KEAMANAN *DATABASE* MENGGUNAKAN METODE ENKRIPSI

Suyanto
Dosen Universitas Bina Darma
Jalan Jenderal Ahmad Yani No.12, Palembang
Pos-el: suyanto@mail.binadarma.ac.id

Abstract:*Growing data processing technology, this has resulted in the implementation of computerized systems in all fields. All parts of the organization or company has implemented various applications to meet their needs. Applications are applied separately raises its own problems which need the same data. So that data exchange becomes an alternative solution. Data security requirements on the database exchange process are become new problems in the process. To ensure data security on the exchange process, the encryption method is the best way. With the implementation of encryption on data that is exchanged, then the data is guaranteed safe and can only be read by those who need it. Applications that use different databases require a standard format so that all applications can read it. This format is XML. With XML, the database (tables) can be read by other applications though with different platforms.*

Keywords:*Encryption*, *Decryption*, *Security*, *Database*, *Cipher*, *XML.*

Abstrak:*Pertumbuhan teknologi pengolahan data, telah menuntut penerapan sistem komputerisasi di segala bidang. Seluruh bagian dari organisasi atau perusahaan telah menerapkan berbagai aplikasi untuk memenuhi kebutuhan mereka. Aplikasi yang diterapkan secara terpisah menimbulkan masalah sendiri untuk aplikasi yang memerlukan data yang sama. Sehingga pertukaran data menjadi solusi alternatif. Persyaratan keamanan data pada proses pertukaran database menjadi masalah baru dalam proses tersebut. Untuk menjamin keamanan data pada proses pertukaran, metode enkripsi adalah cara terbaik. Dengan penerapan enkripsi pada data yang dipertukarkan, maka data dijamin aman dan hanya dapat dibaca oleh mereka yang membutuhkannya. Aplikasi yang menggunakan database berbeda membutuhkan format standar sehingga semua aplikasi dapat membacanya. Format ini adalah XML. Dengan XML, database (tabel) dapat dibaca oleh aplikasi lain meskipun dengan platform yang berbeda.*

Kata kunci*:* Enkripsi, Dekripsi, Keamanan*,* Basis Data, Tersandikan*, XML*

1. PENDAHULUAN

Pemanfaatan aplikasi komputer dalam suatu perusahaan atau organisasi dewasa ini sudah merupakan suatu kebutuhan. Hampir di semua bidang pekerjaan menggunakan alat bantu berupa aplikasi komputer. Dalam satu perusahaan misalnya, aplikasi komputer sering dibuat sendiri oleh team komputer dari perusahaan tersebut.

Dengan berkembangnya pengolahan data, tidak jarang satu aplikasi menggunakan *database* yang sama untuk aplikasi yang lainnya. Jalan yang bisa ditempuh adalah dengan cara mengkopi *database* yang dibutuhkan tersebut dari satu aplikasi ke aplikasi yang lain. Hal ini merupakan jalan pintas, agar aplikasi yang membutuhkan data tadi bisa dioperasikan. Begitu seterusnya sehingga kegiatan kopi *database* merupakan suatu kebutuhan dari aplikasi tersebut.

Melihat kondisi diatas, maka keamanan data menjadi suatu hal yang sangat penting. Karena bukan hal yang mustahil bahwa data yang dipertukarkan akan disalahgunakan oleh pihak-pihak yang tidak bertanggung jawab untuk memenuhi kebutuhan pribadi mereka. Untuk itu

perlu teknik khusus untuk mengamankan *database* yang dipertukarkan tersebut.

Salah satu mekanisme untuk meningkatkan keamanan data dalam *database* adalah dengan menggunakan teknologi enkripsi. Data-data yang disimpan dalam *database* diubah sedemikian rupa sehingga tidak mudah dibaca. Jadi enkripsi adalah proses yang dilakukan untuk mengamankan sebuah data (yang disebut *plaintext*) menjadi data yang tersembunyi (disebut *ciphertext*). *Ciphertext* adalah data yang sudah tidak dapat dibaca dengan mudah. Menurut ISO 7498-2, terminologi yang lebih tepat digunakan adalah "*encipher*". Proses sebaliknya, untuk mengubah *ciphertext* menjadi *plaintext*, disebut dekripsi (*decryption*). Menurut ISO 7498-2, terminologi yang lebih tepat untuk proses ini adalah "*decipher*" (Rahardjo:2002).

Pengetahuan yang mempelajari tentang enkripsi adalah kriptografi. Yang dimaksud dengan kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data. Sedangkan menurut Prasetyo (2006:372) Kriptografi adalah suatu ilmu pengetahuan (atau lebih tepatnya suatu seni) yang mengenkripsi informasi sehingga sepenuhnya terlihat berbeda dari aslinya. Selain itu Munir (2006) menjelaskan Kriptografi adalah ilmu dan seni untuk menjaga keamaan pesan atau data.

Berdasarkan cara memproses teks (*plaintext*), *cipher* dapat dikategorikan menjadi dua jenis: *block cipher* dan *stream cipher* (Rahardjo:2002). *Block cipher* bekerja dengan memproses data secara blok, dimana beberapa karakter/data digabungkan menjadi satu blok. Setiap proses satu blok menghasilkan keluaran satu blok juga. Sementara itu *stream cipher* bekerja memproses masukan (karakter atau data) secara terus menerus dan menghasilkan data pada saat yang bersamaan.

Terdapat beberapa cara untuk melakukan enkripsi (Suryana dkk, 2007:3) yaitu: Enkripsi Simetris (*symmetricencryption*) dan Asimetris(*asymmetricencryption*).

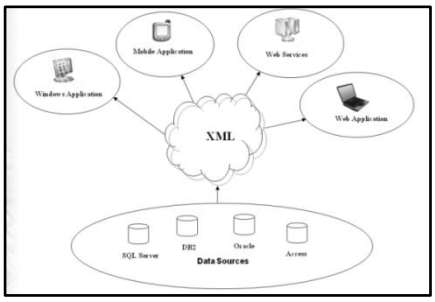
AlgoritmaEnkripsi simetris (*symmetricencryption*)ini menggunakan sebuah kunci rahasia yang sama (*private key*) untuk melakukan proses enkripsi dan dekripsinya. Algoritma ini termasuk dalam algoritma kriptografi klasik yang terdiri dari: s*ubstitution cipher* dan *transposition cipher*.

