# PUSTAKA SANDI KLASIK BERBASIS *COMPONENT OBJECT MODEL* (COM)

M. Miftakul Amin [1]

[1] Jurusan Teknik Komputer, Politeknik Negeri Sriwijaya
Jl. Srijaya Negara Bukit Besar Palembang 30139
Telp. (0711) 353414 Fax. (0711) 355918
e-mail : miftakul_a@polsri.ac.id

## *ABSTRACT*

*Security and confidentiality are important issues that need serious attention in the era of information technology. Cryptography is one technique for securing data and information. In cryptography, the data is disguised in such a way that even if the data can be read, it can not be understood by those who do not have the authority. Data that has not undergone encoding known as The term plaintext, and after camouflaged with an encryption method, then it will turn plaintext into ciphertext. In general, a system equipped with a password to restrict only certain parties were entitled to enter into the system. If the password can be solved or known by an unauthorized person, then the data or information that is in the system is threatened. This research aims to develop a library classical password that can be used as an attempt to realize the aspect of information security in access control, by utilizing the technology component object model (COM). From the results of tests performed can be seen that the COM libraries are packed in COM DDL can be used by a variety of programming languages fairly easily.*

***Keywords—*** *cryptography, component object model (COM)*

## ABSTRAK

Keamanan dan kerahasiaan merupakan hal penting yang perlu mendapatkan perhatian serius dalam era teknologi informasi. Kriptografi merupakan salah teknik untuk mengamankan data dan informasi. Dalam kriptografi, data disamarkan sedemikian rupa sehingga kalaupun data itu dapat dibaca, maka tidak bisa dimengerti oleh pihak yang tidak mempunyai wewenang. Data yang belum mengalami penyandian dikenal dengan isitilah *plaintext*, dan setelah disamarkan dengan suatu cara penyandian, maka *plaintext* ini akan berubah menjadi *ciphertext*. Pada umumnya sebuah sistem dilengkapi dengan *password* untuk membatasi hanya pihak tertentu saja yang berhak masuk ke dalam sistem. Jika *password* tersebut dapat dipecahkan atau diketahui oleh pihak yang tidak berwenang, maka data atau informasi yang ada di dalam sistem tersebut menjadi terancam. Penelitian ini bertujuan untuk mengembangkan sebuah pustaka sandi klasik yang dapat digunakan sebagai usaha untuk mewujudkan keamanan informasi dalam aspek *access control*, dengan memanfaatkan teknologi *component object model* (COM). Dari hasil pengujian yang dilakukan dapat dilihat bahwa *library* COM yang dikemas dalam COM DDL dapat digunakan oleh beragam bahasa pemrograman dengan cukup mudah.

***Kata Kunci—*** *kriptografi, component object model (COM)*

## 1. PENDAHULUAN

Perkembangan teknologi komputer yang begitu pesat tidak selamanya memberikan dampak positif, dampak negatif juga bermunculan sebagai akibat penyalah gunaan teknologi yang meresahkan berbagai pihak. Masalah keamanan merupakan salah satu aspek penting dalam infrastuktur teknologi informasi yang perlu mendapatkan perhatian serius bagi perancang sistem. Banyak kejahatan *cyber* yang menyebabkan kerugian besar. Hampir semua aspek kehidupan masyarakat tidak dapat dipisahkan dari peran teknologi informasi, apalagi informasi tersebut lebih mudah didapatkan dengan dukungan jaringan komputer dan internet.

Keamanan komputer yang perlu mendapatkan perhatian mencakup beberapa aspek [1], diantaranya:

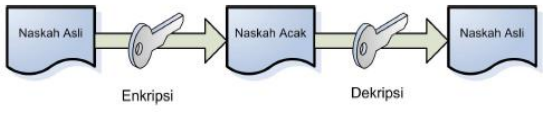
a. *Authentication*, dalam hal ini penerima informasi dapat memastikan keaslian pesan tersebut datang dari orang yang dimintai informasi dan berasal dari orang yang dikehendaki.
b. *Integrity*, keaslian pesan yang dikirim dapat dipastikan bahwa informasi yang dikirim tidak dimodifikasi oleh orang yang tidak berhak dalam perjalanan informasi tersebut.
c. *Nonrepudation*, merupakan fokus pada pihak pengirim informasi. Dalam hal ini si pengirim tidak dapat megelak bahwa dia yang bertanggung jawab mengirim informasi tersebut.
d. *Authority*, informasi yang berada pada sistem tidak dapat dimodifikasi oleh pihak yang tidak berhak atas akses tersebut.
e. *Confidentiality*, merupakan usaha untuk menjaga informasi dari orang yang tidak berhak mengakses.
f. *Privacy*, merupakan lebih ke arah data-data yang sifatnya pribadi.
g. *Availability*, merupakan ketersediaan informasi ketika dibutuhkan. Sistem yang diserang atau dijebol dapat menghambat atau meniadakan akses ke informasi.
h. *Access control*, aspek ini berhubungan dengan cara pengaturan akses kepada informasi. Aspek ini berhubungan dengan masalah *authentication* dan juga *privacy*. Access control seringkali dilakukan menggunakan kombinasi *user id* dan *password* maupun dengan mekanisme tertentu.

Di samping beberapa aspek keamanan juga perlu diperhatikan beberapa aspek ancaman, diantaranya:

a. *Interruption*, merupakan ancaman terhadap *availability*, dimana informasi dan data dirusak atau dihapus.
b. *Interception*, merupakan ancaman terhadap kerahasiaan (*secrecy*), dalam hal ini informasi yang ada disadap atau orang yang tidak berhak mendapatkan akses ke komputer di mana informasi tersebut disimpan.
c. *Modifikasi*, merupakan ancaman terhadap integritas, dimana orang yang tidak berhak berhasil menyadap lalu lintas informasi yang sedang dikirim dan diubah sesuai keinginan orang tersebut.
d. *Pabrikasi*, merupakan ancaman terhadap integritas, dalam hal ini orang yang tidak berhak berhasil meniru (memalsukan) suatu informasi yang ada sehingga orang yang menerima informasi tersebut menyangka informasi berasal dari orang yang dikehendaki oleh pihak yang menerima informasi.

Salah satu usaha untuk mewujudkan keamanan informasi dan data dalam sistem komputer adalah menggunakan kriptografi. Kriptografi tidak hanya menyediakan alat untuk keamanan pesan, juga berisi sekumpulan teknik yang berguna. Kriptografi berasal dari bahasa yunani "*cryptos*" yang berarti rahasia dan "*graphein*" yang berarti tulisan [2]. Kriptografi merupakan ilmu mengenai teknik enkripsi dimana data dilacak menggunakan suatu kunci enkripsi menjadi sesuatu yang sulit dibaca oleh seseorang yang tidak memiliki kunci dekripsi. Dekripsi menggunakan kunci dekripsi untuk mendapatkan kembali data asli [3]. Proses enkripsi dilakukan dengan menggunakan suatu algoritma dengan beberapa parameter.

Dalam kriptografi klasik, teknik enkripsi yang digunakan adalah enkripsi simetris dimana kunci dekripsi sama dengan kunci enkripsi. Gambar 1 merupakan ilustrasi proses enkripsi dan dekripsi, yang menunjukkan efek dari proses enkripsi yang berfungsi untuk mengacak naskah asli (*plaintext*) menjadi naskah acak (*ciphertext*) yang sulit untuk dibaca oleh seseorang yang tidak mempunyai kunci dekripsi.




Gambar 1 Proses Enkripsi dan Dekripsi

*Component object model* (COM) merupakan spesifikasi untuk menulis software yang dapat digunakan kembali (*reusable*). COM juga merupakan model yang didasarkan pada *binary reuse*, artinya bahwa komponen ini tidak bergantung

