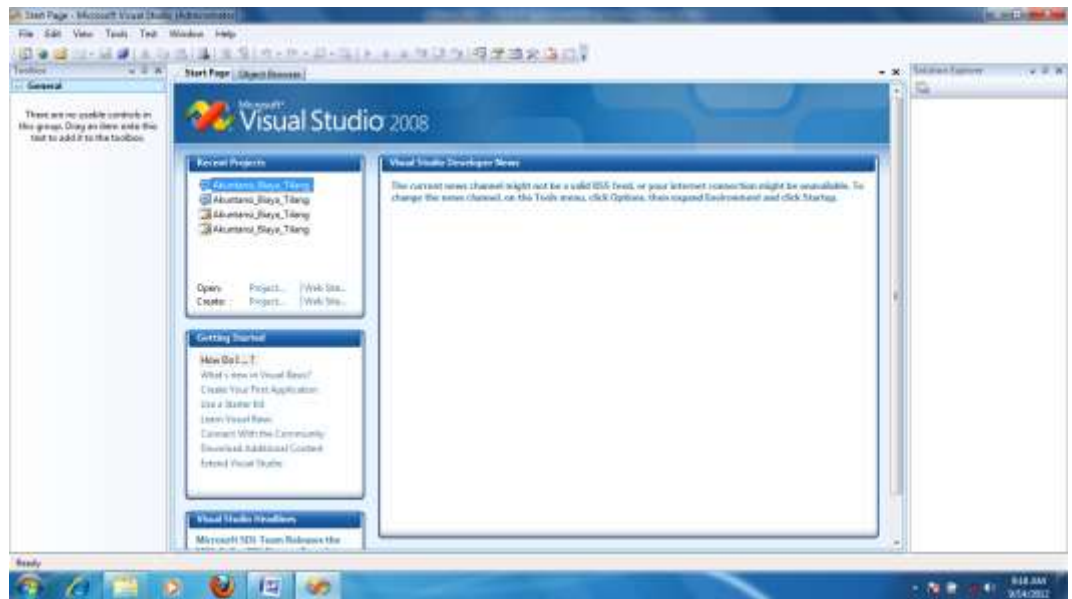
Menurut Hendrayuni (2008, 2010 :1) Microsoft visual basic 2008 merupakan bagian dari kelompok bahasa pemrograman. Visual basic studio 2008 yang dikembangkan oleh Microsoft.

.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 :

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

2.9.1. Membuat Project Baru

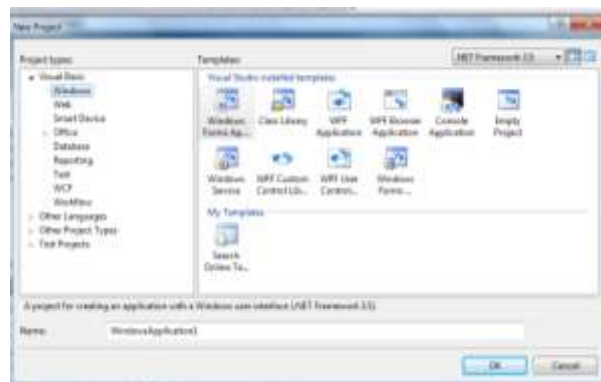
Mulailah dengan membuka Visual Studio .NET, maka akan tampil Start Page yang menampilkan beberapa project terakhir yang anda akses. Pada bagian kiri terdapat beberapa baris hyperlink yang menghubungkan anda dengan beberapa informasi penting.



Gambar II.10. Start Page Visual Basic 2008

**Sumber : “Dasar –dasar pemrograman microsoft visual basic
2008(Hendrayudi ; 2011 : 2)”**

Untuk membuat Project baru pilih New Project untuk menampilkan dialog New Project seperti berikut :



Gambar II.11. dialog New Project

***Sumber : “Dasar –dasar pemrograman microsoft visual basic
2008(Hendrayudi ; 2011 : 3)”***

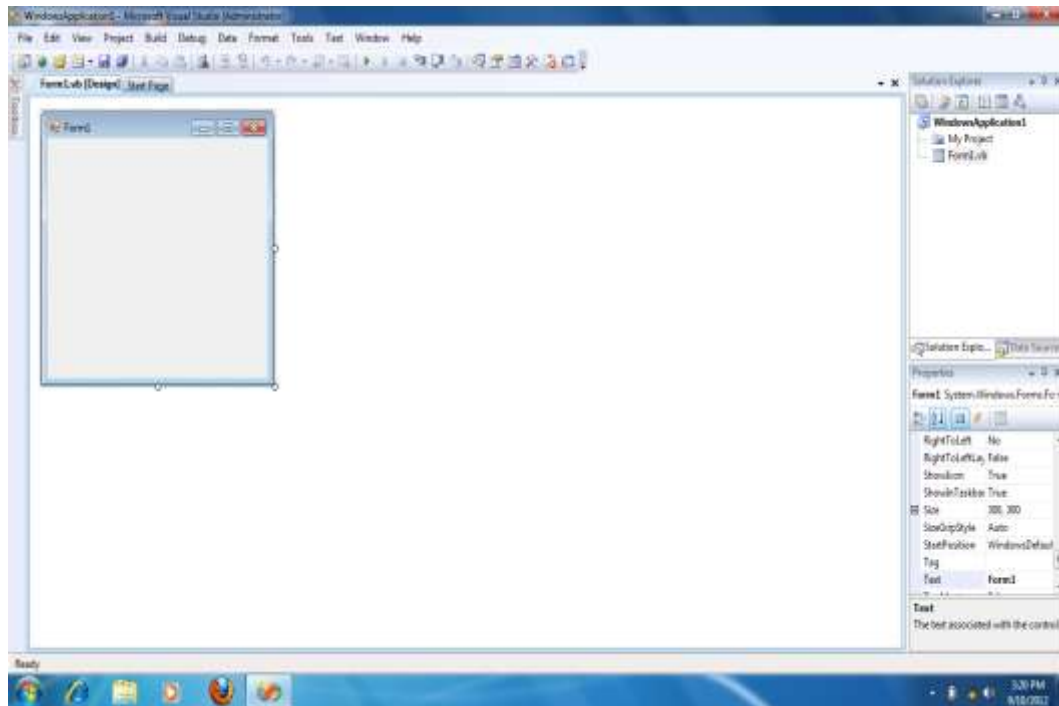
Dalam dialog New Project anda dapat memilih jenis aplikasi yang akan dibuat termasuk bahasa pemrograman digunakan. Jenis aplikasi yang dapat dibuat adalah :

