Implementasi Metode Enkripsi Pergeseran Biner Dengan Microsoft Visual Basic 6.0

Kusuma Prakoso, Haryani, Gesang Kristianto Nugroho
Universitas Surakarta

ABSTRACT : System bind on computer personal using platform of Microsoft Windows™ as machine of activator of public system because cannot prevent all hacker, cracker and attacker to enter and exploit the content from pertinent computer system for the sake of negativity. Especially at network of computer of peer to peer own the Microsoft Windows™ and public network [of] internet. So that as for how to conduct the implementation of binary method encryption shifting with Visually Basic 6.0 competent to be strived.

This research aim to to prove that competent binary method encryption shifting implemented in everyday life all civil consumer for the protection of their data from acting cyber-crime, providing application program light cryptography and easy to just used by whom and for the sake of any kind of, security importance applicable to bind the static (in a stand alone computer and local of area network) and also in a community owning to bind the dinamic (mobile) such as a community in (virtual) private network and public network (internet) and also in physical delivery.

The result of this research can present the data qualitative result of technical measurement and examination according to principles of software engineering so that application elegibility result of accountable engineering.

Keyword : cryptography, encryption, binary shifting

Abstraksi : Mengikat Sistem pada komputer pribadi menggunakan platform Microsoft Windows ™ sebagai mesin penggerak sistem publik karena tidak dapat mencegah semua hacker, cracker dan penyerang untuk masuk dan mengeksploitasi konten dari sistem komputer yang bersangkutan untuk kepentingan negatif: ABSTRAK. Terutama pada jaringan komputer peer to peer memiliki Microsoft Windows ™ dan jaringan publik [dari] internet. Sehingga sebagai untuk bagaimana melakukan implementasi enkripsi metode biner pergeseran dengan Visual Basic 6.0 kompeten untuk diupayakan. Penelitian ini bertujuan untuk membuktikan bahwa metode biner enkripsi yang kompeten tentang pengalihan diimplementasikan dalam kehidupan sehari-hari semua konsumen sipil untuk perlindungan data mereka dari bertindak kejahatan cyber, menyediakan kriptografi aplikasi ringan Program dan mudah hanya digunakan oleh siapa dan untuk kepentingan apapun dari, keamanan penting yang berlaku untuk mengikat statis (dalam saja komputer yang berdiri dan lokal dari jaringan area) dan juga dalam masyarakat yang memiliki mengikat dinamis (mobile) seperti masyarakat dalam (virtual) jaringan pribadi dan jaringan publik (internet) dan juga dalam pengiriman fisik.

Hasil penelitian ini dapat menyajikan hasil kualitatif data pengukuran dan pemeriksaan teknis sesuai dengan prinsip-prinsip rekayasa perangkat lunak sehingga hasil aplikasi kelayakan rekayasa akuntabel.

Kata kunci: kriptografi, enkripsi, pergeseran biner

1. LATAR BELAKANG

Pada saat ini sistem komputer personal yang menggunakan *platform Microsoft Windows*™ sebagai mesin penggerak sistemnya oleh beberapa kalangan pengguna serius dianggap rentan terhadap *cyber-crime*. Apalagi bagi kalangan yang sangat berkepentingan terhadap keamanan sistem berkasnya. Argumentasi dari kalangan tersebut adalah karena sistem komputer personal yang menggunakan sistem operasi *Microsoft Windows*™ dapat digunakan dengan mudah oleh para pengguna yang lain. Bahkan sistem komputer personal yang telah dilengkapi dengan berbagai perangkat keamanan standar milik *Microsoft Windows*™ tidak dapat mencegah para *hacker*, *cracker* dan *attacker* untuk masuk dan memanfaatkan isi dari sistem komputer yang bersangkutan untuk kepentingan negatif.

Tingkat keamanan data justru semakin rentan ketika sistem komputer personal yang bersangkutan mulai bergabung pada sistem komputer jaringan berbasis *peer to peer* milik *Microsoft Windows*™. Pada sejumlah kasus, tingkat keamanan sistem berkas yang akan dilindungi akan berbanding terbalik dengan tingkat kepentingan atau nilai guna dari sistem berkas tersebut.

Sementara itu salah satu metode dalam mengamankan data adalah metode enkripsi terhadap sistem berkas. Dan salah satu metode enkripsi yang paling banyak digunakan oleh para pengembang saat ini, termasuk metode enkripsi yang paling banyak disertifikasi adalah metode yang langsung bekerja pada aras rendah adalah metode biner.

