BAB II
LANDASAN TEORI

2.1 Kriptografi

Kriptografi *(Crytography)* merupakan bidang pengetahuan yang menggunakan persamaan matematis untuk melakukan proses enkripsi *(encrypt)* maupun dekripsi *(decrypt)*. Teknik ini digunakan untuk menkonversi atau mengubah data kedalam bentuk kode-kode tertentu, dengan tujuan informasi yang disimpan maupun ditransmisikan melalui jaringan yang tidak aman (misalnya internet), tidak dapat dibaca oleh siapapun kecuali orang-orang yang berhak.

Kriptografi secara umum digunakan dalam bidang keamanan informasi, namun ada juga bidang pengetahuan lainnya yang digunakan untuk menganalisis dan memecahkan sistem keamanan komunikasi yaitu kriptanalisis *(Cryptanalysis)*, atau yang lebih dikenal sebagai penyerang *(attackers)*.

Kriptografi melakukan proses untuk mengkonversi teks biasa *(plaintext)* kedalam bentuk teks terenkripsi *(chipertext)* dan melakukan proses kebalikannya juga, chipertext dikonversikan kembali menjadi plaintext.

Untuk mewujudkan tujuan keamanan data informasi seperti kerahasiaan data *(confidentiality)*, keabsahan data *(authentication)*, integritas data *(integrity)* dapat menggunakan fungsi-fungsi seperti :

1. Fungsi Enkripsi

Ada tiga metode yang menggunakan metode kunci simetris *(conventional)*, menggunakan kunci asimetris *(public)*, menggunakan gabungan keduanya atau lebih dikenal dengan PGP *(pretty good privacy)*.

2. Aturan (protocol) dalam pembentukan kunci
3. Konsep tanda tangan digital

Suatu Cryptosistem terdiri dari suatu algoritma, yang terdiri dari plaintext, chipertext dan kunci *(key)*.

2.2 Kriptografi dan Sistem Informasi

Keamanan telah menjadi aspek yang sangat penting dari suatu sistem informasi. Sebuah sistem informasi umumnya hanya ditujukan bagi segolongan tertentu. Oleh karena itu sangat penting untuk mencegahnya jatuh ketangan pihak-pihak lain yang tidak berkepentingan. Untuk melaksanakan tujuan tersebutlah dirancang suatu sistem keamanan yang berfungsi melindungi sistem informasi.

Salah satu upaya pengamanan sistem informasi yang dapat dilakukan dalah kriptografi. Kriptografi sesungguhnya merupakan studi terhadap teknik matematis yang terkait dengan aspek keamanan suatu sistem informasi, antara lain seperti kerahasiaan, integritas data, otentikasi, dan ketiadaan penyangkalan.

Keempat aspek tersebut merupakan tujuan fundamental dari suatu sistem kriptografi:

1. Kerahasiaan *(confidentiality)*

Kerahasiaan adalah layanan yang digunakan untuk menjaga informasi dari setiap pihak yang tidak berwenang untuk mengaksesnya. Dengan demikan informasi hanya bisa diakses oleh pihak-pihak yang berhak saja.

2. Integritas Data *(data integrity)*

Integritas data merupakan layanan yang bertujuan untuk mencegah terjadinya pengubahan informasi oleh pihak-pihak yang tidak berwenang. Untuk meyakinkan integritas data ini harus dipastikan agar sistem informasi mampu mendeteksi terjadinya manipulasi data. Manipulasi yang dimaksud di sini meliputi penyisipan, penghapusan, maupun penggantian data.

3. Otentikasi *(authentication)*

Otentikasi merupakan layanan yang terkait dengan identifikasi terhadap pihak-pihak yang ingin mengakses sistem informasi *(entity authentication)* maupun keaslian data dari sistem informasi itu sendiri *(data origin authentication).*

4. Ketiadaan penyangkalan *(non-repudiation)*

Ketiadan penyangkalan adalah layanan yang berfungsi untuk mencegah terjadinya penyangkalan terhadap suatu aksi yang dilakukan oleh pelaku sistem informasi.

