BAB II

TINJAUAN PUSTAKA

II.1. Perancangan

Perancangan adalah spesifikasi umum dan terinci dari pemecahan masalah berbasis komputer yang telah dipilih selama tahap analisis. Berdasarkan definisi tersebut dapat disimpulkan bahwa perancangan adalah kemampuan untuk membuat alternatif pemecahan masalah berbasis komputer selama tahap analisis. (Azhar Susanto; 2004:332)

II.2. Aplikasi

Definisi aplikasi menurut Griffin dkk (2006:86) "aplikasi merupakan paket software yang ditulis oleh orang lain". Definisi lain dari aplikasi menurut Kadir (2005:222) "perangkat lunak aplikasi (*aplication software*) adalah program yang biasa dipakai oleh pemakai untuk melakukan tugas-tugas yang spesifik; misalnya untuk membuat dokumen, memanipulasi foto, atau membuat laporan keuangan".

Berdasarkan definisi di atas, penulis menyimpulkan bahwa aplikasi merupakan software yang dibuat oleh orang lain atau programmer yang memiliki fungsi tertentu untuk melakukan tugas-tugas tertentu.

II.3. Steganografi

Steganografi berasal dari bahasa Yunani yaitu *stegos* yang berarti penyamaran dan *graphia* yang berarti tulisan. *Steganography* adalah teknik

menyembunyikan data rahasia di dalam wadah atau media (*cover*) digital sehingga keberadaan data rahasia tersebut tidak diketahui oleh orang. Hal yang fundamental yang diperlukan dalam metode *steganography* adalah tidak terlihat jelas, artinya pesan yang disisipkan pada media tertentu harus tidak terlihat secara kasat mata. Dua hal lain yang juga dibutuhkan adalah cara untuk memaksimalkan kapasitas pesan yang dapat disisipkan, dan juga menjamin keamanan (*security*) dari pesan yang disisipkan tersebut. Metode penyisipan menggunakan *spread spectrum* adalah metode yang umum dan mudah mengimplementasikannya dalam menyisipkan pesan pada citra digital. Akan tetapi berapa kapasitas maksimum pesan yang dapat disisipkan pada setiap piksel citra digital merupakan hal yang sangat penting dan perlu dipertimbangkan dengan kualitas citra yang dihasilkan setelah disisipi pesan. Ada dua jenis teknik *steganography* pada citra digital, yaitu *spatial-domain* dan *frequency domain*. *Spatial-domain* didasarkan pada penyisipan pesan pada intensitas setiap piksel citra digital secara langsung. Sedangkan pada *frequency domain*, pertama-tama intensitas setiap piksel citra digital ditransformasikan ke *frequency-domain* tertentu, dan setelah itu pesan disisipkan pada koefisien yang diperoleh dari hasil transformasi Tersebut.

Menurut (Rinaldi, 2006) ada beberapa hal yang diperlukan untuk menyembunyikan pesan yaitu:

1. Algoritma Penyisipan (*Embeding Algorithm*).

   Algoritma ini digunakan untuk menyisipkan suatu pesan yang disembunyikan ke dalam suatu data yang akan dikirim. Proses penyisipan ini diproteksi oleh

sebuah *key-word* sehingga hanya orang-orang yang mengetahui *key-word* ini yang dapat membaca pesan yang disembunyikan tersebut.

2. Fungsi Detektor (*Detector Function*).

   Fungsi Detektor ini adalah untuk mengembalikan pesan-pesan yang disembunyikan tersebut.

3. *Carrier Document.*

   Merupakan dokumen yang berfungsi sebagai media yang digunakan untuk menyisipkan informasi. Dokumen ini dapat berupa *file-file* seperti *file* audio,video atau *citra(gambar).*

4. *Key*

   Merupakan kata kunci yang ikut disisipkan kedalam dokumen berguna dan dipakai sebagai proses verifikasi sewaktu informasi akan ditampilkan atau diuraikan.

5. *Secret Message/ Plaintext*

   Merupakan pesan rahasia yang akan disisipkan kedalam *carrier document*. Pesan inilah yang tidak ingin terlihat dan terbaca oleh orang yang tidak berkepentingan.

