BAB II

TINJAUAN PUSTAKA

2.1 Analisis terdahulu

Sebagai pembanding penelitian yang hendak dilakukan maka penting untuk mencantumkan beberapa penelitian terdahulu sebagai berikut:

1. Penelitian yang dilakukan oleh Eko Puji Laksono (2014), "Analisis komparasi algoritma kriptografi antara metode DES dan AES". Penelitian ini fokus menganalisis pada perbandingan antara program Des dan AES.

   Hasil penelitian :

   1. Analisis enkripsi dan deskripsi menggunakan program DES lebih sederhana dibandingkan AES.
   2. Langkah yang panjang membuat enkripsi dan deskripsi AES lebih aman
   3. Penggunaan kriptografi AES 192 bit menghasilkan literasi lebih panjang dan komplek sehingga sulit untuk diretas

   Kesimpulan yang didapat :

   1. Analisis enkripsi dan deskripsi menggunakan DES lebih sederhana dibandingkan AES. Data DES diolah hanya dengan 4 langkah tanpa menggunakan ronde, sehingga hasilnya lebih sederhana dalam bentuk huruf yang jumlahnya sama dengan data yang diolah
   2. DES tidak memiliki ronde. Enkripsi AES memiliki 9 ronde dan deskripsi AES memiliki 9 ronde
   3. Analisis enkripsi dan deskripsi AES menggunakan algoritma kriptografi AES dengan panjang kunci 128 bit memiliki literasi yang cukup panjang sehingga lebih aman
   4. Karakteristik enkripsi DES menggunakan rumus C=P+K+1)mod26 dan deskripsi DES menggunakan rumus P=(C+26-K-1)mod26. Karakteristik enkripsi AES setiaprondenya memiliki SubBytes, ShiftRows, MixColomn, dan RoundKey. Karakteristik deskripsi AES setiap rondenya memiliki invShiftRows, invSubBytes, RoundKey dan addRoundKey.

2. Penelitian yang dilakukan oleh Yusuf Kurniawan (2007), “Perbandingan analisis sandi linier terhadap AES, DES dan AE1”. Penelitian ini mengkaji tentang perbandingan analisis sandi AES, DES dan AE1.

   Hasil penelitian :

   1. AES memiliki ketahanan lebih besar untuk menghadapi ASL.
   2. Ketahanan cipher terhadap analisis sandi tidak otomatis sehingga tahan terhadap sandi lain.
   3. AE1 memiliki ketahanan yang lebih besar.
   4. Pembuatan keamanan algoritma kriptografi lebih sulit dari pada pembuktiaannya.

   Kesimpulan yang didapat :

   1. AES-128 memiliki ketahanan yang besar untuk menghadapi ASL, karena analisis sandi tersebut hanya mampu memecahkan AES hingga 6 ronde, sedangkan AES-128 memiliki 10 ronde. Bandingkan dengan DES lengkap yang dapat dipecahkan ASL dengan 243 plaintext, di mana DES memiliki masukan 64 bit.
   2. Ketahanan cipher terhadap sebuah analisis sandi tidak otomatis menyebabkannya tahan terhadap analisis sandi lainnya. Ini terlihat dari ketahanan Rijndael terhadap ASL setelah 4 ronde, namun dengan Square attack, 4 Ronde Rijndael dapat dipecahkan dengan mudah. Sebaliknya, DES yang lemah menghadapi ASD dan ASL, ternyata memiliki ketahanan yang besar terhadap square attack.
   3. Pembuktian keamanan algoritma kriptografi lebih sulit dari pada pembuatan algoritmanya.

2.2 Kriptografi

Kriptografi (cryptography) berasal dari bahasa Yunani, terdiri dari dua suku kata yaitu kripto dan graphia. Kripto artinya menyembunyikan, sedangkan graphia artinya tulisan. Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi, seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data (Menezes, Oorschot and Vanstone, 1996). Tetapi tidak semua aspek keamanan informasi dapat diselesaikan dengan kriptografi. Kriptografi dapat pula diartikan sebagai ilmu atau seni untuk menjaga keamanan pesan.