Walaupun demikian, kelemahan dari metode-metode yang ada justru terletak dari

telah dipublikasikannya metode tersebut ke publik dan tentu saja setiap deskriptor *illegal* dengan serta merta segera meneliti dan memanfaatkan kelemahan ini untuk kepentingan mereka.

Oleh karena itu dari fakta yang telah dipaparkan di alinea-alinea sebelumnya, maka cukup relevan jika diputuskan untuk dilakukan penelitian dan pengembangan atas sebuah sistem enkripsi yang secara efisien dan efektif dapat digunakan oleh semua pengguna dalam memproteksi dan mengamankan datanya. Terutama data yang memiliki tingkat sensitifitas dan nilai guna tinggi.

1.1. Kebutuhan Perangkat Lunak Pengembangan

1. Bahasa Pemrograman

   a. Mampu digunakan untuk menulis program dengan metode kendali kejadian
   b. Memiliki kemampuan atau pernyataan untuk pengelolaan data terstruktur (basisdata) dan *flat*.

2. Perangkat Lunak Pengembang Program

   a. Merupakan lingkungan pengembang terpadu (*IDE*) dengan *interpreter* untuk pengujian kode, *debuger* untuk pengujian logik, *compiler* dan pembangun *program installer*
   b. Memiliki kemampuan pengelolaan grafis yang baik
   c. Dapat menggunakan atau memanfaatkan komponen sistem operasi secara langsung
   d. Mudah digunakan

2.2. Kebutuhan Perangkat Keras Pengembang

1. 1 unit PC untuk rekayasa komponen grafis baik untuk drawing, animation dan konverter format grafis ke video bertipe avi.
2. 2 unit PC untuk coding sesuai spesifikasi kebutuhan perangkat keras pada perangkat lunak pengambangan
3. 1 unit PC atau lebih (dengan spesifikasi berbeda) untuk pengujian

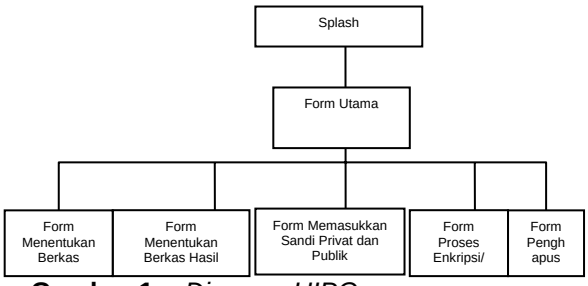
2.2.1. Kebutuhan Perangkat Keras Pengguna

PC dengan spesifikasi kebutuhan perangkat keras yang minimal mampu menjalankan program final.

2.3. KEBUTUHAN PROSES DAN ALGORITMA

2.3.1. Hirarchie Input Process Output

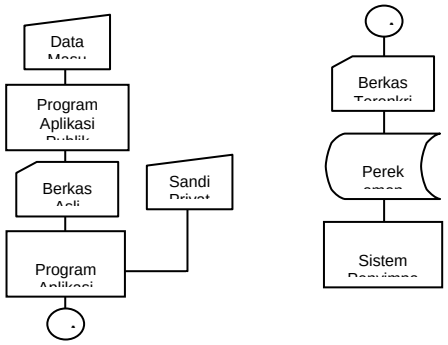
Secara global, sesuai dengan standar lama perihal HIPO, program dirancang beralur sebagai berikut :



Gambar 1. : *Diagram HIPO*

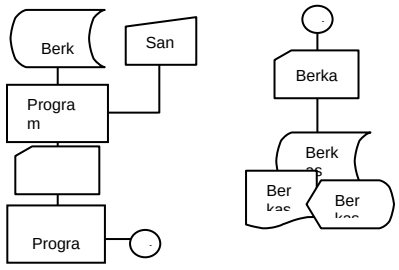
Ke dalam paket program ini ditambahkan program pendukung yang berupa modul panduan dan modul perihal. Masing-masing berupa sebuah form yang independent.