Menurut (Rinaldi, 2006), ada beberapa hal yang harus dimiliki oleh pesan yang disembunyikan yaitu :

1. *Robustness*

Pesan yang disembunyikan dan disisipkan pada data tidak boleh merubah informasi lain pada data. Suatu pesan yang disembunyikan tersebut dikatakan

kuat jika pesan tersebut hanya terdeteksi oleh peralatan yang dipercaya dan pesan tersebut tidak dapat dimodifikasi atau dihapus.

2. *Undetectability*

Suatu pesan yang disembunyikan seharusnya tidak dapat dideteksi oleh orang yang tidak berkepentingan. Suatu pesan yang disembunyikan tersebut tidak terdeteksi (*undetectability*) jika pesan yang disembunyikan tersebut mempunyai model yang sama. Sebagai contoh, jika metode *Steganographic* menggunakan komponen suara pada data digital untuk menyisipkan pesan rahasia (pesan yang disembunyikan) tersebut, maka suara tersebut tidak boleh memiliki perbedaan dengan suara lain pada data digital yang dikirimkan.

3. *Invisibility*

Konsep ini menggunakan kelemahan manusia, yaitu kelemahan sistem penglihatan dan pendengaran pada manusia. Suatu pesan rahasia dikatakan *invisible* jika rata-rata panca indera manusia tidak dapat membedakan suatu data yang mengandung pesan rahasia dan data yang tidak memiliki pesan rahasia ketika dikirimkan.

4. *Security* Suatu pesan rahasia yang disisip dikatakan aman jika pesan rahasia tersebut tidak dapat diubah atau dihapus ketika pesan rahasia tersebut dibaca.

5. *Secure Black-Box Public Detector*

Yaitu suatu detektor pesan yang diterapkan pada suatu perangkat keras (*hardware)*. Perangkat tersebut tidak dapat dibongkar kembali secara *hardware.* Kunci rahasia untuk membaca pesan yang disembunyikan tersebut

akan disimpan pada *black-box* tersebut dan kunci tersebut tidak dapat diganti atau dihapus. Dengan hadirnya *black-box* ini, harus mampu menahan serangan dari pihak luar untuk mengambil kunci rahasia atau menghapus informasi yang terkandung didalamnya.

6. *Secure Public Detector*

Merupakan konsep detektor yang lebih kuat dari semua detektor yang diketahui oleh masyarakat umum. Detektor ini terdapat pada sebuah aplikasi yang memiliki tingkat teknologi yang tinggi. Aplikasi ini dapat mem*filter* gambar-gambar yang memiliki tanda khusus, menampilkan informasi dari sipembuat untuk setiap gambar dan lain-lain. (Sumber : Adi Nugroho, 2006 : 77)

II.3.1. Sejarah Steganografi

Catatan tertua mengenai penggunaan steganografi tercatat pada masa Yunani kuno. Pada saat itu, penguasa Yunani, Histiaues, sedang ditawan oleh Raja Darius di Susa. Histiaeus ingin mengirim pesan rahasia kepada menantunya, Aristagoras, di Miletus. Untuk itu, Histiaeus mencukur habis rambut budaknya dan menatokan pesan rahasia yang ingin dikirim di kepala budak tersebut. Setelah rambut budak tadi tumbuh cukup lebat, barulah ia dikirim ke Miletus.

Menurut Rinaldi, 2006, terdapat beberapa istilah yang berkaitan dengan steganografi yaitu :

1. *Carrier file* : *file* yang berisi pesan rahasia tersebut.