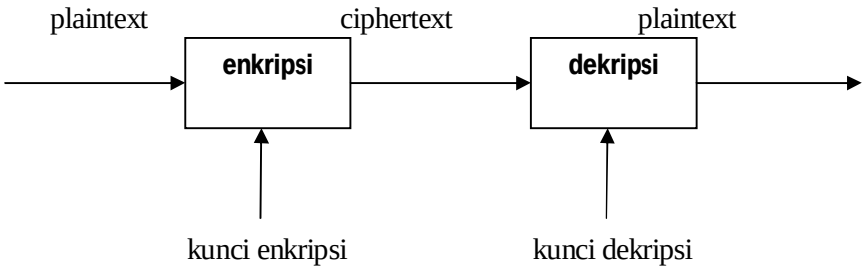
Menurut Munir (2006) ada empat tujuan mendasar dari ilmu kriptografi ini yang juga merupakan aspek keamanan informasi yaitu :

- 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.
- Integritas data, adalah berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem 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.
- 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.
- Non-repudiasi., atau nirpenyangkalan adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman/terciptanya suatu informasi oleh yang mengirimkan/membuat.

Berikut adalah istilah-istilah yang digunakan dalam bidang kriptografi (Munir, 2006) :

- **Plaintext** (M) adalah pesan yang hendak dikirimkan (berisi data asli).
- **Ciphertext** (C) adalah pesan ter-enkrip (tersandi) yang merupakan hasil enkripsi.
- **Enkripsi** (fungsi E) adalah proses pengubahan *plaintext* menjadi *ciphertext.*
- **Dekripsi** (fungsi D) adalah kebalikan dari enkripsi yakni mengubah *ciphertext* menjadi *plaintext,* sehingga berupa data awal/asli.
- **Kunci** adalah suatu bilangan yang dirahasiakan yang digunakan dalam proses enkripsi dan dekripsi.

Kriptografi itu sendiri terdiri dari dua proses utama yakni proses enkripsi dan proses dekripsi. Seperti yang telah dijelaskan di atas, proses enkripsi mengubah *plaintext* menjadi *ciphertext* (dengan menggunakan kunci tertentu) sehingga isi informasi pada pesan tersebut sukar dimengerti (Munir, 2006).



Gambar 1 Diagram proses enkripsi dan dekripsi

Peranan kunci sangatlah penting dalam proses enkripsi dan dekripsi (disamping pula algoritma yang digunakan) sehingga kerahasiaannya sangatlah penting, apabila kerahasiaannya terbongkar, maka isi dari pesan dapat diketahui.

Secara matematis (Munir, 2006), proses enkripsi merupakan pengoperasian fungsi E (enkripsi) menggunakan e (kunci enkripsi) pada M *(plaintext)* sehingga dihasilkan C *(ciphertext),* notasinya

$$\mathbf{E_e(M) - C}$$

Sedangkan untuk proses dekripsi (Munir, 2006), merupakan pengoperasian fungsi D (dekripsi) menggunakan d (kunci dekripsi) pada C *(ciphertext)* sehingga dihasilkan M *(plaintext),* notasinya :

$$\mathbf{D_d(C) = M}$$

Sehingga dari dua hubungan diatas berlaku :

$$\mathbf{D_d(E_e(M)) = M}$$

2.3 *Data Encryption Standard* (DES)