Cara kerja dari algoritma s*ubstitution cipher* ini adalah dengan menggantikan setiap karakter dari *plaintext* dengan karakter lain. Algoritma ini pertama kali digunakan oleh julius caesar, dan disebut juga sebagai *shift cipher*, yaitu dengan cara menggeser urutan abjadnya. Substitusi ini kadang dikenal dengan c3 (untuk caesar menggeser 3 tempat). Secara umum sistem *cipher* caesar dapat ditulis Zi = cn(pi). Dimana Zi adalah karakter-karekter *ciphertext*, Cn adalah transformasi substitusi alfabetik, n adalah jumlah huruf yang digeser, dan pi adalah karakter-karakter plaintext (Kelompok 124 IKI-83408 MTI UI, 2005).

Guna mendukung bermacam-macam format *database*, maka hasil enkripsi kemudian digenerate menjadi format XML.

Dalam buku NIIT (2009) yang berjudul "*Introduction to Web Content Development"* menjelaskan tentang XML sebagai berikut :

a) *XML is a text-based markup language that enables storage of data in a structured format.*
b) *XML is a cross-platform, hardware and software independent markup language that enables structured data transfer between heterogeneous systems.*
c) *XML is used as a common data interchange format in a number of applications.*



Gambar 1. Penggunaan Dokumen XML

2. METODE PENELITIANDAN PERANCANGAN

2.1 Metode Penelitian

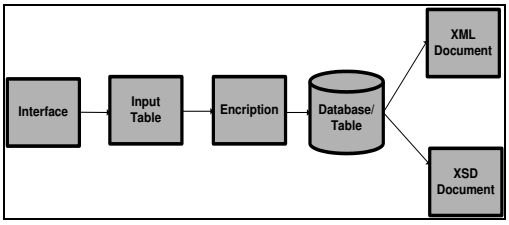
Metodeyang digunakan pada penelitian ini adalah metode penelitian tindakan *(action research).* Metode inimerupakan metode penelitian yang bertujuan mengembangkan keterampilan-keterampilan baru atau cara pendekatan baru dan untuk memecahkan masalah dengan penerapan langsung di dunia kerja atau dunia nyata lainnya.

Untuk mengumpulkan data-data dan bahan penelitian maka penulis menggunakan Studi Literatur. Studi literatur adalah mencari referensi teori yang relefan dengan kasus atau permasalahan yang ditemukan. Referensi ini dapat dicari dari buku, jurnal, artikel, laporan penelitian, dan situs-situs di internet. Output dari studi literatur ini adalah terkoleksinya referensi yang relefan dengan perumusan masalah. Tujuannya adalah untuk memperkuat permasalahan serta sebagai dasar teori dalam melakukan studi dan juga menjadi dasar untuk melakukan desain dari permasalahan yang diteliti (http://digilib.its.ac.id/).

Metode pengembangan perangkat lunak yang digunakan dalam penelitian ini adalah *Prototyping*. *Prototyping paradigma* dimulai dengan pengumpulan kebutuhan. Pengembang dan pelanggan bertemu dan mendefinisikan obyektif keseluruhan dari *software*, mengidentifikasi segala kebutuhan yang diketahui, dan area garis besar dimana definisi lebih jauh merupakan keharusan kemudian dilakukan "perancangan kilat.

Perancangankilat berfokus pada penyajian dari aspek –aspek software tersebut yang akan nampak bagi pelanggan atau pemakai (contohnya pendekatan input dan format output). Perancangan kilat membawa kepada konstruksi sebuah prototipe. Prototipe tersebut dievaluasi oleh pelanggan/pemakai dan dipakai untuk menyaring kebutuhan pengembangan software. Iterasi terjadi pada saat prototipe disetel untuk memenuhi kebutuhan pelanggan, dan pada saat yang sama memungkinkan pengembang untuk secara lebih baik memahami apa yang harus dilakukannya.

Kerangka pikir pengembangan perangkat lunak yang akan dibangun dapat digambarkan sebagai berikut :



Gambar 2. Kerangka Pikir Pengembangan

Dari kerangka pikir diatas maka dihasilkan rancangan interface, rancangan algoritma enkripsi dan dekripsi sebagai berikut :

2.2 Rancangan Sistem

Perancangan sistem secara detailnya mengacu pada tahapan pengembangan sistem dengan *Prototype Model*. Dalam *Prototype Model* mencakup tiga tahapan yang akan dilalui dalam pengembangan perangkat lunak, yaitu mendefinisikan tujuan dan kebutuhan pemakai, melakukan perancangan secara cepat dan melakukan uji coba.

2.2.1 Mendefinisikan Tujuan Dan Mengidentifikasi Kebutuhan Pemakai

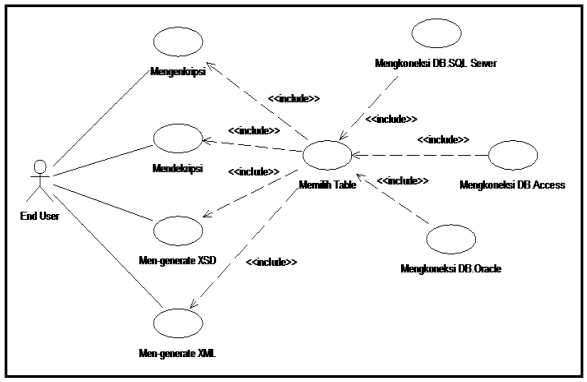
Tujuan dari pengembangan sistem ini adalah membangun sebuah aplikasi yang akan dijadikan alat bantu (*tools*) dalam proses pemindahan data dari satu *database* ke *database* yang lain. Aplikasi ini berperan sebagai alat bantu dalam melakukan enkripsi data. Enkripsi ini diperlukan karena pada saat pemindahan data masih dilakukan secara offline, artinya data dipindahkan dari satu *database* ke *database* lain dengan media tertentu seperti : CD, Flashdisk, Kartu Memori, Hardisk Eksternal dan perangkat penyimpanan data yang lain. Hal ini memungkinkan orang-orang yang tidak bertanggung jawab untuk membuka, melihat bahkan menyalin atau mengubah isi data tersebut. Agar data tersebut tidak bisa dibaca oleh orang yang tidak berhak, maka data tersebut harus dalam keadaan terenkripsi (disandikan).