2. *Steganalysis* : proses untuk mendeteksi keberadaan pesan rahasia dalam suatu *file*.
3. *Stego-medium* : media yang digunakan untuk membawa pesan rahasia.
4. *Redundant bits* : sebagian informasi yang terdapat di dalam *file* yang jika dihilangkan tidak akan menimbulkan kerusakan yang signifikan (setidaknya indera manusia).
5. *Payload* : informasi yang akan disembunyikan. (Sumber : Sandro Sembiring, 2013 : 46)

II.4. *End of File* (EOF)

Metode *End of File* (EOF) Metode ini merupakan metode pengembangan LSB.Dalam metode ini pesan disisipkan diakhir berkas.Pesan yang disisipkan dengan metode ini jumlahnya tidak terbatas.Akan tetapi efek sampingnya adalah ukuran berkas menjadi lebih besar dari ukuran semula. Ukuran berkas yang terlalu besar dari yang seharusnya, tentu akan menimbulkan kecurigaan bagi yang mengetahuinya. Misalnya pada sebuah citra skala keabuan 6x6 piksel disisipkan pesan yang berbunyi "#aku". Kode ASCII dari pesan tersebut adalah :

35 97 107 117

Misalkan matriks tingkat derajat keabuan citra sebagai berikut :

196 10 97 182 101 40

67 200 100 50 90 50

25 150 45 200 75 28

176 56 77 100 25 200

101 34 250 40 100 60

44 66 99 125 190 200

Kode biner pesan disisipkan di akhir citra, sehingga citra menjadi :

196 10 97 182 101 40

67 200 100 50 90 50

25 150 45 200 75 28

176 56 77 100 25 200

101 34 250 40 100 60

44 66 99 125 190 200

35 97 107 117

Kriteria yang harus diperhatikan dalam penyembunyian data adalah[7] [4] :

1. Tidak dapat dipersepsi (*Impercepbility*)

   Keberadaan data rahasia tidak dapat dipersepsi oleh indera manusia.Jika pesan disisipkan ke dalam sebuah citra, citra yang telah disisipi pesan harus tidak dapat dibedakan dengan citra asli saat dilihat dengan mata.Begitu pula dengan suara, telinga haruslah mendapati tidak ada perbedaan antara suara asli dan suara yang telah disisipi.

2. Ketepatan (*Fidelity*)

   Kualitas citra penampung tidak jauh berubah setelah penyisipan data rahasia.Pengamat tidak mengetahui kalau di dalam citra tersebut terdapat data rahasia.

3. Kapasitas (*Capacity*)

Berhubungan dengan jumlah informasi yang dapat disisipkan ke dalam media penampung.

4. Ketahanan (*Robustness*)

Data yang disembunyikan harus tahan (*robust*) terhadap berbagai operasi manipulasi yang dilakukan pada citra penampung.

5. Tidak terdeteksi (*Undetecability*)

Kemampuan untuk menghindari deteksi oleh indera manusia maupun analisis statistik.

6. Pemulihan (*Recovery*)

Data yang disembunyikan harus dapat diungkapkan kembali (reveal).

Dalam melakukan proses s*teganografi*, ada beberapa faktor yang saling berkompetisi satu sama lain (*trade-off*), artinya saat salah satu faktor ditingkatkan maka kemungkinan faktor lain akan mengalami penurunan. (Sumber : Mukharrom Edisuryana dkk, 2013, vol 2)

II.5. Kriptografi

Kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan berita [**bruce Schneier** – *Applied Cryptography*]. "Crypto" berarti "secret" (rahasia) dan "graphy" berarti "writing" (tulisan). Sebuah algoritma kriptografik (cryptographic algorithm), disebut cipher, merupakan persamaan matematik yang digunakan untuk proses enkripsi dan dekripsi. Biasanya kedua persamaan matematik (untuk enkripsi dan dekripsi) tersebut memiliki hubungan matematis yang cukup erat.

Ada empat tujuan mendasar dari ilmu kriptografi ini yang juga merupakan aspek keamanan informasi yaitu :

Kerahasiaan, adalah layanan yang digunakan untuk menjaga isi dari informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka/mengupas informasi yang telah disandi.

1. Integritas data, adalah berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, system harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubsitusian data lain kedalam data yang sebenarnya.
2. Autentikasi, adalah berhubungan dengan identifikasi/pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan melalui kanal harus diautentikasi keaslian, isi datanya, waktu pengiriman, dan lain-lain.
3. Non-repudiasi., atau nirpenyangkalan adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman/terciptanya suatu informasi oleh yang mengirimkan/membuat. (Sumber : Rinaldi Munir: 2006, 9)

Tujuan pokok dari *kriptografi* adalah untuk mencapai keempat tujuan di atas baik secara teori maupun prakteknya. *Kriptografi* merupakan ilmu yang berhubungan dengan pencegahan dan deteksi terhadap penjiplakan dan aktivitas kriminal lainnya.