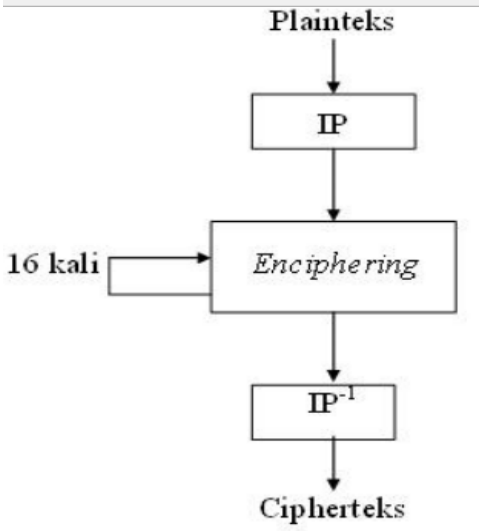
DES merupakan salah satu algoritma *Chiper Block* dengan ukuran 64 bit dan ukuran kunci 56 bit. DES dibuat dengan menggunakan algoritma terdahulu yang bernama Lucifer oleh IBM. Des pertama kali dipublikasikan di Federal Register pada 17 Maret 1975, setelah itu DES diadopsi sebagai algoritma standar yang digunakan oleh NBS (*National Bureau of Standards*) 15 januari 1977. Sejak saat itu DES digunakan untuk melindungi data agar tidak bisa dibaca oleh orang lain.

Namun demikian, DES juga mengundang banyak kontroversi dari para ahli di seluruh dunia. Salah satu kontroversi tersebut adalah *S-Box* yang digunakan pada DES. *S-Box*

merupakan bagian vital dari DES karena merupakan bagian yang paling sulit dipecahkan. Hal ini disebabkan karena *S-Box* merupakan satu – satunya bagian dari DES yang komputasinya tidak linear. Sementara itu, rancangan dari *S-Box* sendiri tidak diberitahukan kepada publik. Karena itulah, banyak yang curiga bahwa *S-Box* dirancang sedemikian rupa sehingga memberikan trapdoor kepada NSA agar NSA bisa membongkar semua ciphertext yang dienkripsi dengan DES kapan saja. Kontroversi yang kedua adalah jumlah bit pada kunci DES yang dianggap terlalu kecil, hanya 56 bit. Akibatnya DES rawan terhadap serangan brute force.

2.3.1 Panjang kunci dan ukuran Blok DES

Algoritma DES dikembangkan di IBM dibawah kepemimpinan W.L. Tuchman pada tahun 1972. Algoritma ini didasarkan pada algoritma Lucifer yang dibuat oleh Horst Feistel. Algoritma ini telah disetujui oleh National Bureau of Standard (NBS) setelah penilaian kekuatannya oleh National Security Agency (NSA) Amerika Serikat. DES termasuk ke dalam kriptografi kunci-simetri dan tergolong jenis cipher blok. DES beroperasi pada ukuran blok 64 bit. Panjang kunci ekternal = 64 bit (sesuai ukuran blok), tetapi hanya 56 bit yang dipakai (8 bit paritas tidak digunakan). Setiap blok (plainteks atau cipherteks) dienkripsi dalam 16 putaran. Setiap putaran menggunakan kunci internal berbeda. Kunci internal (56-bit) dibangkitkan dari kunci eksternal. Setiap blok mengalami permutasi awal (IP), 16 putaran enciphering, dan inversi permutasi awal (IP-1). (lihat Gambar 2)



Gambar 2 Skema Global Algoritma DES

1. Blok plainteks dipermutasi dengan matriks permutasi awal (initial permutation atau IP).
2. Hasil permutasi awal kemudian di-enciphering - sebanyak 16 kali (16 putaran). Setiap putaran menggunakan kunci internal yang berbeda.
3. Hasil enciphering kemudian dipermutasi dengan matriks permutasi balikan (invers initial permutation atau IP-1) menjadi blok cipherteks.