2.3.2. Spesifikasi Proses Enkripsi



Gambar 2. : *Spesifikasi Proses Enkripsi.*

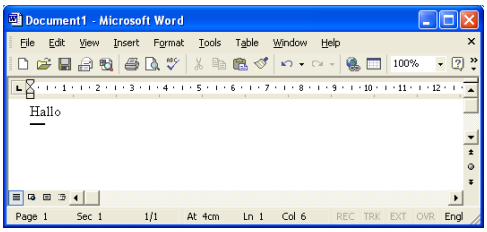
2.3.3. Spesifikasi Proses Dekripsi



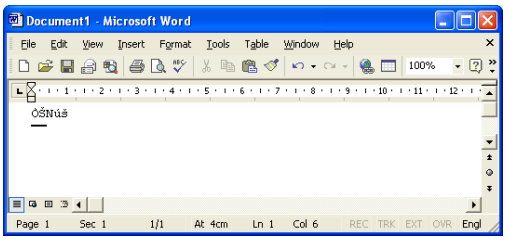
Gambar 3. : *Spesifikasi Proses Dekripsi.*

2.4. MODEL KELUARAN

Karena program yang akan dikembangkan adalah sebuah program yang dapat berfungsi dupleks (*vise-versa/back and fore*) maka keluaran yang diharapkan dari program ini adalah berkas terenkripsi dan berkas terdekripsi (format asli). Berikut adalah sebuah contoh berkas yang dibuat oleh program publik Microsoft Word dan contoh berkas yang telah di enkripsi.



Gambar 4. : *Contoh Original File*



Gambar 5. : *Contoh Chiper File*

Untuk berkas data tertutup (*psd, *.cdr, *.pdf, *.avi, dsb), berkas terenkripsi bahkan sudah tidak dapat lagi diakses oleh program aplikasi yang bersesuaian.

2.5. Rancang Bangun Basisdata

Program ini secara khusus dirancang untuk tidak menggunakan basis data dengan struktur tertentu. Semua berkas dianggap sebagai sebuah berkas basis data flat (dbf; database file) dengan tipe akses acak (Random database file) dan hanya memiliki satu byte field pada setiap recordnya. Hal ini tercermin dari pernyataan yang digunakan oleh penulis dalam membuka dan mengakses setiap berkas yang akan di enkripsi/deskripsi oleh penulis.

```
varBerkas = "C:\My Documents\Contoh.gif"
OPEN varBerkas FOR RANDOM AS #1 LEN = 1
GET #1, varCHAR
```

Dengan kata lain, program harus mampu melakukan manajemen sendiri atas koleksi datanya.

2.6. Rancang Bangun Program

2.6.1. Penetapan Lokasi Program Pada Sistem

Rancang bangun program yang disajikan penulis adalah program bebas (*Independent Module*) atau yang lazim disebut sebagai program bantu (*Utility Program*) jadi penggunaannya tidak terikat oleh satu sistem informasi tertentu. Pengguna dapat memanfaatkan program ini kapan saja dan dalam kapasitas apapun. Oleh karena itu langkah penetapan lokasi program pada sistem dapat dilewatkan.

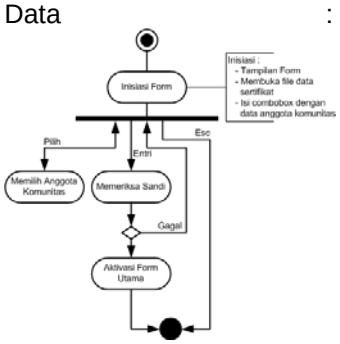
2.6.2. Spesifikasi Proses Terinci

Program dikembangkan sesuai dengan struktur yang secara deskriptif dijelaskan pada uraian berikut ini :

a. Splash Screen - Splash.frm

Fungsi : Splash, Verifikasi Sertifikat, Authentikasi Pengguna

Indeks : 0
Nama Form : Splash.frm (Human.skn)
Tipe : Borderless, illuminasi/transparan
Komponen : Combobox pengguna pada sertifikat, textbox sandi publik/privat
Data : Sertifikat

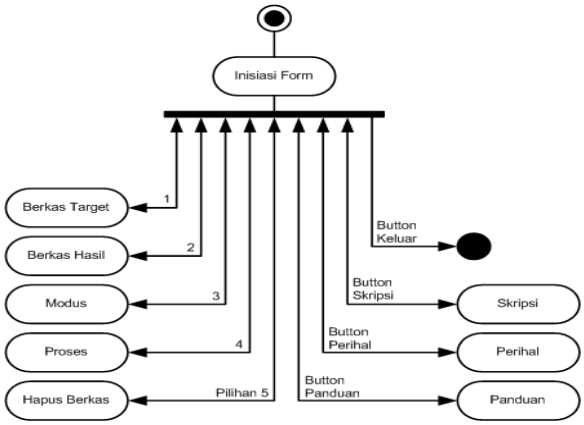


Gambar 6. : *Activity Diagram Splash Screen - Splash.frm*

b. Jendela Utama/Penyaji Utilitas - Utama.frm

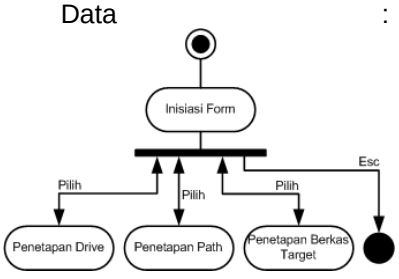
Fungsi : Penyaji Utilitas.
Indeks : 1
Nama Form : Utama.frm (FormUtama.skn)