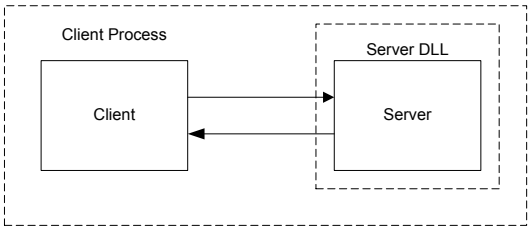
pada bahasa pemrograman untuk membuatnya. Di samping itu COM sendiri memakai sistem *object oriented*. Beberapa definisi dari COM dapat diketahui dari beberapa definisi sebagai berikut [4]:

a. COM adalah spesifikasi, yang menggambarkan suatu standar yang memungkinkan kita melakukan *interoperable* COM. Standar inilah yang harus diikuti dalam membuat COM.
b. COM adalah kumpulan *service*, yang dibuat untuk service terutama COM *library* yang merupakan bagian dari sistem operasi pada *platform Win32*. Ada juga yang berfungsi sebagai *package* yang terpisah dari sistem operasi sehingga COM dapat dikatakan sebagai kumpulan *service*.
c. COM adalah *modular programming*, komponen COM dapat di-*pakcage* dalam bentuk DLL dan EXE yang menyediakan mekanisme komunikasi yang mengizinkan antar komponen untuk berkomunikasi.
d. COM adalah *object oriented model*, merupakan sebuah *true object* yang mempunyai identitas. Disamping itu COM juga dapat melakukan *polymorphism* sehigga COM menerapkan sistem *object oriented* dalam mekanisme pembuatannya.
e. COM mudah dalam penggunaanya dan upgrade pada aplikasi, komponen COM mudah dibuat dan dapat dihubungkan dengan komponen lain secara dinamis. Komponen COM juga menerapkan sistem lokasi sehingga COM tidak perlu di-*compile* ulang, cukup meletakkannya ke tempat awalnya.
f. COM dapat didistribusi, yang mempunyai kemampuan untuk mendistribusikan baik di komputer lokal maupun di luar komputer. Teknologi ini disebut dengan *Distributed COM* (DCOM).

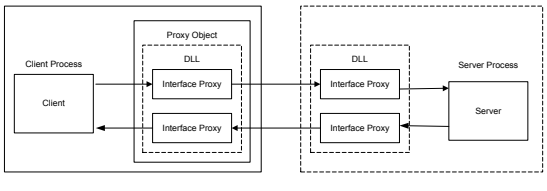
COM dapat dibuat dengan banyak bahasa pemrograman, COM adalah *binary standard* sehingga COM dapat dibuat oleh banyak bahasa pemrograman seperti C/C++, Java, Jscript, Visual Basic, VBScript, Delphi, Power Builder dan lain-lain.

Secara arsitektural COM dapat berupa 2 model, yaitu COM DLL dan COM EXE. Komponen ini dapat dijalankan pada komputer lain melalui *remot*e. Sebagian besar aplikasi *client* tidak peduli di mana lokasi file COM server berada. Gambar 2 merupakan ilustrasi sederhana yang menggambarkan hubungan antara COM server dan COM *client* untuk COM DLL, sedangkan COM EXE dapat dilihat pada Gambar 3. Sistem COM DLL menerapkan

*in-process* yang mana DLL server akan dibawa masuk ke dalam proses dari aplikasi *client*. Aplikasi *client* juga tidak dapat memproteksi proses penulisan yang dilakukan oleh DLL server sehingga hal ini terkadang membuat aplikasi menjadi *crash*. Sedangkan untuk COM EXE, sistem operasi akan mengizinkan COM EXE server untuk memproteksi dirinya sendiri sehingga disediakan alokasi memori di lokasi tertentu.




Gambar 2 Arsitektur COM DLL




Gambar 3 Arsitektur COM EXE

Proses yang terjadi pada COM EXE adalah *out-process*, sehingga hal ini menyebabkan kinerjanya lebih rendah dibandingkan dengan COM DLL karena ada *cross process* antara EXE server dan aplikasi *client*. Komunikasi antara COM EXE *server* dan aplikasi *client* melalui *proxy-stub* DLL. Ketika COM EXE server sedang *running*, EXE server akan menciptakan *interface stub*, sedangkan di sisi aplikasi *client* akan diciptakan *interface proxy* sehingga melalui *interface* inilah terjadinya proses komunikasi. Baik untuk COM DLL maupun COM EXE masing-masing mempunyai kelebihan dan kekurangan, tergantung kepada kebutuhan dari para pengembang aplikasi.