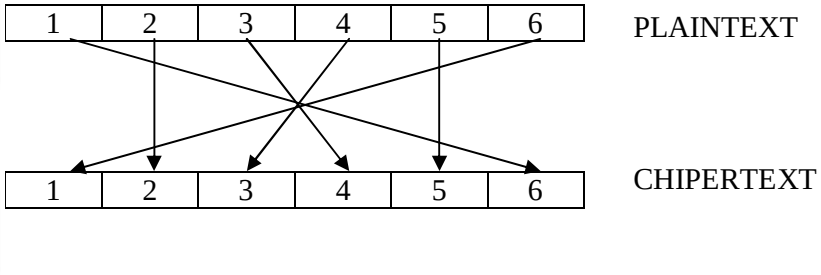
2.3.2 Teknik Dasar Kriptografi

a. Substitusi

Dalam kriptografi, sandi substitusi adalah jenis metode enkripsi dimana setiap satuan pada teks terang digantikan oleh teks tersandi dengan siste yang teratur. Metode penyandian substitusi telah dipakai dari zaman dulu (kriptografi klasik) hingga kini (kriptografi modern), langkah pertama adalah membuat suatu tabel substitusi. Tabel substitusi dapat dibuat sesuka hati, dengan catatan bahwa penerima pesan memiliki tabel yang sama untuk keperluan decrypt. Bila tabel substitusi dibuat secara acak, akan semakin suli pemecahan *chipertext* oleh orang yang tidak berhak. Metode ini dilakukan dengan mengganti setiap huruf dari teks asli dengan huruf lain sebagai huruf sandi yang telah didefinisikan sebelumnya olehalgoritma kunci. (www.mercubuana.ac.id)

b. Permutasi

Salah satu teknik yang penting pada enkripsi adalah permutasi atau sering juga disebut transposisi. Teknik ini memindahkan atau menotasikan karakter dengan aturan tertentu dengan prinsip yang berlawanan dengan teknik substitusi. Dalam teknik substitusi, karakter berada pada posisi yang tetap tapi identitasnya yang diacak. Pada teknik permutasi, identitas karakternya tetap, namun posisinya yang diacak. Sebelum dilakukan permutasi, umumnya plaintext terlebih dahulu dibagi menjadi blok – blok dengan panjang yang sama. *Plaintext* akan dibagi menjadi blok – blok yang terdiri dari 6 karakter, dengan aturan permutasi (www.mercubuana.ac.id), sebagai berikut :

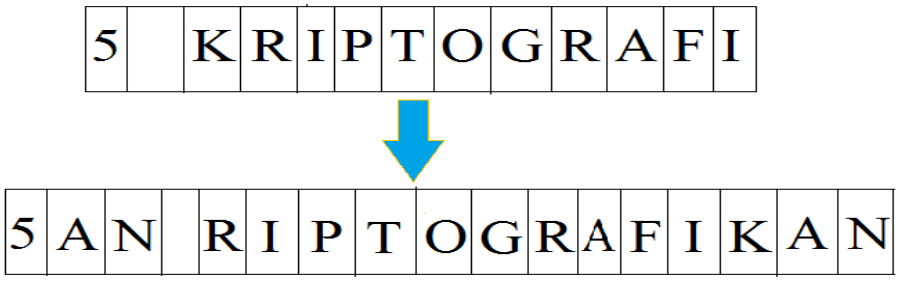


Gambar 4 Enkripsi dengan Permutasi

Dari gambar diatas dapat kita lihat proses permutasi yang menjadikan plainteks : 1 2 3 4 5 6 menjadi chiperteks : 6 2 4 3 5 1

c. Ekspansi

Suatu metode sederhana untuk mengacak pesan adalah dengan memperlebar pesan itu dengan aturan tertentu. Salah satu contoh penggunaan teknik ini adalah dengan meletakkan huruf konsonan atau bilangan ganjil yang menjadi awal dari suatu kata di akhir kata itu dan menambahkan akhiran "an". Bila suatu kata dimulai dengan huruf vokal atau bilangan genap, ditambahkan akhiran "i". proses enkripsi dengan cara ekspansi terhadap plaintext. (www.mercubuana.ac.id)



Gambar 5 Enkripsi dengan Ekspansi

Chipertext adalah "5AN RIPTOGRAFIKAN". Aturan ekspansi dapat dibuat lebih kompleks dengan menggabungkan dengan teknik lainnya.

2.4 Triple Data Encryption (3DES)

Menurut Hidayat, 2010 *Triple* DES adalah sebuah *chiper* blok yang dibentuk oleh DES dengan menggunakan tiga kali. *Triple* DES atau 3DES menggunakan DES 3 kali. Penggunaan tiga langkah ini penting untuk mencegah *meet – in – middle attack* sebagai mana pada *Double* DES.

Triple DES memiliki 2 varian yakni EEE dan EDE.