Tipe :
Borderless, illuminasi/transparan
Komponen :
Button Utilitas berkas target, berkas hasil, modus, proses dan hapus berkas
Button panduan, perihal dan skripsi
Button untuk keluar
Data : -



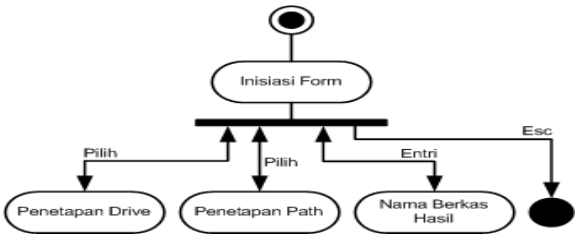
Gambar 7. : *Activity Diagram Jendela Utama/Penyaji Utilitas – Utama.frm*

c. Jendela Anak 1 - BerkasAsli.frm
Fungsi : Browsing berkas target
Indeks : 2
Nama Form : Berkas Asli.frm
Tipe : Borderless
Komponen : DriveListBox, DirListBox, FileListBox, Close button.
Data : Related File



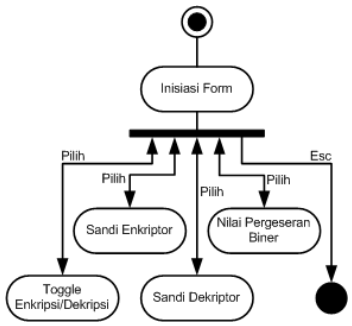
Gambar 8. : *Activity Diagram Jendela Anak 1 - BerkasAsli.frm*

d. Jendela Anak 2 - BerkasHasil.frm
Fungsi : Menetapkan drive dan path serta memasukkan nama berkas hasil.
Indeks : 3
Nama Form : Berkas Hasil.frm
Tipe : Borderless
Komponen : DriveListBox, DirListBox, FileListBox, Close button.
Data : Related File



Gambar 9. : *Activity Diagram Jendela Anak 2 - BerkasHasil.frm*

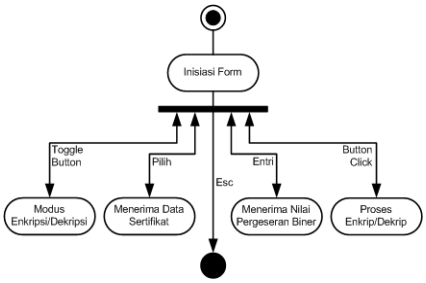
e. Jendela Anak 3 - Sandi.frm
Fungsi :
Memasukkan sandi privat/publik untuk kepentingan enkripsi/dekripsi dan sekaligus memasukkan nilai pergeseran modus enkripsi/dekripsi
Indeks : 4
Nama Form : Sandi.frm
Tipe : Borderless
Komponen :
Toggle button untuk memilih modus, textbox pergeseran nilai biner, Combobox ID enkriptor/ dekriptor pada sertifikat
Data : Sertifikat, Related File



Gambar 10. : *Activity Diagram Jendela Anak 3 - Sandi.frm*

f. Jendela Anak 4 - Proses.frm

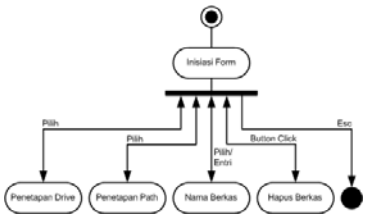
Fungsi	:	Melaksanakan proses enkripsi atau dekripsi berdasarkan data yang telah ditetapkan pada modul-modul anak yang lain
Indeks	:	5
Nama Form	:	Proses.frm
Tipe	:	Borderless
Komponen	:	Toggle button pemilih modus, combobox ID Enkriptor dan Dekriptor, textbox nilai pergeseran biner, command button eksekusi proses enkripsi/dekripsi
Data		Sertifikat, Related File