Sistem lain yang akan menerima data yang dipindahkan, ada kemungkinan mempunyai format *database* yang berbeda. Perlu sebuah format data khusus yang berlaku secara umum sehingga data tersebut bisa dibaca di berbagai format *database* yang berbeda. Untuk memenuhi kebutuhan tersebut, maka sistem yang akan dibangun harus bisa menghasilkan data yang bisa dibaca oleh berbagai format *database* tersebut, dalam hal ini maka sistem akan men-generate data tersebut ke dalam format XML. Karena XML bersifat mudah untuk dibaca dan ditulis baik oleh manusia maupun komputer, maka XML merupakan sebuah format yang dapat digunakan untuk pertukaran data (*interchange*) antar aplikasi dan platform yang berbeda (*platform independent*).

2.2.2 Melakukan Perancangan Secara Cepat Sebagai Dasar Untuk Membuat Prototype

2.2.2.1 Use Case Diagram

Diagram Use Case dibuat untuk menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar dan menjelaskan sistem secara fungsional yang terlihat user. Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem dan bukan "bagaimana". Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem.Adapun diagram use case dari sistem enkripsi yang akan dibuat adalah sebagai berikut:



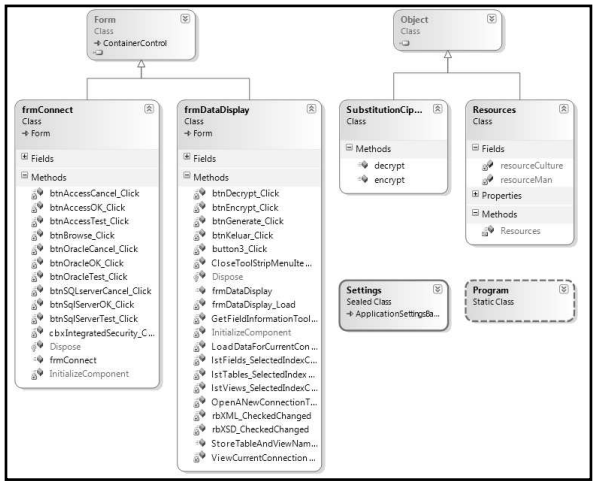
Gambar 3. Use Case Diagram Aplikasi Enkripsi

2.2.2.2 Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek. Class menggambarkan keadaan (*atribut/properti*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

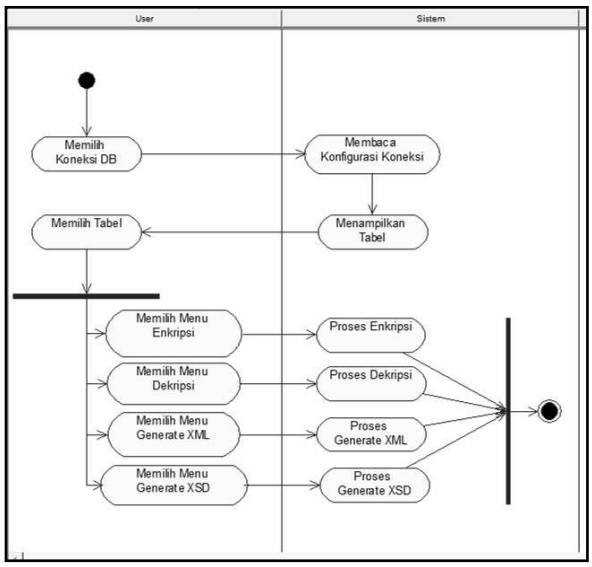
Adapun *HierarchiClass Diagram* dari sistem ini digambarkan sebagai berikut :



Gambar 4. Diagram *Hierarchi Class Symmetric Encryption*

2.2.2.3 Activity Diagram

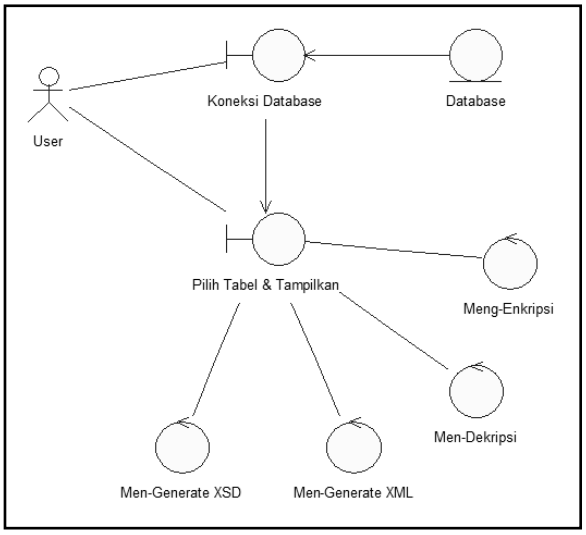
Diagram Aktivitas menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Diagram activitas tidak menggambarkan *behaviour internal* sebuah secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Aktivitas yang tejadi pada sistem ini digambarkan sebagai berikut :



Gambar 5. Diagram Aktivitas *Symmetric Encryption*

2.2.2.4 Robustness Diagram

Untuk menjelaskan hubungan interface, entitas dan proses maka dapat digambarkan dengan menggunakan diagram robustness. Diagram robustness dari sistem ini sebagai berikut:



Gambar 6. Diagram Robustness *Symmetric Encryption*

Dari diagram robustness diatas bisa dilihat bahwa user melalui interface tertentu (*form*) memilih koneksi *database* dalam hal ini terdapat 3 macam format *database* yaitu : SQL Server, Access dan Oracle. Jika koneksi berhasil, maka *database* yang bersangkutan akan dibaca. Kemudian melalui interface list, tabel dari *database* tersebut akan ditampilkan dan user bisa memilih tabel yang diinginkan untuk ditampilkan datanya dan proses. Proses tersebut meliputi : enkripsi, dekripsi, generate XML dan generate XSD.