Beberapa penelitian terkait dengan kriptografi klasik pernah dilakukan oleh Nathasia [5] yang mengembangkan teknik kriptografi untuk pengamanan basis data. Kriptografi yang digunakan adalah *stream-cipher* dengan melakukan beberapa proses enkripsi terhadap data yang tersimpan di dalam tabel. Penelitian serupa juga dilakukan oleh Dharmawan [6] yang mengembangkan kirptografi pada proses login di halaman web. Serangan yang sering terjadi pada login halaman web adalah adanya *SQL Injection*. Proses login menjadi lebih baik dengan ditanamnya kriptografi pada proses login. Sedangkan Juliadi [7] melakukan proses untuk memperkuat teknik kriptografi dengan mengkombinasikan kriptografi klasik *affine* dan *vignere cipher*. Proses enkripsi dilakukan sebanyak 2 kali sehingga menghasilkan karakter yang berbeda dengan *plaintext*, sehingga teks yang dihasilkan tidak mudah untuk didekripsi. Winantu [8] mencoba menterjemahkan kriptografi klasik ke dalam bahasa pemrograman PHP. Dalam penelitiannya Winantu hanya membuat bagaimana proses enkripsi *Caesar cipher*

dapat dilaksanakan, dan hanya sebatas karakter A-Z. Sementara karakter yang lainnya belum dapat diujicoba.

Penelitian ini bertujuan untuk mengembangkan sebuah pustaka (*library*) sandi klasik yang dapat digunakan sebagai usaha untuk mewujudkan keamanan informasi dalam aspek *access control*, dengan memanfaatkan teknologi *component object model* (COM).

## 2. METODE PENELITIAN

Kriptografi klasik yang digunakan dalam penelitian ini adalah teknik substitusi yaitu dengan mengganti setiap karakter dari *plaintext* dengan karakter lainnya [1]. Teknik substitusi yang digunakan adalah *Caesar Cipher*, yaitu dengan mengganti posisi huruf awal dari *alphabet*, sebagai contoh:

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | B | C | D | E | F | G | H | I | J | K | L | M |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| | | | | | | | | | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| D | E | F | G | H | I | J | K | L | M | N | O | P |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| Q | R | S | T | U | V | W | X | Y | Z | A | B | C |

Proses untuk melakukan enkripsi dan dekripsi menggunakan caesar cipher menggunakan notasi matematis untuk melakukan proses enkripsi dari *plaintext*:

$$C = E(P) = (P + K) \bmod 26 \quad ...1)$$

Sedangkan proses untuk melakukan dekripsi dari *ciphertext*:

$$P = D(C) = (C - K) \bmod 26 \quad ... 2)$$

Jika terdapat *plaintext* : SAYA BELAJAR

Menjadi *ciphertext* : VDBD EHODMDU

Selanjutnya untuk membuat COM DLL digunakan bahasa pemrograman visual basic 6 dengan membuat 2 buah fungsi, yaitu fungsi EnkripsiCaesar() dan DekripsiCaesar(). Kedua fungsi tersebut memiliki 2 buah parameter dan dikemas dalam bentuk *class module*.