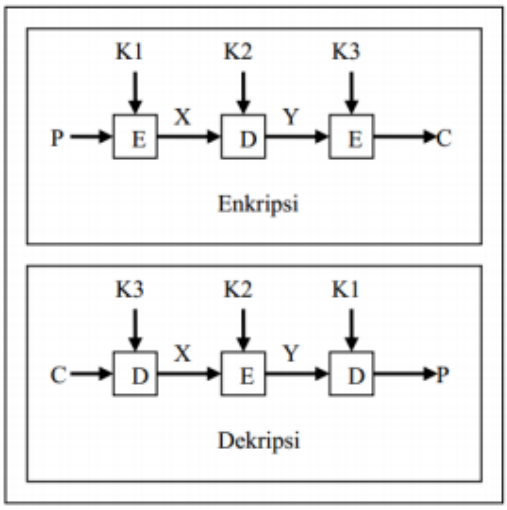
1. EEE adalah enkripsi menggunakan 3 kunci yaitu K1, K2, dan K3 dimana semua kunci berbeda sehingga panjang kunci yang didapat 128 bit, 3x dari panjang kunci DES.

   Bentuk sederhana dari 3DES EEE adalah :

   Enkripsi : $C = E_{k3}\ (E_{k2}(E_{k1}(P)))$

   Dekripsi : $P = D_{k1}\ (D_{k2}(D_{k3}(C)))$

2. EDE adalah enkripsi DES tunggal dengan kunci K1 = K2 = K3. Gambar dibawah ini memperlihatkan versi 3DES dengan 2 buah kunci. Penggunaan enkripsi tidak mempengaruhi keamanan algoritma.