2.2.2.5 Rancangan Algoritma Enkripsi

Algoritma enkripsi yang digunakan adalah enkripsi simetris subtitusi, yaitu dengan menggeser karakter sebanyak n. Nilai n ini bisa diubah-ubah sesuai kebutuhan karena n disini bersifat parametris.

Sedangkan algoritma enkripsi simetris subtitusinya sebagai berikut :

Algoritma ESubstitutionChiper(plaintext,n)

1. Baca plaintext
2. Ulang sebanyak panjang karakter plaintext
    a. Baca karakter demi karakter
    b. Convert karakter ke nilai integer
    c. Tambah dengan nilai n
    d. Kembalikan ke karakter

2.2.2.6 Rancangan Algoritma Dekripsi

Algoritma yang digunakan adalah simetris enkripsi, sehingga kunci yang digunakan dalam melakukan enkrispi dan dekripsi adalah sama. Oleh karena itu pada algoritma dekripsi ini, harus menggunakan nilai n yang sama dengan proses enkripsi agar data yang dihasilkan merupakan data yang sesuai dengan data asalnya.

Dan algoritma Dekripsinya dijelaskan sebagai berikut :

Algoritma DSubstitutionChiper(plaintext,n)

1. Baca plaintext hasil enkripsi
2. Ulang sebanyak panjang karakter plaintext
    a. Baca karakter demi karakter
    b. Convert karakter ke nilai integer
    c. Kurangi dengan nilai n
    d. Kembalikan ke karakter

Kedua algoritma diatas merupakan algoritma makro, artinya algoritma yang dibuat diatas hanya menjelaskan pokok-pokok langkah yang akan dikerjakan dengan kata lain hanya ditulis langkah-langkah globalnya saja.

2.2.3 Menguji Coba Dan Mengevaluasi Prototype

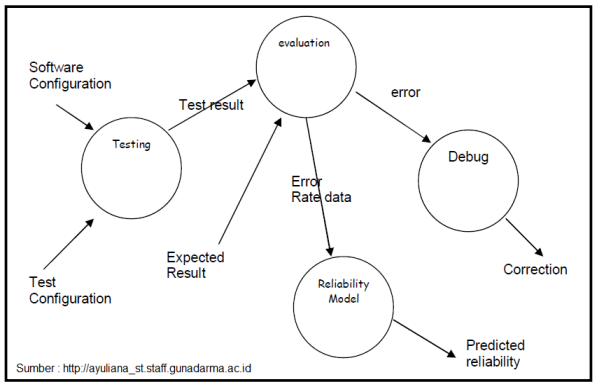
Ujicoba merepresentasikan ketidak-normalan yang terjadi pada pengembangan software. Selama definisi awal dan fase pembangunan, penulis berusaha untuk membangun *software* dari konsep yang abstrak

sampai dengan implementasi yang memungkinkan.

Pengembang membuat serangkaian uji kasus yang bertujuan untuk ”membongkar” *software* yang mereka bangun. Kenyataannya, ujicoba merupakan salah satu tahapan dalam proses pengembangan software yang dapat dilihat (secara psikologi) sebagai destruktif, dari pada sebagai konstruktif. Pengembang software secara alami merupakan orang konstruktif. Ujicoba yang diperlukan oleh pengembang adalah untuk melihat kebenaran dari software yang dibuat dan konflik yang akan terjadi bila kesalahan tidak ditemukan. Dari sebuah buku, Glen Myers (dalam Ayuliana:2009) menetapkan beberapa aturan yang dapat dilihat sebagai tujuan dari ujicoba :

1. Ujicoba merupakan proses eksekusi program dengan tujuan untuk menemukan kesalahan.
2. Sebuah ujicoba kasus yang baik adalah yang memiliki probabilitas yang tinggi dalam menemukan kesalahan-kesalahan yang belum terungkap.
3. Ujicoba yang berhasil adalah yang mengungkap kesalahan yang belum ditemukan.

Sehingga tujuan dari ujicoba ini adalah mendesain serangkaian tes yang secara sistematis mengungkap beberapa jenis kesalahan yang berbeda dan melakukannya dalam waktu dan usaha yang minimum. Jika pengujian diselenggarakan dengan sukses, maka akan membongkar kesalahan yang ada didalam perangkat lunak, manfaat lain dari pengujian adalah menunjukkan bahwa fungsi perangkat lunak telah bekerja sesuai dengan spesifikasi, dan kebutuhan fungsi telah tercapai. Sebagai tambahan, data yang dikumpulkan pada saat pengujian dilaksanakan akan menyediakan suatu indikasi keandalan perangkat lunak yang baik dan beberapa indikasi mutu perangkat lunak secara keseluruhan. Alur pengujian perangkat lunak dapat lihat pada gambar berikut :



Gambar 7. Alur Informasi Pengujian

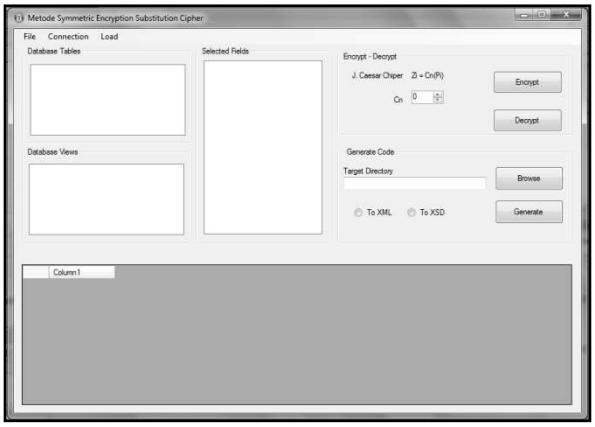
Dari gambar tersebut diatas, maka hal-hal yang perlu dipersiapkan dalam proses pengujian antara lain :

a. Konfigurasi Perangkat Lunak
   i. Menginstal SQL Server
   ii. Menginstal Microsoft Access
   iii. Menginstal Oracle
   iv. Menginstal C#
b. Konfigurasi Pengujian
   i. Menyiapkan *database* dalam format SQL Server
   ii. Menyiapkan *database* dalam format Microsoft Access
   iii. Menyiapkan *database* dalam format Oracle
c. Menetapkan hasil yang diharapkan
d. Mencatat Pelacakan Kesalahan