**Fungsi EnkripsiCaesar**

```
Public Function EnkripsiCaesar(ByVal plaintext As
String, kunci As Integer) As String
    Dim teks As String
    Dim P As String
    Dim C As String
    Dim indexAbjad(26) As String
    Dim indexChiper As String
    Dim PanjangTeks As Integer
    Dim i, j As Integer
    Dim posisi As Integer
    Dim hiddentext As String

    indexAbjad(0) = "A": indexAbjad(10) = "K":
indexAbjad(20) = "U"
    indexAbjad(1) = "B": indexAbjad(11) = "L":
indexAbjad(21) = "V"
    indexAbjad(2) = "C": indexAbjad(12) = "M":
indexAbjad(22) = "W"
    indexAbjad(3) = "D": indexAbjad(13) = "N":
indexAbjad(23) = "X"
    indexAbjad(4) = "E": indexAbjad(14) = "O":
indexAbjad(24) = "Y"
    indexAbjad(5) = "F": indexAbjad(15) = "P":
indexAbjad(25) = "Z"
    indexAbjad(6) = "G": indexAbjad(16) = "Q"
    indexAbjad(7) = "H": indexAbjad(17) = "R"
    indexAbjad(8) = "I": indexAbjad(18) = "S"
    indexAbjad(9) = "J": indexAbjad(19) = "T"

    PanjangTeks = Len(UCase(Trim(plaintext)))
    For i = 1 To PanjangTeks
        If Mid(Trim(plaintext), i, 1) <> Chr(32)
Then
            P = Mid(plaintext, i, 1)
            'mengambil index
            For j = 0 To 25
                'jika karakter adalah antara A-Z
                If UCase(indexAbjad(j)) =
UCase(P) Then
                    posisi = j
                    indexChiper = (posisi +
kunci) Mod 26
                    C = indexAbjad(indexChiper)
                    'mempertahankan besar dan
kecilnya huruf plaintext
                    If (StrComp(P, C, 0) = 1)
Then C = LCase(C)
                    Exit For
                Else
                    'jika karakter selain A-Z
                    C = P
                End If
            Next j

            hiddentext = hiddentext + C
        Else
            hiddentext = hiddentext + " "
        End If
    Next i
    EnkripsiCaesar = hiddentext
End Function
```

```
Fungsi DekripsiCaesar
Public Function DekripsiCaesar(ByVal chipertext
As String, kunci As Integer) As String

    Dim teks As String
    Dim P As String
    Dim C As String
    Dim indexAbjad(26) As String
    Dim indexChiper As String
    Dim indexAscii As Integer
    Dim PanjangTeks As Integer
    Dim h, i, j As Integer
    Dim posisi As Integer
    Dim hiddentext As String
    Dim Abjad As String

    indexAscii = 65 + kunci
    For h = 0 To 25
        Abjad = Chr(indexAscii)
        If Abjad > Chr(90) Then
            indexAscii = 65
            indexAbjad(h) = Chr(indexAscii)
            indexAscii = indexAscii + 1
        Else
            indexAbjad(h) = Abjad
            indexAscii = indexAscii + 1
        End If
    Next h

    PanjangTeks = Len(UCase(Trim(chipertext)))
    For i = 1 To PanjangTeks
        If Mid(Trim(chipertext), i, 1) <> Chr(32)
Then
            C = Mid(chipertext, i, 1)

            'mengambil index
            For j = 0 To 25
                'jika karakter adalah A-Z
                If UCase(indexAbjad(j)) =
UCase(C) Then
                    posisi = j
                    indexChiper = (posisi -
kunci) Mod 26
                    If (indexChiper < 0) Then
indexChiper = indexChiper + 26
                    P = indexAbjad(indexChiper)
                    'mempertahankan besar dan
kecilnya huruf chipertext
                    If (StrComp(C, P, 0) = 1)
Then P = LCase(P)
                    Exit For
                Else
                    'jika karakter selain A-Z
                    P = C
                End If
            Next j
            hiddentext = hiddentext + P
        Else
            hiddentext = hiddentext + " "
        End If
    Next i
    DekripsiCaesar = hiddentext
End Function
```

File DLL yang terbentuk diberi nama KriptoCOMDLLPrj.dll dan disimpan di dalam lokasi folder C:\Windows\System32\. Langkah selanjutnya melakukan proses registrasi ke dalam sistem operasi, supaya COM DLL yang telah dibuat dapat dikenali oleh bahasa pemrograman yang berjalan dalam sistem operasi windows. Untuk melakukan proses registrasi file COM DLL digunakan perintah:

```
Regsvr32
"C:\Windows\System32\KriptoCOMDLLP
rj.dll"
```

## 3. HASIL DAN PEMBAHASAN

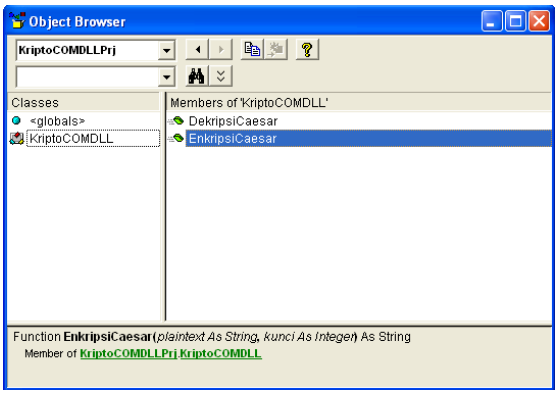
Dalam implementasinya COM DLL yang telah terbentuk dapat digunakan sebagai sebuah file pustaka (*library*) yang dapat digunakan oleh bahasa pemrograman yang berjalan di atas sistem operasi windows. Tugas dari file *library* tersebut adalah melakukan proses enkripsi dan dekripsi sesuai dengan algoritma *caesar cipher* yang dapat dilihat pada Tabel 1 berisi beberapa kata yang digunakan sebagai pengujian *password*. Tampak pada data pengujian, bahwa karakter yang berubah hanyalah karakter A-Z, selebihnya akan tetap dipertahankan.