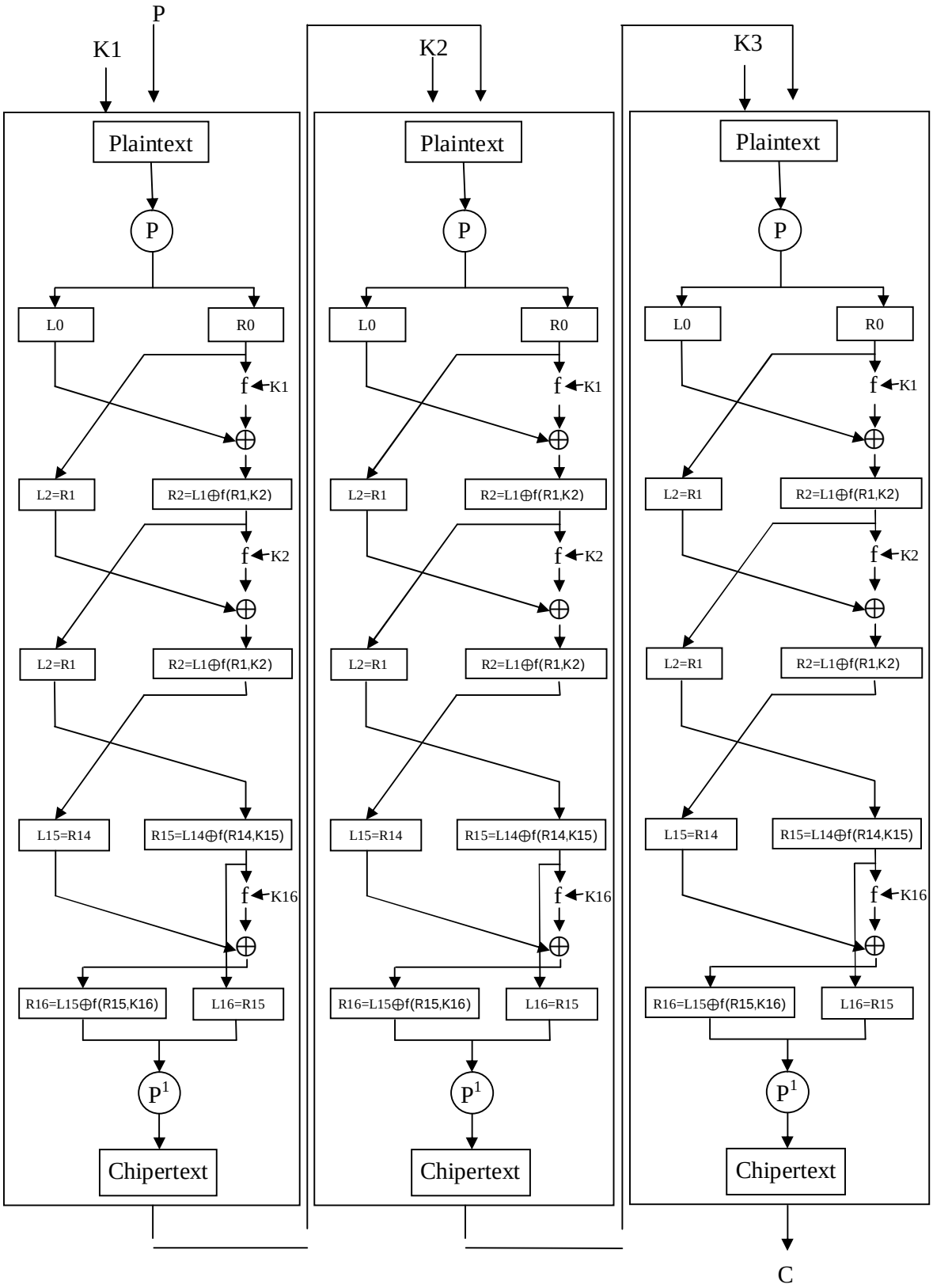


Gambar 6 Diagram enkripsi dan dekripsi 3DES (Hidayat,2010)

Perlu diingat bahwa DES bukanlah sebuah grup (dalam matematika), karena jika merupakan grup, pembangunan algoritma 3DES akan ekuivalen dengan operasi algoritma DES yang berarti tidak aman. Varian ini umum dikenal dengan metode penyandian EEE. Untuk menyederhanakan *interoperability* antara DES dan 3DES, maka pada langkah di tengah (pada proses enkripsi 3DES) diganti dengan dekripsi (mode EDE).

2.4.1 Algoritma *Triple* DES

Algoritma enkripsi atau dekripsi *Triple* DES seperti algoritma kriptografi lainnya yaitu memiliki algoritma umum. 3DES merupakan suatu algorirma pengembangan dari algoritma DES.



Gambar 7 Tahapan algoritma 3 DES (Hidayat, 2010)

Gambar diatas menjelaskan tentang tahapan algoritma 3DES. Pada dasarnya algoritma yang digunakan sama. Hanya pada 3DES dikembangkan dengan melakukan enkripsi dengan implementasi algoritma DES sebanyak tiga kali.

3DES memiliki 3 buah kunci yang berukuran 168-bit (tiga kali 56 bit dari DES). Pada algoritma 3DES dibagi menjadi tiga tahap, setiap tahapnya merupakan implementasi dari algoritma DES.

Tahap pertama, plainteks yang diinputkan dioperasikan dengan kunci eksternal pertama (K1) dan melakukan proses enkripsi dengan menggunakan algoritma DES, sehingga menghasilkan pra-chiperteks pertama. Tahap kedua, pra-chiperteks pertama yang dihasilkan pada tahap pertama, kemudian dioperasikan dengan kunci eksternal (K2) dan melakukan proses enkripsi atau proses dekripsi (tergantung cara pengenkripsian yang digunakan) dengan menggunakan algoritma DES. Sehingga menghasilkan pra-chiperteks kedua. Tahap terakhir, pra-chiperteks kedua yang dihasilkan pada tahap kedua, dioperasikan dengan kunci eksternal (K3) dan melakukan proses enkripsi dengan menggunakan algoritma DES, sehingga menghasilkan chiperteks (C).

2.4.1.1 Proses Enkripsi *Triple* DES

Proses enkripsi algoritma 3DES dapat dicapai dengan menggunakan tiga kunci :

Enkripsi : $C = E_{K3}(E_{K2}(E_{K1}(P)))$

Penjelasan :

Enkripsi pesan P mula – mula dengan kunci1 lalu hasilnya di enkripsi lagi dengan kunci K2, kemudian dienkripsi lagi dengan kunci K3 dan hasil enkripsi terakhir adalah chiperteks.