Proses yang dilakukan untuk mengamankan sebuah pesan (yang disebut *plaintext*) menjadi pesan yang tersembunyi (disebut *ciphertext*) adalah enkripsi (*encryption*). *Ciphertext* adalah pesan yang sudah tidak dapat dibaca dengan mudah. Menurut ISO 7498- 2, terminologi yang lebih tepat digunakan adalah "encipher". Kriptografi bisa dilakukan dengan algoritma sandi. Algoritma tersebut harus memiliki kekuatan untuk melakukan (dikemukakan oleh Shannon):

1. Konfusi/pembingungan (*confusion*), dari teks terang sehingga sulit untuk direkonstruksikansecara langsung tanpa menggunakan algoritma dekripsinya.
2. Difusi/peleburan (*difusion*), dari teks terang sehingga karakteristik dari teks terang tersebut hilang.

Sehingga dapat digunakan untuk mengamankan informasi. Pada implementasinya sebuah algoritmas sandi harus memperhatikan kualitas layanan/*Quality of Service* atau QoS dari keseluruhan sistem dimana dia diimplementasikan. Algoritma sandi yang handal adalah algoritma sandi yang kekuatannya terletak pada kunci, bukan pada kerahasiaan algoritma itu sendiri. Teknik dan metode untuk menguji kehandalan algoritma sandi adalah kriptanalisa.

Secara umum berdasarkan kesamaan kuncinya, algoritma sandi dibedakan menjadi :

1. Kunci-simetris/*symetric-key*, sering disebut juga algoritma sandi konvensional karena umumnya diterapkan pada algoritma sandi klasik.
2. Kunci-asimetris/*asymetric-key*

Berdasarkan arah implementasi dan pembabakan jamannya dibedakan menjadi :

1. Algoritma sandi klasik classic *Cryptography*.
2. Algoritma sandi modern modern *Cryptography*. (Rifki Sadikin: 2012, 9)

II.6. Audio

Audio atau suara merupakan gelombang yang mengandung sejumlah komponen penting (amplitudo, panjang gelombang dan frekuensi) yang dapat menyebabkan suara yang satu berbeda dari suara lain. Amplitudo adalah kekuatan atau daya gelombang sinyal. Tinggi gelombang yang bisa dilihat sebagai grafik, Gelombang yang lebih tinggi diinterpretasikan sebagai volume yang lebih tinggi, Suara beramplitudo lebih besar akan terdengar lebih keras.

Gelombang suara adalah gelombang yang dihasilkan dari sebuah benda yang bergetar. Sebagai contoh, senar gitar yang dipetik, gitar akan bergetar dan getaran ini merambat di udara, atau air, atau material lainnya. Satu - satunya tempat dimana suara tak dapat merambat adalah ruangan hampa udara. Gelombang suara ini memiliki lembah dan bukit, satu buah lembah dan bukit akan menghasilkan satu siklus atau periode. Siklus ini berlangsung berulang-ulang, yang membawa pada konsep frekuensi.

Audio digital merupakan versi digital dari suara analog. Pengubahan suara analog menjadi suara digital membutuhkan suatu alat yang disebut *Analog to Digital Converter* (ADC). ADC akan mengubah amplitudo sebuah gelombang analog ke dalam waktu interval (sampel) sehingga menghasilkan penyajian digital dari suara.

II.6.1. Format File Audio

WAV adalah *format file audio standar Microsoft* dan IBM untuk personal computer (PC), biasanya menggunakan pengkodean PCM (*Pulse Code Modulation*). WAV adalah data tidak terkompres sehingga seluruh sampel audio disimpan semuanya di harddisk. Perangkat lunak yang dapat menciptakan WAV dari sinyal analog misalnya adalah *Windows Sound Recorder*. WAV jarang sekali digunakan di internet karena ukurannya yang relatif besar dengan batasan maksimal untuk file WAV adalah 2GB.