2.3 Mekanisme Kriptografi

Keamanan telah menjadi aspek yang sangat penting dari suatu sistem informasi. Sebuah informasi umumnya hanya ditujukan bagi segolongan tertentu.

Oleh karena itu sangat penting untuk mencegahnya jatuh kepada pihak-pihak lain yang tidak berkepentingan.

Suatu sistem kriptografi *(kriptosistem)* bekerja dengan cara menyandikan suatu pesan menjadi suatu kode rahasia yang dimengerti oleh pelaku sistem informasi saja. Pada dasarnya mekanisme kerja semacam ini telah dikenal sejak zaman dahulu, bangsa Mesir kuno sekitar 4000 tahun yang lalu bahkan telah mempraktekannya dengan cara yang sangat primitif.

Dalam era teknologi informasi sekarang ini, mekanisme yang sama masih digunakan tetapi tentunya implementasi sistemnya berbeda. Sebelum membahas lebih jauh mekanisme kriptografi modern, berikut ini diberikan beberapa istilah yang umum digunakan dalam pembahasan kriptografi.

1. Plaintext

Plaintext *(message)* merupakan pesan asli yang ingin dikirimkan dan dijaga keamanannya. Pesan ini tidak lain dari informasi tersebut.

2. Ciphertext

Chipertext merupakan pesan yang telah dikodekan (disandikan) sehinga siap untuk dikirimkan.

3. Cipher

Cipher merupakan algoritma matematis yang digunakan untuk proses penyandian plaintext menjadi chipertext.

4. Key

Nilai yang dipakai pada algoritma kriptografi untuk mengacak plainteks sehingga menghasilkan cipherteks.

5. Enkripsi

Enkripsi merupakan proses mengubah plainteks menjadi suatu cipherteks dengan menggunakan algoritma tertentu.

6. Dekripsi

Dekripsi merupakan proses mengubah kembali cipherteks menjadi plainteks sehingga dapat dimengerti kembali oleh pengguna.

2.4 Enkripsi dan Dekripsi

Pada dasarnya metode kriptografi terdiri dari dua proses yaitu enkripsi dan dekripsi, kedua proses tersebut akan dijelaskan berikut ini.

2.4.1 Enkripsi

Enkripsi merupakan proses awal kriptografi dengan memasukan input berrupa plainteks kedalam algoritma tertentu beserta suatu kunci untuk memperoleh output beberapa cipherteks yang tidak dapat langsung dimengerti oleh orang lain, proses enkripsi dapat diformulakan sebagai berikut:

$C = E_k (P)$

Dimana:

P = Plaintext

E_k = Proses enkripsi dengan menggunakan algoritma tertentu dan kunci K

C = Ciphertext

Proses enkripsi biasanya melibatkan perubahan bentuk data menjadi suatu bentuk yang sulit untuk dibaca dan diartikan tanpa sebuah kunci. Enkripsi

digunakan untuk memastikan bahwa data atau pesan yang disimpan atau dikirimkan melalui suatu jaringan terlindungi dari orang-orang yang tidak berhak untuk membacanya.

2.4.2 Dekripsi

Dekripsi merupakan kebalikan dari proses enkripsi, yaitu proses mengubah bentuk yang sudah diacak pada proses enkripsi kembali kebentuk semula. Pada dekripsi, ciphertext akan diinput dan akan diolah dengan suatu algoritma dan suatu kunci untuk memperoleh output berupa plaintext.

Proses dekripsi dapat diformulakan sebagai berikut:

$P = D_k (C)$

Dimana:

P = Plaintext

DK = Proses dekripsi dengan menggunakan algoritma tertentu dan kunci K

C = Ciphertext

Enkripsi dan dekripsi biasanya membutuhkan penggunaan beberapa informasi rahasia yang berhubungan dengan sebuah kunci. Untuk beberapa mekanisme enkripsi kunci yang sama digunakan untuk enkripsi dan dekripsi berbeda.