3. HASIL

Hasil dari penelitian ini adalah berupa perangkat lunak enkripsi yang dibuat dengan menggunakan bahasa pemrograman C# yang merupakan bagian dari Microsoft Visual Studio 2005. Perangkat lunak ini merupakan perangkat lunak aplikasi atau perangkat lunak bantu (*tools*) yang dimaksudkan untuk membantu bagian administrator *database* dalam melakukan pemindahan data dari *database* satu dengan *database* yang lain. Perangkat lunak ini dibutuhkan untuk menjamin keamanan data yang dipindahkan dengan cara mengenkripsi. Selain itu perangkat lunak ini bisa mengubah data menjadi format XML. Hal ini bertujuan agar data tersebut bisa dibaca dan diterima oleh *database* yang mempunyai format yang berbeda atau dengan kata lain bisa dibaca pada platform yang berbeda.

Sebelum melakukan enkripsi, user harus menentukan format *database* yang akan dibaca terlebih dahulu. Format *database* yang bisa dibaca oleh perangkat lunak ini antara lain : SQL Server Server 2005 (*SQL Server* 9.0.1399), *MicrosoftAccess* 2003 dan *Oracle Database 10g*. Selain itu, user harus menentukan jumlah pergeseran digit karakter sekaligus sebagai kunci enkripsi. Tampilan perangkat lunak ini sebagai berikut:



Gambar 8. Tampilan Menu Utama *Symmetric Encryption*

3.1 Menu File

Pada menu File ini hanya berisi menu *Close*. Menu ini berfungsi untuk keluar dari aplikasi.

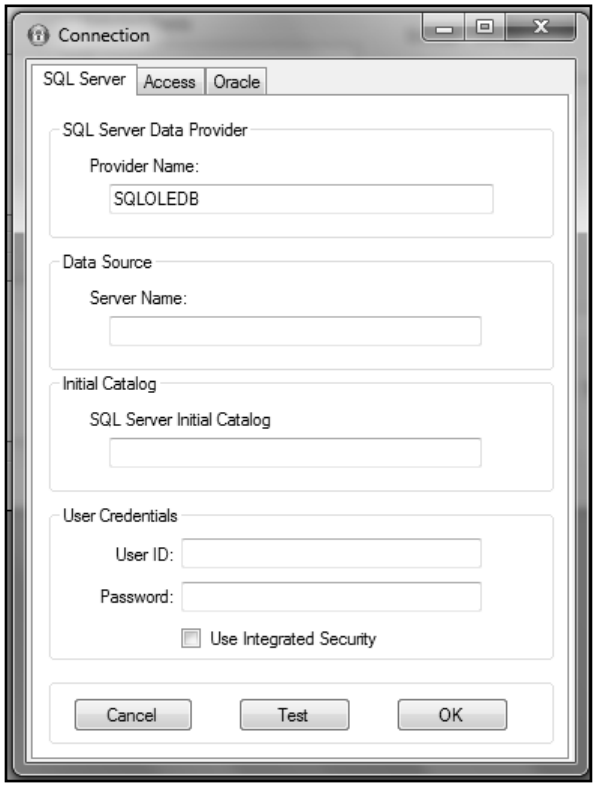
3.2 Connection

Menu *Connection* berfungsi untuk menentukan dan memilih jenis koneksi *database* yang akan digunakan serta menampilkan informasi koneksi *database* yang sedang terkoneksi, sehingga menu ini memiliki dua sub menu yaitu : *Open a New Connection* dan *View Current Connection. Open a New Connection* berguna untuk membuat koneksi baru sedangkan *View Current Connection* berfungsi menampilkan jenis koneksi yang sedang terjadi. Untuk memilih koneksi *database*, user harus mengerti jenis *database* apa yang akan diprosesnya, karena jenis *database* yang berbeda memerlukan jenis koneksi yang berbeda pula. Menu *Connection* ini mempunyai tiga macam jenis koneksi yaitu : SQL Server, Access dan Oracle.

3.2.1 Koneksi *Database* SQL Server

Berikut ini adalah menu koneksi *database* melalui SQL Server. Formulir yang perlu

diisikan pada *form* ini antara lain : Nama Server (*Server Name*), Nama *Database* (*SQL Server Initial Catalog*), *User Id* dan *Password*. Adapun form dari koneksi *database* SQL Server sebagai berikut:



Gambar 9. Tampilan Menu *Connection-SQL Server*

Penjelasan dari form diatas sebagai berikut :

a. *Server Name*, diisi dengan nama server dimana *database* yang akan dienkripsi tersebut disimpan. Apabila servernya lokal, maka biasanya nama server tersebut diisi sesuai nama komputer lokal tersebut.
b. *SQL Server Initial Catalog,* diisi berdasarkan nama *database* yang akan dibaca.
c. *User Credentials : User Id* dan *Password,* diisi sesuai login user dan *Password* untuk akses *database* pada SQL Server. Untuk Server lokal, biasanya dengan User Id : sa.
d. *Use Integrated Sequrity* digunakan bila login SQL Servernya menggunakan *Windows Authentication* artinya user id dan passwordnya sesuai dengan user id dan passswordnya windows yang digunakan pada server lokal.