2.4.1.2 Proses Dekripsi *Triple* DES

Proses dekripsi algoritma 3DES dapat dicapai dengan menggunakan tiga kunci :

Dekripsi : $P = D_{K1}(D_{K2}(D_{K3}(C)))$

Penjelasan :

Mula – mula kunci K3 digunakan untuk mendekripsi C, lalu hasilnya didekripsi lagi dengan kunci K2, kemudian didekripsi lagi dengan kunci K1 dan hasil dekripsi terakhir adalah pesan semula (P).

2.4.2 Keamanan *Triple* DES

Secara umum 3DES dengan tiga kunci berbeda memiliki kunci berukuran 168-bit (3 kali kunci 56-bit dari DES), namun dengan metode *meet-in-the-middle* keamanan yang diberikan hanyalah 112-bit. Sebuah varian, Double 3DES, menggunakan kunci K1=K3, yang berarti mengecilkan ukuran ke 112-bit dan ukuran storage menjadi 128-bit.

Meet-in-middle attack adalah salah satu serangan terhadap kriptografi yang mencari nilai dalam setiap jarak dan domain dari komposisi dua fungsi sehingga *forward mapping* dari fungsi pertama sama dengan *image inverse* yang lainnya ke fungsi kedua. (Dian Intania, 2006)

2.4.3 Implementasi Kriptografi

Menurut Munir (2004), kriptografi banyak diguakan dalam kehidupan sehari – hari antara lain :

- Pengiriman data melalui saluran komunikasi
- Penyimpanan data di dalam disk penyimpanan
- *Automatic Teller Machine* (ATM)
- Telepon genggam (*Handphone)*
- Keamanan Jaringan Internet

Contoh aplikasi kriptografi pada file dokumen :

Plainteks (plain.doc) :

> Ketika saya berjalan – jalan di pantai, saya menemukan banyak sekali kepiting yang merangkak menuju laut. Mereka adalah anak-anak kepiting yang baru menetas dari dalam pasir. Naluri mereka mengatakan bahwa laut adalah tempat kehidupan mereka.

Chiperteks (chiper.doc) :

> 瑱 鎁蕫候垭ㅍ ·枯ㅁ禮慒幔캚縝띈ь厒仗階儕柒끔뻔쟞· 薹鮒鱦毲督 · ·걧轠〝十릗芑耐 ·灞鑣ᩚ ·먕· 滾庠ㄴ烝坽沫綄 ·ㅈ텃첁搔· · 骋 ·Ψ듦 醼鏨黃鴿· ·:敓 ·鉶◌ㅀ緗玸膃站魅菲庖잚ǂ 槤 · ·蔔%鱌ゾ誽佃쁰膚窒塔 頦棒督擸趬熜掅쇹∘ᄇ瘏褒惣幻揷ᛉ렛ᡐ탄擒醣⓱ ·ㅛ ·叽璘쥛쥢蛱· 晉钆

Hasil dekripsi terhadap berkas chiper.doc :

> Ketika saya berjalan – jalan di pantai, saya menemukan banyak sekali kepiting yang merangkak menuju laut. Mereka adalah anak-anak kepiting yang baru menetas dari dalam pasir. Naluri mereka mengatakan bahwa laut adalah tempat kehidupan mereka.

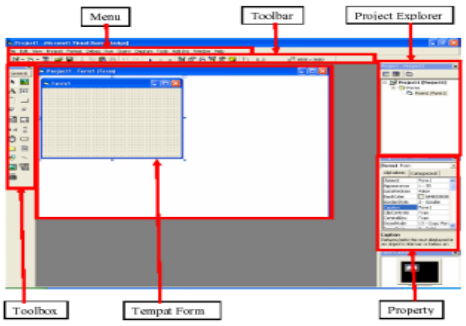
2.5 Visual Studio VB.NET

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (suite) yang dapat digunakan untuk melakukan pengembangan aplikasi. Baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya dalam bentuk aplikasi console, aplikasi Windows, ataupun aplikasi Web. Visual Studio mencakup kompiler, SDK, Integrated Development Environment (IDE), dan dokumentasi (umumnya berupa MSDN Library). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe.