2.5 Algoritma Kriptografi

Berdasarkan kunci yang dipakai, algoritma kriptografi dapat dibedakan atas dua golongan, yaitu kunci simetris dan kunci asimetris yang akan dijelaskan lebih lanjut di bawah ini:

2.5.1 Kunci Simetris

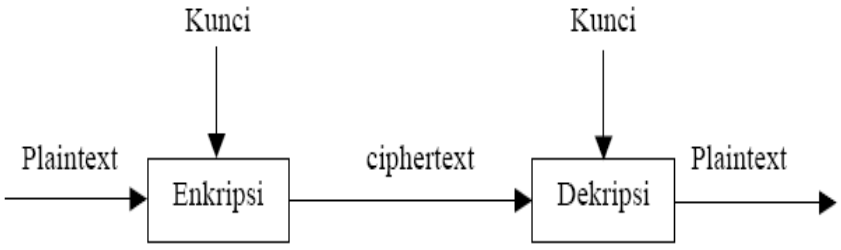
Kunci Simetris adalah jenis kriptografi yang paling umum digunakan. Kunci untuk membuat pesan yang di sandikan sama dengan kunci untuk membuka pesan yang disandikan itu. Jadi pembuat pesan dan penerimanya harus memiliki kunci yang sama persis. Siapapun yang memiliki kunci tersebut termasuk pihak-pihak yang tidak diinginkan dapat membuat dan membongkar rahasia ciphertext.

Algoritma simetri disebut juga sebagai algoritma konvensional yaitu algoritma yang menggunakan kunci enkripsi yang sama dengan kunci dekripsinya. Disebut konvensional karena algoritma yang biasa digunakan sejak berabad-abad yang lalu adalah algoritma jenis ini. Algoritma simetri sering juga disebut algoritma kunci rahasia, algoritma kunci tunggal, atau algoritma satu kunci, dan mengharuskan pengirim dan penerima menyetujui suatu kunci tertentu sebelum mereka dapat berkomunikasi dengan aman.

Keamanan algoritma simetri tergantung pada kunci, membocorkan kunci nerarti bahwa orang lain dapat mengenkrip dan mendekrip pesan. Agar komunikasi tetap aman, kunci harus tetap dirahasiakan. Yang termasuk algoritma kunci simetri adalah OTP, DES, RC2, RC4, RC5, RC6, IDEA, Twofish, Magenta,

FEAL, SAFER, LOKI, CAST, Rijndael (AES), Blowfish, GOST, A5, Kasumi dan lain-lain.

Proses enkripsi-dekripsi algoritma kriptografi kunci simetris dapat dilihat pada gambar dibawah ini:



Gambar 2.1. Proses Enkripsi-dekripsi kunci simetris

Algoritma kriptografi simeteris dibagi menajdi 2 kategori yaitu algoritma aliran (Stream Ciphers) dan algoritma blok (Block Ciphers). Berikut adalah penjelassannya :

1. Cipher Aliran (*Stream Cipher*)

Algoritma kriptografi ini beroperasi pada plainteks atau cipherteks dalam bentuk bit tunggal, yang dalam hal ini rangkaian bit dienkripsikan atau didekripsikan bit per bit.

2. Cipher blok (*Block Cipher*)

Algoritma kriptografi ini beroperasi pada plainteks atau cipherteks dalam bentuk blok bit, yang dalam hal ini rangkaian bit dibagi menjadi blok-blok bit yang panjangnya sudah ditentukan sebelumnya.

Pada penulisan skripsi ini penulis akan memfokuskan pembahasan mengenai cipher blok, karena algoritma LOKI97 merupakan algoritma yang menerapkan cipher blok.

Mode operasi yang lazim diterapkan pada sistem blok cipher ada empat buah, semuanya akan dibahas di bawah ini :

1. *Electronic Code Book* (ECB)

Pada mode ini setiap blok plainteks P_i dienkripsi secara individual dan independen menjadi blok cipherteks C_i. Secara matematis enkripsi dengan mode ECB dinyatakan sebagai

$$C_i = E_i (P_i)$$

Dan dekripsi sebagai

$$P_i = D_i (C_i)$$

Yang dalam hal ini, P_i dan C_i masing-masing blok plainteks dan cipherteks ke-i. pada mode ECB, blok plainteks yang sama selalu dienkripsi menjadi blok cipherteks yang sama.