Secara umum data audio digital dari WAV memiliki karakteristik yang dapat dinyatakan dengan parameter-parameter berikut:

1. Laju sampel (sampling rate) dalam sampel/detik, misalnya 22050 atau 44100 sampel/detik.
2. Jumlah bit tiap sampel, misalnya 8 atau 16 bit.
3. Jumlah kanal (channel), yaitu 1 untuk mono dan 2 untuk stereo.

File MP3 MPEG - 1 audio layer III atau yang lebih dikenal dengan MP3, adalah pengkodean dalam digital audio dan juga merupakan format kompresi audio yang memiliki sifat "menghilangkan". Istilah menghilangkan yang dimaksud adalah kompresi audio ke dalam format mp3 menghilangkan aspek-aspek yang tidak signifikan pada pendengaran manusia untuk mengurangi besarnya file audio.

Sejarah MP3 dimulai dari tahun 1991 saat proposal dari Phillips (Belanda), CCET (Perancis), dan Institut für Rundfunktechnik (Jerman) memenangkan proyek untuk DAB (*Digital Audio Broadcast*). Produk mereka

seperti Musicam (lebih dikenal dengan layer 2) terpilih karena kesederhanaan, ketahanan terhadap kesalahan, dan perhitungan komputasi yang sederhana untuk melakukan pengkodean yang menghasilkan keluaran yang memiliki kualitas tinggi. Pada akhirnya ide dan teknologi yang digunakan dikembangkan menjadi MPEG -1 audio layer 3. MP3 adalah pengembangan dari teknologi sebelumnya sehingga dengan ukuran yang lebih kecil dapat menghasilkan kualitas yang setara dengan kualitas CD. (Mhd Ridwan, 2007 : 7)

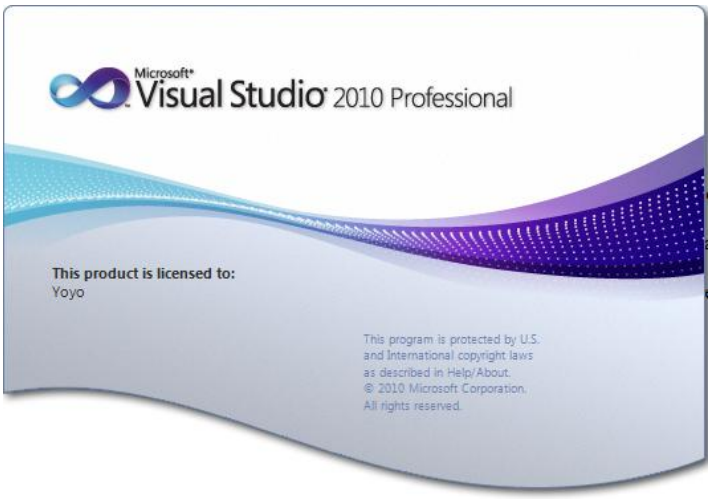
II.7. *VB.net* 2010

Visual Studio 2010 merupakan merupakan sebuah lingkunan kerja (*Integrated Development Environtment (IDE*)*)* yang digunakan untuk pemrograman .NET yang dapat digunakan untuk beberapa bahasa pemrograman, seperti *Visual Basic* (VB), C# (baca C *Sharp*), Visual C++, J# (baca J *sharp*), F# (baca F *Sharp*), dan lain-lain. (Wahana Komputer, 2012:2)