- a. *Windows Application* : adalah aplikasi yang paling umum dibuat, menggunakan interface windows. Biasanya Windows Application merupakan interface aplikasi sedangkan logic aplikasi terdapat di dalam Class Library. Windows Application dapat berisi form, class, XML file, maupun file VB Script dan Jscript.
- b. *Class Library* : merupakan fondasi dasar untuk membuat komponen yang menjalankan fungsi tertentu. Class merupakan fondasi dasar untuk membentuk obyek dalam pemrograman berorientasi obyek. Class Library tidak memiliki interface tertentu seperti form, tetapi dapat diakses oleh aplikasi lain untuk menjalankan berbagai fungsi yang terdapat di dalamnya. Class Library dapat disamakan dengan teknologi ActiveX DLL (.dll) dan ActiveX EXE dalam pemrograman VB6.
- c. *Windows Control Library* : tidak puas dengan built in control yang disediakan VS .NET ? Anda dapat berkreasi membuat kontrol sendiri dan memasukkan berbagai fungsi yang anda inginkan di dalam kontrol tersebut. Fasilitas untuk

membuat kontrol tersebut adalah Windows Control Library. Kontrol ini sama dengan ActiveX Control (.ocx) dalam pemrograman VB6.

- d. *ASP .NET Web Application* : adalah project yang digunakan untuk membuat aplikasi web. Teknologi yang digunakan adalah ASP .NET yang memiliki berbagai kelebihan dibandingkan ASP klasik. Perubahan utamanya adalah dapat diprogram menggunakan berbagai bahasa .NET seperti VB, C++, C# maupun J#. ASP .NET juga menyediakan berbagai kontrol yang bersifat event drivent programming sehingga lebih menghemat waktu pembuatan aplikasi.
- e. *ASP .NET Web Service* : Web service merupakan salah satu ide utama dalam .NET. Anda dapat membuat web service dan meletakkannya di web server untuk diakses berbagai aplikasi. Sebuah web service dapat diakses oleh aplikasi windows, web, console, maupun mobile device. Web service hampir sama dengan Class Library, perbedaan utamanya adalah web service tersebut diletakkan di web server sehingga dapat diakses dengan lebih mudah dan tidak terbatas pada aplikasi berbasis windows saja.
- f. *Console Application* : merupakan aplikasi dengan tampilan text mode atau DOS. Aplikasi jenis ini biasa digunakan sebagai monitoring service atau remote application dimana sumber daya komputer dan bandwidth sangat terbatas.
- g. *Windows Service* : adalah aplikasi yang berjalan sebagai service di windows, yang di load bersamaan dengan proses start up windows. Aplikasi ini berjalan di background dan biasanya tidak memiliki interface. Penerapan aplikasi ini misalnya untuk pembuatan scanning antivirus, server FTP, dan remote server.

- h. *Web Control Library* : Hampir sama dengan Windows Control Library tetapi digunakan untuk aplikasi web.



Gambar II.12. Form aplikasi

***Sumber : “Dasar –dasar pemrograman microsoft visual basic
2008(Hendrayudi ; 2011 : 3)”***

Gambar di atas menjelaskan garis besar IDE yang biasa digunakan. Di bagian atas terdapat toolbar yang sudah tidak asing lagi, mencakup berbagai fasilitas editing seperti *cut*, *copy*, *paste*, dan tombol *Start*. Di bagian kanan

terdapat *Solution Explorer* yang menampilkan berbagai obyek dalam aplikasi seperti form, *class* dan *component*.

Anda dapat berpindah dari *mode View Code* ke *View Designer* atau sebaliknya. *View Code* menampilkan *coding* untuk obyek-obyek dalam *aplikasi*, sedangkan *View Designer* menampilkan *representasi* fisik obyek tersebut.

Di bawah *Solution Explorer* terdapat *Properties Window* yang berisi properti obyek yang sedang aktif di bagian *designer*. Anda dapat mengatur properti obyek di bagian ini baik dari segi tampilan maupun perilaku obyek tersebut dalam aplikasi. Selain menetapkan properti di bagian ini dapat pula mengeset properti secara run time dengan menggunakan *coding*. Bagian yang sering digunakan adalah *Toolbox* yang terdapat di sisi kiri, yang pada gambar di atas sengaja dihide untuk menghemat tempat.