2. *Chiper Block Chaining* (CBC)

Mode ini menerapkan mekanisme umpan balik (feedback) pada sebuah blok, yang dalam hal ini hasil dari enkripsi sebelumnya diumpanbalikan ke dalam enkripsi blok yang *current*.

Secara matematis, enkripsi dengan mode CBC dinyatakan sebagai

$$C_i = E_i (P_i \oplus C_{i-1})$$

Dan dekripsi sebagai

$$P_i = D_i (C_i) \oplus C_{i-1}$$

Pada mode CBC, blok plainteks yang sama menghasilkan blok cipherteks yang berbeda hanya jika blok-blok plainteks sebelumnya berbeda.

3. *Cipher Feedback* (CFB)

Pada mode CFB, data dienkripsikan dalam unit yang lebih kecil daripada ukuran blok. Sehingga enkripsi dapat dilakukan meskipun data yang diterima belum lengkap. Unit yang dienkripsikan dapat berupa bit per bit (seperti cipher aliran), 2 bit, 3bit dan seterusnya. Secara umum CFB n-bit mengenkripsi plainteks sebanyak n bit setiap kalinya, yang mana $n \leq m$ (m = ukuran blok).

Mode CFB membutuhkan sebuah antrian (*queue*) yang berukuran sama dengan ukuran blok masukan. Secara formal, mode CFB n-bit dapat dinyatakan sebagai:

Proses enkripsi :

$$C_i = P_i \oplus MSB_m\,(E_k\,(X_i))$$

$$X_{i+1} = LSB_{m-n}\,(X_i) \,||\, C_i$$

Proses dekripsi :

$$P_i = C_i \oplus MSB_m(D_k(X_i))$$

$$X_{i+1} = LSB_{m-n}(X_i) \,||\, C_i$$

yang dalam hal ini:

Xi = isi antrian dengan *Xi* adalah *IV*

E = fungsi enkripsi dengan algoritma *cipher* blok

D = fungsi dekripsi dengan algoritma *cipher* blok

K = kunci

m = panjang blok enkripsi/dekripsi

n = panjang unit enkripsi/dekripsi

|| = operator penyambungan (*concatenation*)

MSB = Most Significant Byte

LSB = Least Significant Byte

Ada satu kesalahan fatal pada mode CFB, yaitu adanya perambatan kesalahan. Kesalahan 1 bit pada plainteks akan merambat pada blok-blok cipherteks yang berkoresponden dan blok-blok cipherteks selanjutnya pada proses enkripsi. Hal yang kebalikan juga terjadi pada proses dekripsi.

4. *Output Feedback* (OFB)

Mode OFB mirip dengan mode CFB, kecuali n-bit dari hasil enkripsi terhadap antrian disalin terhadap elemen posisi paling kanan di antrian.

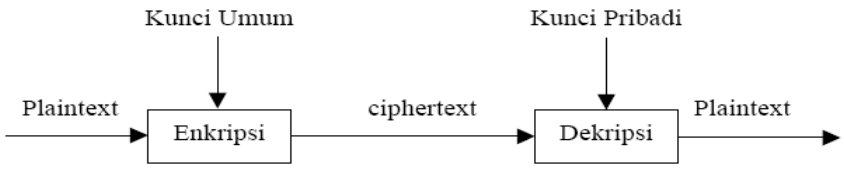
2.5.2 Kunci Asimetris

Kunci asimetris adalah pasangan kunci kriptografi yang salah satunya digunakan untuk proses enkripsi dan yang satu lagi untuk dekripsi. Semua orang yang mendapatkan kunci publik dapat menggunakannya untuk mengenkripsikan suatu pesan,data ataupun informasi, sedangkan hanya satu orang saja yang memiliki rahasia tertentu dalam hal ini kunci privat untuk melakukan pembongkaran terhadap sandi yang dikirim untuknya.