Tabel 1. Pengujian Karakter

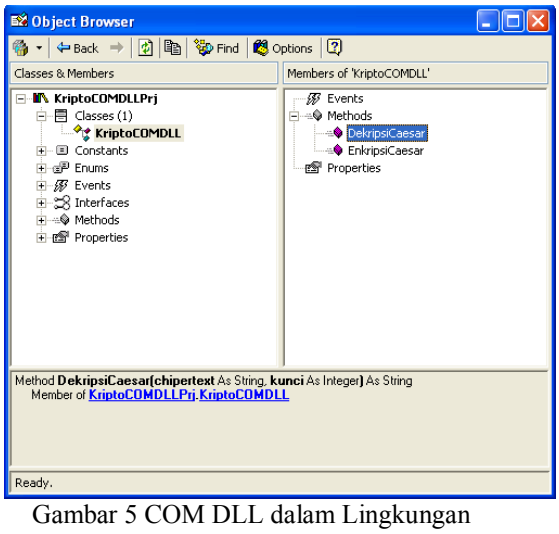
| **Kata** | **Proses Enkripsi C = E (P)** | **Proses Dekripsi P = D (P)** |
|---|---|---|
| amin2015 | dplq2015 | Amin2015 |
| ayah | dbdk | ayah |
| yevi2015 | bhyl2015 | yevi2015 |
| sifa2015 | vlid2015 | sifa2015 |
| mbak | pedn | Mbak |
| fira2015 | ilud2015 | fira2015 |
| adek | dghn | Adek |
| fa4%89Dhu | id4%89Gkx | fa4%89Dhu |
| s3rdos&*^t | v3ugrv&*^w | s3rdos&*^t |
| say123 | vdb123 | say123 |

Pada proses pengujian ini akan dicoba pada 2 buah bahasa pemrograman yang mendukung pemrosesan COM DLL,

yaitu MS. Visual Basic 6.0 dan MS. Visual FoxPro 6.0. Kedua bahasa pemrograman tersebut dapat menelusuri keberadaan COM DLL di dalam lingkungan bahasa pemrogramannya seperti dapat dilihat pada Gambar 4 dan 5. Pada kedua bahasa pemrograman tersebut terlihat bahwa KriptoCOMDLLPrj merupakan file pustaka (*library*) yang secara fisik dikenali oleh sistem operasi windows. Sedangkan KriptoCOMDLL merupakan nama *class* yang dimiliki oleh KriptoCOMDLLPrj. Dan dalam class KriptoCOMDLL terdapat dua method yaitu DekripsiCaesar() dan EnkripsiCaesar(). Kedua method tersebut memiliki parameter yang sama, yaitu *plaintext/ciphertext* dan nilai pergeseran sesuai dengan formula 1) dan 2).




Gambar 4 COM DLL dalam Lingkungan Kerja MS. Visual Basic 6.0

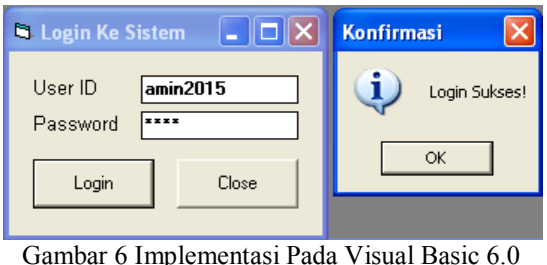



Gambar 5 COM DLL dalam Lingkungan Kerja MS. Visual FoxPro 6.0

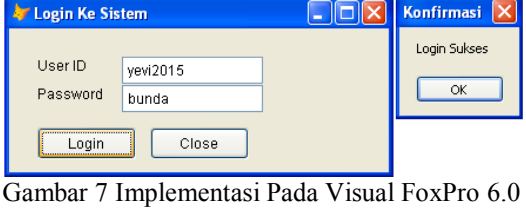
Notasi program yang digunakan untuk mereferensi file pustaka (*library*) yang terbentuk adalah dengan cara *early binding* seperti berikut:

```
Dim obj As New KriptoCOMDLL
obj.DekripsiCaesar(rs.Fields("pwd_c").Va
lue, 3)
```