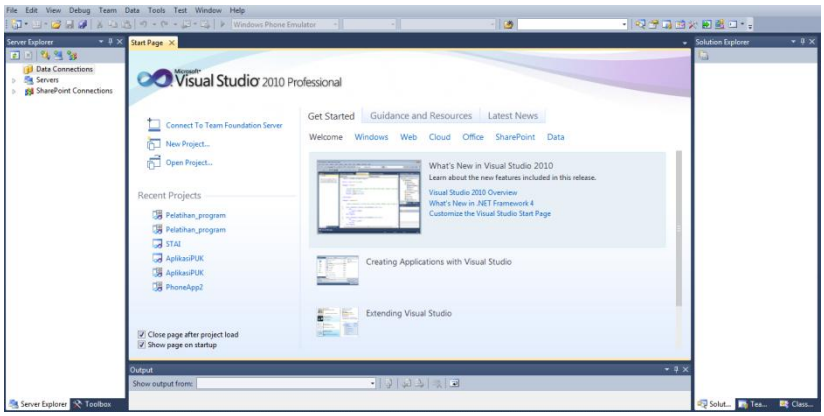
Teknologi .NET *Framwork* adalah sebuah *Application Programming Language* (API), yaitu kumpulan kelas atau sebuah pustaka inti yang digunakan untuk melakukan pemrograman .NET. Kelas-kelas *core* (inti) .NET ini menyediakan berbagai macam kelass yang berfungsi untuk melakukan apapun yang diinginkan dilingkungan *windows*, ataupun lingkungan web, mulai dari bekerja dengan data hingga bekerja dengan *form* (jendela) dan kontrol. (Wahana Komputer, 2012:2)

Visual Studio Profesional 2010 menyediakan berbagai *tool* yang lengkap bagi para pengembang untuk membangun aplikasi yang berjalan di .NET

Framwork. Berbagai *tool*, antara lain *Toolbox* yang berisi komponen *visual*, sehingga anda tinggal *drag* and *drop* komponen dan *Visual Studio* 2010 akan menuliskan kode untuk anda. (Wahana Komputer, 2012:7)



Gambar II.1 *Splash Screen Visual Studio* 2010
(Sumber : Wahana Komputer, 2012, 17)



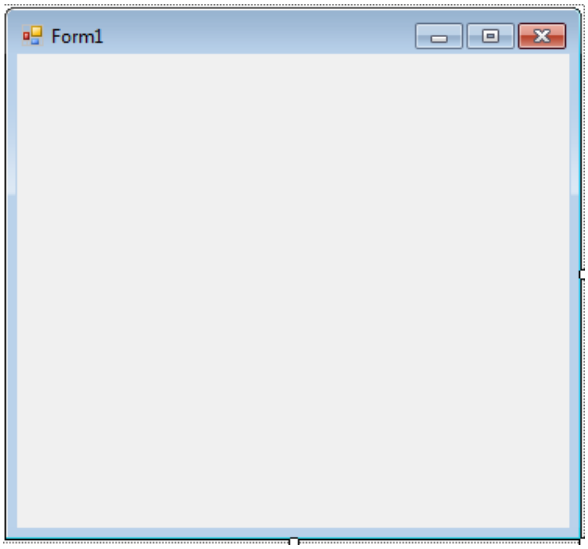
Gambar II.2 IDE *Visual Studio* 2010
(Sumber : Wahana Komputer, 2012, 17)

Adapun objek-objek yang digunakan dalam program ini :

1. *Form*

Form merupakan komponen VB yang memiliki sifat container, karena fungsi utama dari form adalah sebagai tempat komponen VB yang lain. Pembuatan

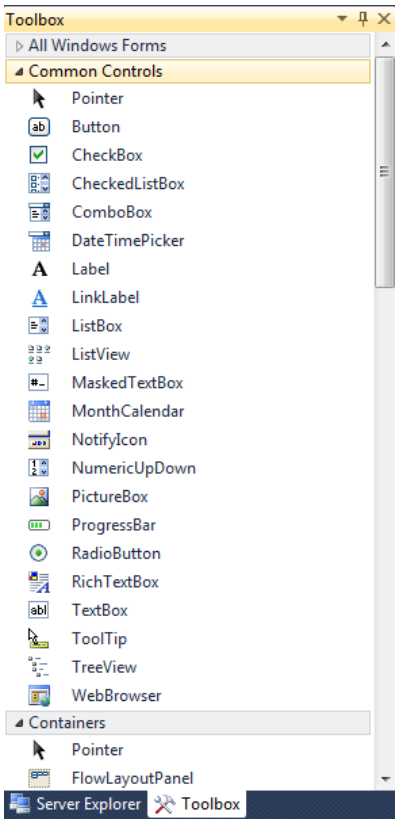
form dalam VB dapat kita lakukan dengan menggunakan menu **Project** > **Add Windows Form**.