Algoritma asimetrik didesain sedemikian rupa sehingga kunci yang digunakan untuk enkripsi berbeda dengan kunci yang digunakan untuk dekripsi. Lebih jauh lagi, kunci dekripsi tidak dapat (sedikitnya dalam waktu yang dapat diterima) dihitung dari kunci enkripsi. Algoritma disebut kunci public karena

kunci enkripsi dapat dibuat public yang berarti semua orang boleh mengetahuinya. Sembarang orang dapat menggunakan kunci enkripsi tersebut untuk mengenkrip pesan, namun hanya orang tertentu (calon penerima pesan dan sekaligus pemilik kunci dekripsi yang merupakan pasangan kunci public) yang dapat melakukan dekripsi terhadap pesan tersebut. Dalam sistem ini kunci enkripsi sering disebut kunci public, sementara kunci dekripsi sering disebut kunci privat. Kunci privat kadang-kadang disebut kunci rahasia. Jadi anda harus berhati-hati dalam membacanya. Yang termasuk algoritma asimetri adalah ECC, LUC, RSA, El Gamal, dan DH.

Proses enkripsi-dekripsi algoritma kunci asimetris dapat dilihat pada gambar dibawah ini:



Gambar 2.2. Proses enkripsi-dekripsi kunci asimetris

Pada algoritma public key ini, semua orang dapat mengenkripsi data dengan memakai public key penerima yang telah diketahui secara umum. Akan tetapi data yang telah terenkripsi tersebut hanya dapat didekripsi dengan menggunakan private key yang hanya diketahui oleh penerima.

2.6 Algoritma LOKI97

LOKI97 merupakan sebuah algoritma blok cipher menggunakan kunci privat yang mengoperasikan data sebesar 128 bit dengan kunci sebesar 256 bit. LOKI97 merupakan evolusi dari LOKI89 dan LOKI91 dengan memperkuat penjadwalan kunci dan memperbesar ukuran kunci yang digunakan. LOKI97 diajukan sebagai kandidat pengganti DES (Data Encryption Standard). Struktur dan desain LOKI97 secaara keseluruhan dibuat oleh Dr. Lawrie Browns, sedangkan desain dari S-Box yang digunakan dibuat oleh Prof. Pieprzyk.

2.6.1 Spesifikasi LOKI97

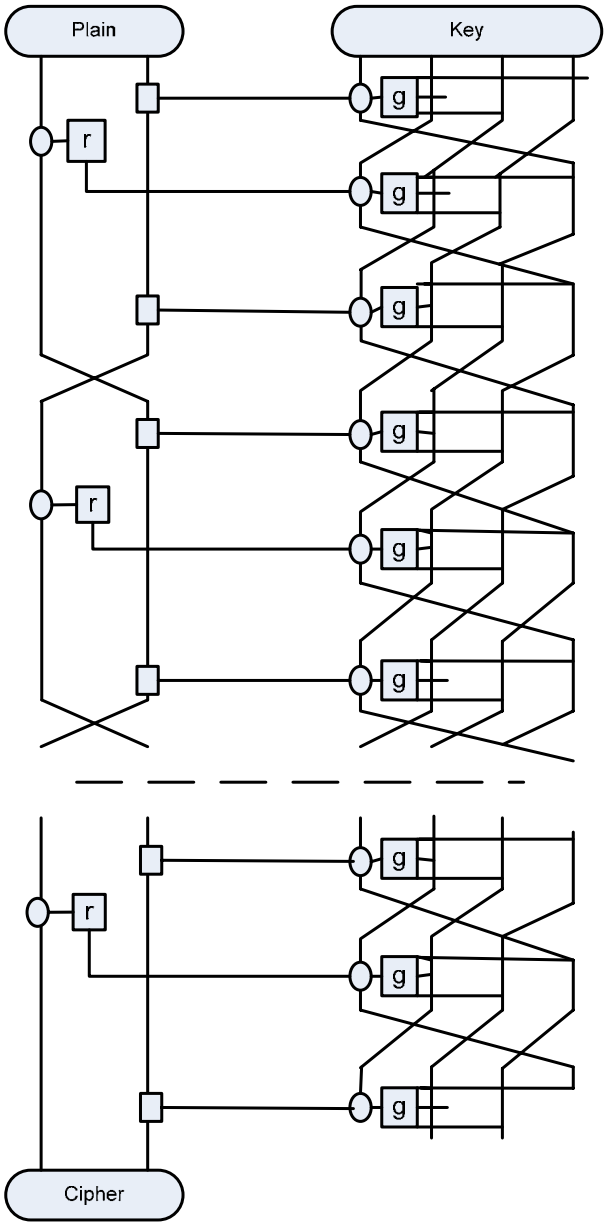
LOKI97 melakukan proses enkripsi dan dekripsi data 128 bit menggunakan kunci sepanjang128, 192, atau 256 bit. Karena LOKI97 menerima masukan data sepanjang 128 bit dan dikarenakan LOKI97 merupakan algoritma block cipher maka LOKI97 akan mengjasilkan data keluaran sepanjang 128 bit pula.