Setelah semua form diisi maka untuk mengetahui koneksi berhasil atau tidak user menekan tombol *Test.* Jika koneksi tidak berhasil, maka akan muncul pesan seperti gambar berikut ini:

Gambar 10. Tampilan Koneksi Gagal

Sedangkan jika koneksi berhasil maka akan muncul pesan sebagai berikut:

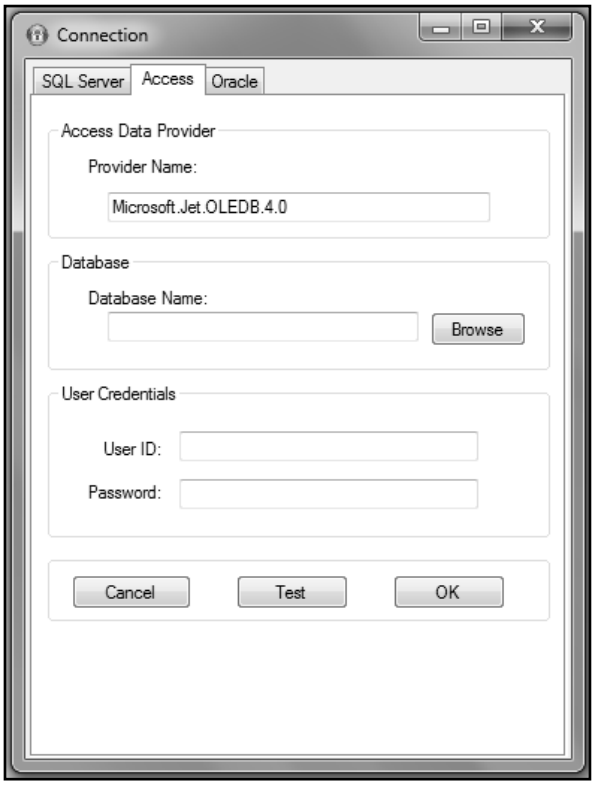
Gambar 11. Tampilan Koneksi Berhasil

Menu ini diakhiri dengan menekan tombol OK, dan tampilan akan kembali ke menu awal dari aplikasi ini.

3.2.2 Koneksi *Database* Microsoft Access

Berikut ini adalah menu koneksi *database* melalui Access. Formulir yang perlu diisikan

pada *form* ini antara lain : Nama *Database*, *User Id* dan *Password* jika ada. Untuk pengisian nama *database* dengan cara menekan tombol *Browse* dan arahkan pencarian file ke path dimana *database* access disimpan. Adapun form dari koneksi *database* Microsoft Access sebagai berikut:



Gambar 12. Tampilan Menu *Connection-Access*

Setelah semua isian diatas diisi lengkap, langkah berikutnya adalah menekan tombol *Test* untuk mengetahui keberhasilan dari koneksi, dan menekan tombol OK bila koneksi sudah berhasil.

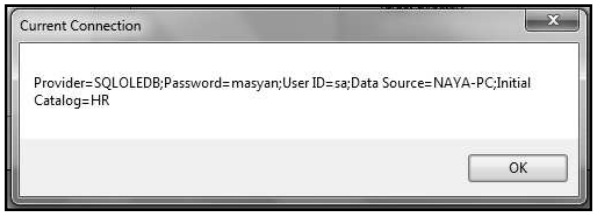
3.2.3 Koneksi *Database* Oracle

Berikut ini adalah menu koneksi *database* melalui Oracle. Formulir yang perlu diisikan pada *form* ini antara lain : Nama *Database*, *User Id* dan *Password*. Untuk pengisian nama *database* dengan cara mengetikkan langsung melalui form isian. Adapun form dari koneksi *database* Oracle sebagai berikut:

Gambar 13. Tampilan Menu *Connection-Oracle*

Setelah semua isian diatas diisi lengkap, langkah berikutnya adalah menekan tombol *Test* untuk mengetahui keberhasilan dari koneksi, dan menekan tombol OK bila koneksi sudah berhasil.

Setelah berhasil melakukan koneksi *database* baik *database* SQL Server, Access maupun Oracle, maka untuk melihat koneksi yang sedang berjalan saat ini dengan cara mengklik menu *View Current Connection.* Adapun tampilan dari menu tersebut sebagai berikut:

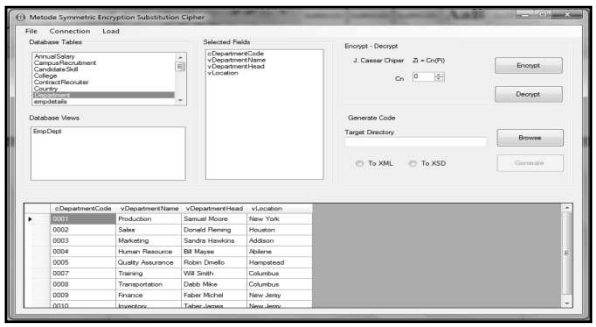


Gambar 14. Tampilan Menu *View Current Connection*

Tampilan *View Current Connection* diatas dilakukan dengan koneksi SQL Server, dengan nama *server* NAYA-PC, nama *database* HR, User Id sa dan password masyan. Dengan sudah adanya koneksi *database* seperti yang tersebut diatas, maka data tabel baru bisa dipanggil melalui menu *Load.*

3.3 Load

Menu *Load* berguna untuk memanggil data dari *database* yang telah terkoneksi. Dari menu ini akan menampilkan daftar tabel dari *database* yang bersangkutan. Daftar tabel tersebut akan ditampilkan pada data list. Setelah tabel-tabel tersebut ditampilkan, user bisa memilih salah satu tabel yang akan dienkripsi. Pada saat user memilih tabel, maka sistem akan menampilkan seluruh data (*record*) ke dalam datagrid, sehingga user bisa melihat isi dari tabel yang bersangkutan. Adapun tampilan hasil dari *Load data for Current Connection* sebagai berikut:



Gambar 15. Tampilan Menu *Load Data for Current Connection*

Dari menu ini, maka user baru bisa memilih menu *Encrypt, Decrypt* dan *Generate*.