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam native code (dalam bentuk bahasa mesin yang berjalan di Windows) ataupun managed code (dalam bentuk Microsoft Intermediate Language di atas .NET Framework). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi Silverlight, aplikasi Windows Mobile (yang berjalan di atas .NET Compact Framework).

1. Antar Muka Visual Studio

   Interface atau antar muka Visual Studio, berisi menu, toolbar, toolbox, form, project explorer dan property seperti terlihat pada Gambar berikut:



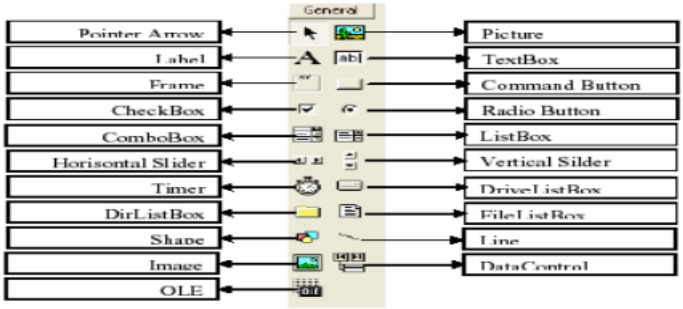
Gambar 8 Antarmuka *Visual Studio*

Pembuatan program aplikasi menggunakan Visual Studio dilakukan dengan membuat tampilan aplikasi pada form, kemudian diberi script program di dalam komponen-komponen yang diperlukan. Form disusun oleh komponen-komponen yang berada di [Toolbox], dan setiap komponen yang dipakai harus diatur propertinya lewat jendela [Property].

Menu pada dasarnya adalah operasional standar di dalam sistem operasi windows, seperti membuat form baru, membuat project baru, membuka project dan menyimpan project. Di samping itu terdapat fasilitas-fasilitas pemakaian Visual Studio pada menu. Untuk lebih jelasnya Visual Studio menyediakan bantuan yang sangat lengkap dan detail dalam MSDN (Microsoft Developer Network).

a. Toolbox

Toolbox berisi komponen-komponen yang bisa digunakan oleh suatu project aktif, artinya isi komponen dalam toolbox sangat tergantung pada jenis project yang dibangun. Komponen standar dalam toolbox dapat dilihat pada berikut ini.



Gambar 9 *Toolbox*

Toolbox Visual Studio dengan semua kontrol intrinsik. Jendela Toolbox merupakan jendela yang sangat penting. Dari jendela ini dapat mengambil komponen-komponen (object) yang akan ditanamkan pada form untuk membentuk user interface.

b. Variabel

Variabel adalah tempat dalam memori komputer yang diberi nama (sebagai pengenal) dan dialokasikan untuk menampung data. Sesuai data yang ditampung maka variabel harus mempunyai tipe data yang sesuai dengan isinya.

c. Operator

Operator digunakan untuk menghubungkan variabel dengan variabel lain untuk melakukan berbagai manipulasi dan pengolahan data.

2. Konsep Dasar Pemrograman Dalam Visual Studio

Konsep dasar pemrograman Visual Studio adalah pembuatan form dengan mengikuti aturan pemrograman Property, Metode dan Event. Hal ini berarti:

a. Property : Setiap komponen di dalam pemrograman Visual Studio dapat diatur propertinya sesuai dengan kebutuhan aplikasi.
b. Metode : Bahwa jalannya program dapat diatur sesuai aplikasi dengan menggunakan metode pemrograman yang diatur sebagai aksi dari setiap komponen. Metode merupakan tempat untuk mengekpresikan logika pemrograman dari pembuatan suatu program aplikasi.
c. Event : Setiap komponen dapat beraksi melalui event, seperti event click pada command button yang tertulis dalam layar script Command1_Click. [2].