Proses pertama yang dilakukan dalam melakukan enkripsi adalah menjadi blok masukan menjadi dua buah blok yang sama panjang yaitu L dan R. Setelah data dibagi menjadi dua buah bagian yang sama besar, dilakukan proses putaran sebanyak 16 kali dengan menggunakan jaringan feistel. Proses yang dilakukan dalam setiap putaran adalah sebagai berikut.

$R = L_{i-1}$ XOR $f(R_{i-1} + SK_{3i-2}, SK_{3i-1})$

$L = R_{i-1} + SK_{3i-2} + SK_{3i}$

Setiap putaran menggunakan operasi XOR dan + (modulo 2^{64}) terhadap data 64 bit data masukan bersamaan dengan hasil fungsi f yang merupakan hasil fungsi nonlinier yang rumit sehingga memaksimalkan *avalanche* antara bit-bit masukan.



Gambar 2.3. Proses enkripsi LOKI97

Sedangkan untuk proses dekripsi, proses putaran hanya dibalik. Dengan demikian perhitungannya adalah sebagai berikut.

$L_{i-1} = R_i$ XOR $f(L_i - SK_{3i}, SK_{3i-1})$

$R_{i-1} = L_i - SK_{3i} - SK_{3i-2}$

2.7 Teori Angka

Kriptografi tidak terlepas dari ilmu matematika. Untuk itu di bawah ini dijelaskan tentang berbagai dasar matematika yang digunakan dalam kriptografi.

2.7.1 Aritmatika Modular

Aritmatika modular adalah suatu perhitungan matematika yang berhubungan dengan hasil sisa pembagian. Dalam aritmatika modular suatu bilangan bulat positif **n** dan bilangan bulat **a** jika dilakukan pembagian **a** dan **n**, maka akan dihasilkan suatu bilangan **q** dan sisa pembagian **r**. dalam notasi matematika hal tersebut dapat dinyatakan sebagai berikut:

$a = qn + r ; 0 \leq r \leq n ; q = [a/n]$

Dimana [x] adalah bilangan bulat terbesar yang lebih kecil atau sama dengan x.

2.7.2 Invers Modulo

Invers modulo adalah sebuah proses bilangan yang lebih sulit daripada invers biasa, karena invers modulo menggunakan dua variable yang berbeda, misalnya:

4 * x = 1 mod 7 ;

Contoh diatas dapat digambarkan sebagai 4x = 7k +1 ; x dan k bilangan integer. Notasi invers modulo dapat dituliskan dengan: $a^{-1} = x \pmod{n}$

2.7.3 Operasi Logika XOR

XOR (Exlusive OR) merupakan logika operasi yang sering ditemukan pada bahasa pemrograman. XOR juga merupakan operator bitwise pada pengoperasian biner. Ada enam operator bitwise yang terdapat pada bahasa pemrograman C, sebagai berikut :

Tabel 2.1 Operator bitwise

Operator	Simbol
AND	&
Inclusive OR	|
Exclusive OR (XOR)	^
Pergeseran bit ke kanan	>>
Pergeseran bit ke kiri	<<
Complement	~