3.4 Tombol Encrypt

Sebelum memilih *Encrypt*, user harus menentukan dulu nilai Cn (nilai pergeseran karakter). Berdasarkan nilai Cn inilah data akan dienkrip dengan cara menggeser karakter tersebut sebanyak n karakter. Algoritma untuk melakukan Enkripsi sebagai berikut :

Algoritma ESubstitutionChiper(plaintext,n)

1. Baca plaintext
2. Ulang sebanyak panjang karakter plaintext
    a. Baca karakter demi karakter
    b. Convert karakter ke nilai integer
    c. Tambah dengan nilai n
    d. Kembalikan ke karakter

Penerapan algoritma tersebut dalam implementasi program dibuat sebagai *class,* yaitu *class SubstitutionCipher* dan sebagai fungsi yaitu fungsi *encrypt* sehingga nantinya bisa dipanggil secara langsung dan berulang-ulang untuk melakukan proses enkripsi. Adapun *coding* algoritma enkripsi diatas sebagai berikut :

```
class SubstitutionCipher
    {
        public string encrypt(string val,
int mov)
        {

            string enkrip = "";
            for (int iChar = 0; iChar <
val.Length; iChar++)
            {
                enkrip += (char)(val[iChar]
+ mov);
            }
            return enkrip;
        }
    }
```

Hasil dari enkripsi bisa dilihat di gambar berikut ini:

cDepartmentCode	vDepartmentName	vDepartmentHead	vLocation
0001	Production	Samuel Moore	New York
0002	Sales	Donald Fleming	Houston
0003	Marketing	Sandra Hawkins	Addison
0004	Human Resource	Bill Mayse	Abilene
0005	Quality Assurance	Robin Dmello	Hampstead
0007	Training	Will Smith	Columbus
0008	Transportation	Dabb Mike	Columbus
0009	Finance	Faber Michel	New Jersy
0010	Inventory	Taber James	New Jersy

Gambar 16. Sebelum Dienkripsi

	cDepartmentCode	vDepartmentName	vDepartmentHead	vLocation
▶	<<<=	\~{pou{z	_myqx,Y{{~q	Zq,e{~w
	<<<>	_mxq	P{zmxp,Rxqyuzs	T{{z
	<<<?	Ym~wquzs	_mzp~m,Tmwuz	Mppu{z
	<<<@	Tymz,^q{~oq	Nuxx,Ym	Mnuxqzq
	<<<A	}mxu,M~mzoq	^{nuz,Pyqxx{	Tmy{qmp
	<<<C	`~muzuzs	cuxx,_yut	O{xyn
	<<<D	`~mz{~mu{z	Pmnn,Yuwq	O{xyn
	<<<E	Ruzmzoq	Rmnq~,Yuotqx	Zq,Vq~
	<<=<	Uzqz{~	`mpq~,Vmyq	Zq,Vq~

Gambar 17. Setelah Dienkripsi dengan Cn=12

3.5 Tombol Decrypt

Sedangkan untuk mendekripsi data, maka dengan algoritma sebagai berikut:

Algoritma DSubstitutionChiper(ciphertext,n)

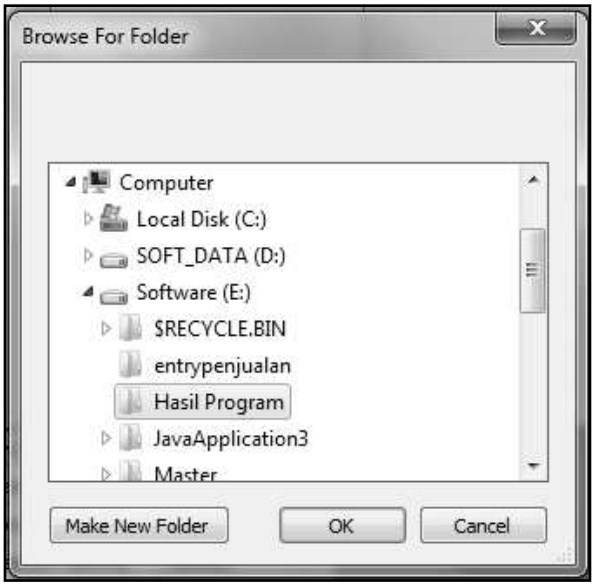
1. Baca ciphertext hasil enkripsi
2. Ulang sebanyak panjang karakter ciphertext
   a. Baca karakter demi karakter
   b. Convert karakter ke nilai integer
   c. Kurangi dengan nilai n
   d. Kembalikan ke karakter

Penerapan algoritma tersebut dalam implementasi program dibuat dalam *class* yang sama dengan fungsi *encrypt,* yaitu *class SubstitutionCipher* dan sebagai fungsi yaitu fungsi *decrypt* sehingga nantinya bisa dipanggil secara langsung dan berulang-ulang untuk melakukan proses dekripsi. Adapun *coding* algoritma dekripsi diatas sebagai berikut :

```
public string decrypt(string val, int
mov)
        {
            string dekrip = "";
            for (int iChar = 0; iChar
< val.Length; iChar++)
            {
                dekrip +=
(char)(val[iChar] - mov);
            }
            return dekrip;
        }
```

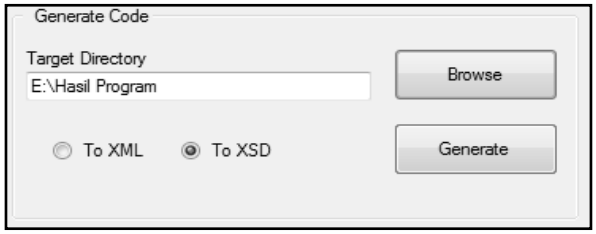
3.6 Tombol Browse

Tombol ini berfungsi untuk mengarahkan proses penyimpanan file hasil *generate,* baik file XML maupun file *schema XML* (*XSD*). Artinya tombol ini akan mengarahkan lokasi penyimpanan file, sesuai yang kita inginkan. *Path* file ini akan dituliskan pada *textbox Target Directory*. Selain menggunakan tombol *Browse* kita juga bisa menuliskan langsung lokasi penyimpanan file tersebut di *textbox Target Directory.* Gambar dibawah ini menjelaskan tampilan setelah penekanan tombol *Browse.*