Gambar 6 dan 7 memperlihatkan implementasi dalam bahasa pemrograman Visual Basic 6.0 dan Visual FoxPro 6.0.




Gambar 6 Implementasi Pada Visual Basic 6.0




Gambar 7 Implementasi Pada Visual FoxPro 6.0

## 4. SIMPULAN

1. Dengan menggunakan *Component Object Model* (COM) memudahkan dalam mengemas sebuah algoritma untuk melakukan proses enkripsi dan dekripsi yang dapat digunakan oleh beberapa bahasa pemrograman.
2. Untuk melakukan enkripsi dan dekripsi cukup dilakukan dengan mengirimkan parameter berupa teks yang akan diproses dan nilai pergeseran sebagai implementasi dari teknik substitusi.
3. Karakter yang dapat disubstitusi merupakan karakter A-Z, sedangkan karakter lainnya dipertahankan sebagaimana karakter aslinya.

## PENELITIAN LANJUTAN

Penelitian lanjutan yang dapat dilakukan adalah dengan mencoba menanamkan beberapa metode atau algoritma kriptografi ke dalam sebuah file COM DLL (*library*). Pada penelitian ini hanya satu metode saja yang ditanamkan dalam file COM DLL (*library*). Penambahan metode kriptografi ke dalam COM DLL akan memberikan beragam alternatif yang dapat dipilih untuk memproses data yang akan dienkripsi maupun didekripsi.

## UCAPAN TERIMA KASIH

Ucapan terimakasih kepada seluruh teman sejawat di Jurusan Teknik Komputer Politeknik Negeri Sriwijaya Palembang, juga kepada seluruh civitas akademika IIB darmajaya Bandar Lampung terutama kepada lembaga penelitian yang telah memberikan kesempatan kepada penulis sehingga naskah ini dimuat dalam jurnal Informatika.

## DAFTAR PUSTAKA

[1] Ariyus, Doni. 2006. “Computer Security”. Yogyakarta: Penerbit Andi Offset Yogyakarta.

[2] Munir, Rinaldi. 2006. “Kriptografi”. Bandung: Penerbit Informatika Bandung.

[3] Kromodimoeljo, Sentot. 2009. “Teori dan Aplikasi kriptografi”. Jakarta: Penerbit SPK IT Consulting.

[4] Kurniawan, Agus. 2003. “Pemrograman COM, DCOM dan COM+ dengan Visual Basic 6.0”. Jakarta: Penerbit Elexmedia Komputindo.

[5] Nathasia, Dian, Novi; Wicaksono, Eko, Anang. 2011. “Penerapan Teknik Kriptografi Stream Cipher Untuk Pengamanan Basis Data”, Jurnal Basis Data, ICT Research

Center, Unas, Volume 6 Nomor 1 Tahun 2011.

[6] Dharmawan, Adhitya, Eka; Yudaningtyas, Erni; Sarosa, M. 2013. "Perlindungan Web Pada Login Sistem Menggunakan Algoritma RIjndael", Jurnal EECCIS Volume 7 Nomor 1 Juni 2013

[7] Juliadi; Prihandono, Bayu; Kusumastuti, Nilamsari. 2013. "Kriptografi Klasik Dengan Metode Modifikasi Affine Cipher Yang Diperkuat Dengan Vignere Cipher", Buletin Ilmiah Mat. Stat. dan Terapannya (Bimaster) Volume 2 Nomor 2 Tahun 2013

[8] Winantu, Asih. 2014. "Impelementasi Algoritma Kriptografi Klasik Ke Dalam Bahasa Pemrograman PHP", Yogyakarta: STMIK Elrahma