Operator XOR akan menghasilkan nilai 1, apabila salah satu dari bit pertama atau bit kedua bernilai 1, dan akan menghasilkan nilai 0 apabila kedua bit bernilai 0 atau 1. Dalam ilmu matematika notasinya ditulis $\oplus$, dan berikut adalah ilustrasinya :

$0 \oplus 0 = 0$

$0 \oplus 1 = 1$

$1 \oplus 0 = 1$

$1 \oplus 1 = 0$

Sebagai contoh:

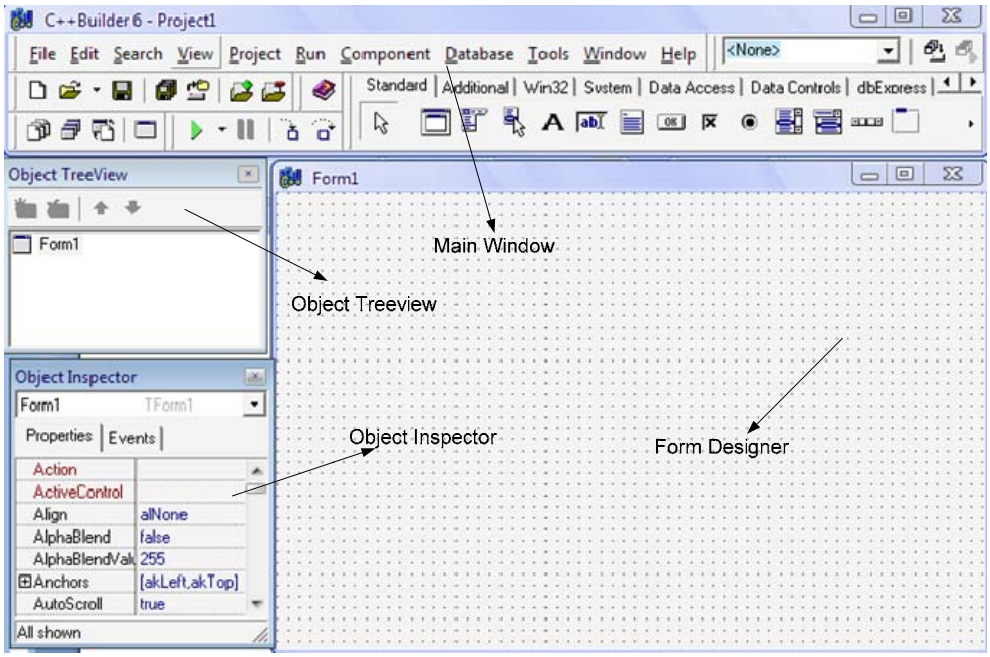
10011110 $\oplus$ 11100101 = 01111011

Operator XOR banyak digunakan pada algoritma kriptografi karena sederhana untuk diterapkan, dan juga ketika kita ingin menginverskannya, hal itu mudah untuk dilakukan.

2.8 Borland C++ Builder 6

Borland C++ builder adalah suatu alat pengembang aplikasi (*development tool*) berbasis Microsoft Windows yang menerapkan konsep visualisasi. Dengan adanya dukungan visualisasi ini C++ Builder menjadi mudah digunakan untuk membuat aplikasi-aplikasi secara cepat. Dengan C++ Builder, kita dapat melakukan desain, testing, debugging, maupun proses deployment aplikasi secara mudah.

Bahasa dasar yang digunakan dalam pemrograman C++ Builder adalah bahasa C++, yang telah terkenal dengan keampuhannya dalam pembuatan program. Linkungan yang terdapat pada C++ Builder menawarkan kemudahan dan kenyamanan dalam proses pengembangan aplikasi. Lingkungan semacam ini sering disebut dengan IDE (*Integrated Development Environment*). Secara umum IDE C++ Builder dibagi menjadi lima bagian besar, Yaitu *Main Window, Object Treeview*, *Form Designer*, *Object Inspector* dan *Code Editor*.



Gambar 2.4 Tampilan utama Borland C++ Builder 6