Gambar II.3 Form Design
(Sumber : Wahana Komputer, 2012, 16)

2. ToolBox

ToolBox merupakan jendela yag berisikan grup dari bermacam –macam control dan komponen yang dapat dipasang di dalam form. Kontrol seperti *Textbox, Button, Radio Button, Checkbox, Combo Box* dan lain-lain ditambahkan ke dalam *form* dengan *cara drag* dan *drop*, atau mengklik dan menggoreskannya pada *form*.



Gambar II.4 *ToolBox*
(Sumber : Wahana Komputer, 2012, 17)

Beberapa kontrol pada ToolBox :

a. *Button* : komponen yang digunakan untuk memberikan aksi saat ditekan. *Button* menjalankan proses menyimpan, mengubah, menghapus, dan lain-lain.

b. *Label* : untuk menampilkan teks. Biasa digunakan untuk memberikan informasi pada kontrol lain.

c. *Textbox* : untuk input data. Komponen ini paling sering digunakan bersama button.

d. *CheckBox* : digunakan untuk memberikan pilihan *input* kepada *user user* dapat memilih lebih dari 1 item data.

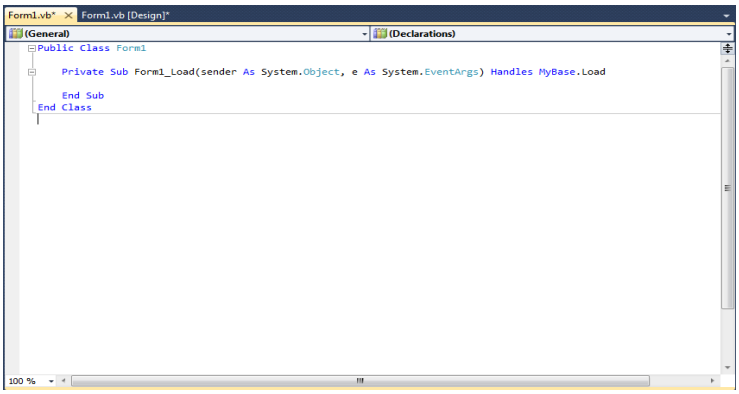
e. *Combobox* : komponen yang menampilkan pilihan secara drop down.

f. *ListBox* : sebuah kotak yang di dalamnya berisi item-item, *ListBox* menampilkan item lebih dari 1.

g. *RadioButton* : untuk memilih pada suatu daftar pilihan, dengan satu pilihan yang dapat dipilih. (Wahana Komputer, 2012:114)

3. Jendela Editor Kode

Agar kontrol dapat bekerja sesuai fungsinya dan interaktif, Anda harus menambahkan sebuah kode di belakang layar control tersebut. Untuk menulis kode ini anda harus masuk ke dalam jendela editor kode. Untuk membukanya, klik 2 kali control yang akan ditambahi kode.(Wahana Komputer, 2012:19)



Gambar II.5 Jendela Editor Kode
(Sumber : Wahana Komputer, 2012, 19)

II.8. UML

UML singkatan dari *Unifed Modeling Language* yang berarti bahasa pemodelan standar. (Chonoles, 2003:bab 1) mengatakan sebagai bahasa, berarti UML memiliki sintaks dan semantik. Ketika kita membuat model menggunakan konsep UML ada aturan-aturan yang harus diikuti.(Menggunakan UML, 2011, 6) Blok pembangun utama UML adalah diagram. Beberapa diagram ada yang rinci (jenis *timing diagram*) dan lainnya ada yang bersifat umum (misalnya diagram kelas). Para pengembang sistem berorientasi objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasi sistem yang mereka rancang. UML merupakan alat komunikasi yang konssisten dalam mensuport para pengembang sistem saat ini. (Evi Triandini dan Gede Suardika: 2012: 7)

II.8.1. Diagram – diagram UML

Beberapa literature menyebut bahwa UML menyediakan Sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung. Jenis diagram itu antara lain:

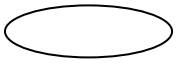
1. Diagram kelas. Bersifat statis.
2. Diagram Paket (*Package Diagram*). Bersifat statis.
3. Diagram *Use-Case*. Bersifat statis.
4. Diagram interaksi dan *Squence* (urutan). Bersifat dinamis.
5. Diagram Komunikasi (*Communication Diagram).* Bersifat dinamis.
6. Diagram Statechart (*Statechart Diagram*). Bersifat dinamis.
7. Diagram Aktivitas (*Acctivity Diagram*). Bersifat dinamis.

8. Diagram Komponen (*Component Diagram*). Bersifat statis.
9. Diagram *Deployment* (*Deployment Diagram*). Bersifat statis. (Evi Triandini dan Gede Suardika: 2012: 12)

II.8.2. Diagram *Use Case*

John Satzinger, 2010, dalam buku *System Analysis and Design in a Changing World* menyatakan bahwa "*Use Case* adalah sebuah kegiatan yang dilakukan oleh sistem biasanya dalam menanggapi permintaan dari pengguna sistem." (Evi Triandini dan Gede Suardika, 2012: 18)

Tabel II.1 Tabel Simbol *Use Case*

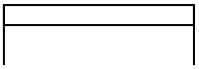
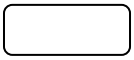
Notasi	**Keterangan**	**Simbol**
Actor	Stick Figure mewakili sebuah peran	
Garis Penghubung	Menunjukan actor mana yang menjalankan use case yang mana	
Use Case	Bentuk fungsionalitas dari sebuah system	

(Sumber : Evi Triandini dan Gede Suardika, 2012: 18)

II.8.3. *Activity Diagram*

John Satzinger, 2010, dalam buku *System Analysis and Design in Changing World* menyatakan bahwa "*Activity Diagram* adalah sebuah Diagram alur kerja yang menjelaskan berbagai kegiatan pengguna (atau sistem), Orang yang melakukan masing-masing aktivitas, dan aliran sekuensial dari aktivitas-aktivitas tersebut."(Evi Triandini dan Gede Suardika: 2012: 38)

Tabel II.2 Tabel Simbol *Activity Diagram*

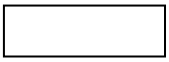
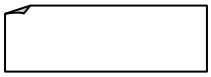
Notasi	Keterangan	Simbol
Swimlane	Mewakili agen yang melakukan aktivitas	
InitialState	Awal dari alur kerja	
FinalState	Akhir dari alur kerja	
ActionState	Aktivitas tersendiri dalam alur kerja	
Decision	Titik pengambil keputusan di mana aliran proses tersebut akan mengikuti satu jalur atau jalur lainnya	
Transition	Urutan diantara aktivitas	
Synchronization	Membagi alur kerja menjadi beberapa alur yang berbarengan ataupun menggabungkan lagi alur yang berbarengan	

(Sumber : Evi Triandini dan Gede Suardika, 2012 : 38)

II.8.4. *Sequence* Diagram

Menurut John Satzinger, 2010 dalam buku *System Analysis and Design in a Changing World* menyatakan bahwa "*System Squence* Diagram (SSD) adalah diagram yang digunakan untuk mendefenisikan input dan output serta urutan interaksi antara pengguna dan sistem untuk sebuah *use case*." (Evi Triandini dan Gede Suardika: 2012: 72)

Tabel II.3 Tabel Simbol *Squence Diagram*

Notasi	**Keterangan**	**Simbol**
Actor	Peran yang berinteraksi dengan system	
Kotak Berlabel	Objek yang mewakili keseluruhan sistem yang terotomalisasi	
Anak Panah	Mewakili message yang dikirim atau diterima oleh actor dari system	
Garis Putus-putus Vertikal	Perpanjangan objek tersebut, baik actor maupun objek, sepanjang durassi dari sequence diagram	
Message diberi label	Menggambarkan maksud message dan input apapun yang sedang dikirim	

(Sumber : Evi Triandini dan Gede Suardika, 2012 : 72)