Gambar 18. Tampilan Browse

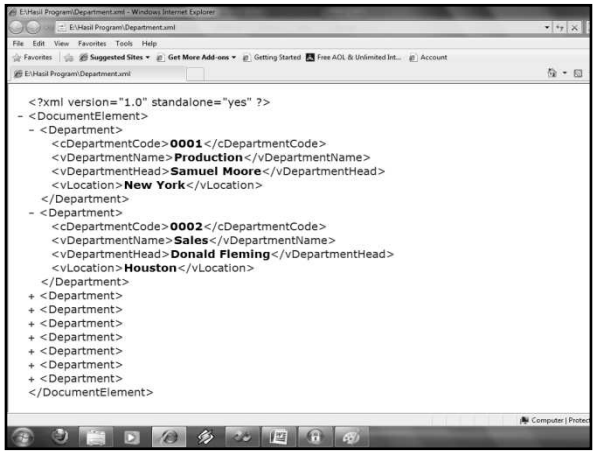
Dari gambar diatas bisa kita ketahui dimana lokasi penyimpanan file hasil generate nantinya, dan bahkan kita juga bisa membuat *folder* baru untuk lokasi penyimpanan file kita, yaitu dengan menekan tombol *Make New Folder* di *drive* manapun yang kita inginkan. Setelah menekan tombol OK maka akan muncul lokasi yang sudah dipilih pada *textbox Target Directory*, sebagaimana gambar dibawah ini.



Gambar 19. Pemilihan Lokasi Penyimpanan File

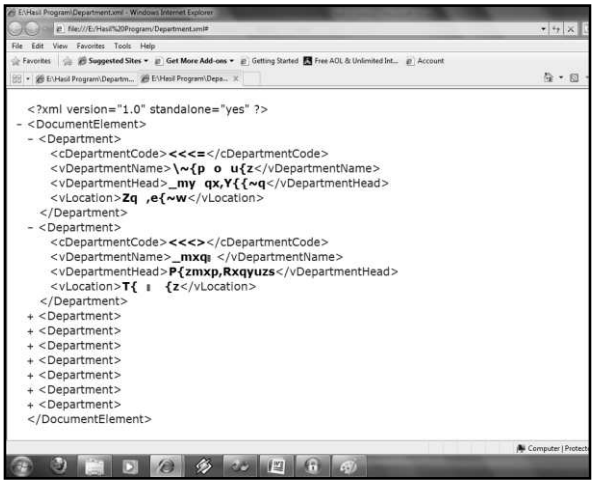
3.7 Tombol Generate

Setelah menentukan lokasi penyimpanan file, maka untuk menghasilkan file XML ataupun XSD langkah berikutnya adalah menekan tombol *Generate.* Gambar di bawah ini adalah hasil *Generate* ke XML sebelum dienkripsi terhadap tabel Department.



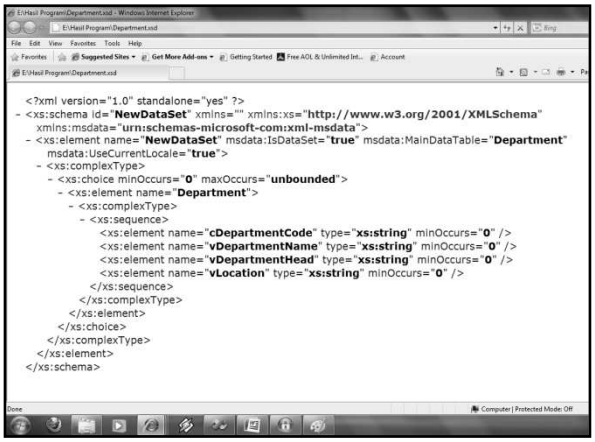
Gambar 20. Hasil Generate ke file XML Sebelum Dienkripsi

Sedangkan gambar dibawah ini adalah hasil generate ke XML setelah proses enkripsi. Tanda + (*plus*) di samping <Department> bisa di klik untuk menampilkan isi record department sehingga akan menjadi tanda – (*minus*).



Gambar 21. Hasil Generate ke file XML Setelah Dienkripsi

Tetapi apabila dipilih *generate* ke file Schema XML (file XSD) maka hasilnya bisa dilihat pada gambar 5.16. Dari gambar ini bisa diketahui *field-field* yang ada pada tabel *department* antara lain: *cDepartmentCode, vDepartmentName, vDepartmentHead* dan *vDepartmentLocation*.



Gambar 22. Hasil Generate ke file Schema XML (XSD)

4. SIMPULAN

Dari penelitian yang telah dilakukan, maka dapat ditarik beberapa kesimpulan, antara lain:

a. Penelitian ini menghasilkan sebuah perangkat lunak aplikasi pemanfaatan metode *symmetric encryption substitution cipher.*
b. Aplikasi metode *symmetric encryption substitution cipher* bisa digunakan untuk mengenkripsi 3 jenis *database*, yaitu: SQL Server 2005 (*SQL Server* 9.0.1399), *MicrosoftAccess* 2003 dan *Oracle Database 10g*.
c. Proses enkripsi dapat dilakukan berulang-ulang dengan key yang berbeda-beda sesuai nilai Cn.
d. Hasil enkripsi bisa digenerate ke dalam format XML atau XSD tanpa mengubah data sumbernya.

DAFTAR RUJUKAN

Kelompok 124 IKI-83408 MTI UI. 2005. *Cryptography*, http://bebas.vlsm.org/v06/Kuliah/MTI-Keamanan-Sistem-Informasi/2005/124/124P-04-draft-Cryptography.pdf. Diakses Tanggal 20 Nopember 2010.

Munir, Rinaldi. 2006. *Kriptografi***.** Bandung : Penerbit Informatika.

NIIT. 2009. *Introduction to Web Content Development***.**India:NIIT.

Noname. http://digilib.its.ac.id/public/ITS-Undergraduate-5121-4201100038-bab3.pdf. Diakses Tanggal 20 Nopember 2010.

Prasetyo, Didik Dwi. 2006. *Pemgrograman Aplikasi Database Vb.net 2005 + Cd*. Jakarta : Elex Media Komputindo.

Rahardjo, Budi. 2002. *Keamanan Sistem informasi Berbasis Internet***.** Bandung : PT. Insan Komunikasi Indonesia.

Suryana, Aulya dkk. 2007. *Enkripsi***.** Sekolah Tinggi Manajemen Informatika Dan Teknik Komputer (STMIK). Bali : STIKOM.