Gambar 11. : *Activity Diagram Jendela Anak 4 - Proses.frm*

g. Jendela Anak 5 - HapusBerkas.frm

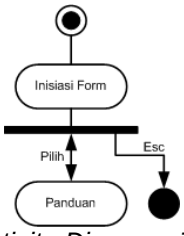
Fungsi	:	Utilitas penghapus berkas
Indeks	:	6
Nama Form	:	Hapus.frm
Tipe	:	Borderless
Komponen	:	DriveListBox, DirListBox, FileListBox, proses button
Data	:	Related File



Gambar 12. : *Activity Diagram Jendela Anak 5 - HapusBerkas.frm*

h. Jendela Anak 6 - Panduan.frm

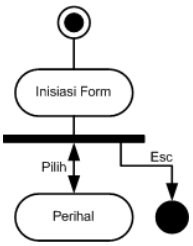
Fungsi	:	Utilitas panduan
Indeks	:	7
Nama Form	:	Panduan.frm
Tipe	:	3D Border, OS Standard
Komponen	:	RTF Viewer
Data	:	Related File



Gambar 13. : *Activity Diagram Jendela Anak 6 - Panduan.frm*

i. Jendela Anak 7 - Perihal.frm

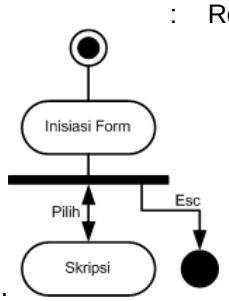
Fungsi	:	Utilitas perihal
Indeks	:	8
Nama Form	:	Perihal.frm
Tipe	:	3D Border, OS Standard
Komponen	:	SWF Player
Data	:	Related File



Gambar 14. : *Activity Diagram Jendela Anak 7 - Perihal.frm*

j. Jendela Anak 8 - Skripsi.frm

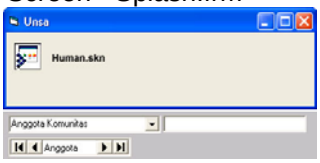
Fungsi	:	Utilitas penaja skripsi
Indeks	:	9
Nama Form	:	Skripsi.frm
Tipe	:	3D Border, OS Standard
Komponen	:	Microsoft Word Document Viewer
Data	:	Related File



Gambar 15. : *Activity Diagram Jendela Anak 8 - Skripsi.frm*

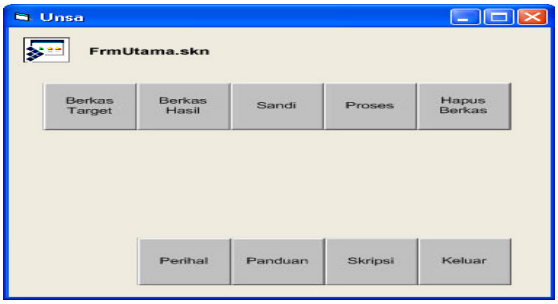
2.6.3. Perancangan Antarmuka

a. Splash Screen - Splash.frm



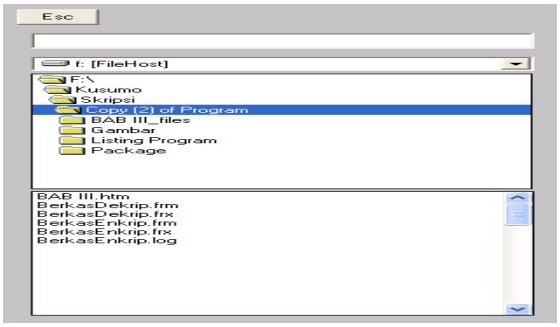
Gambar 16. : *UID SplashScreen – Splash.frm*

b. Jendela Utama/Penyaji Utilitas - Utama.frm



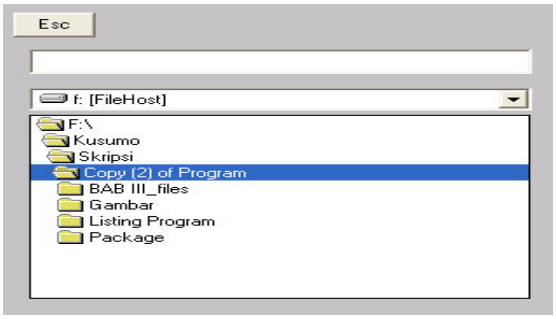
Gambar 17. : *UID Jendela Utama/Penyaji Utilitas – Utama.frm*

c. Jendela Anak 1 - BerkasAsli.frm



Gambar 18. : *UID Jendela Anak 1 - BerkasAsli.frm*

d. Jendela Anak 2 - BerkasHasil.frm



Gambar 19. : *UID Jendela Anak 2 - BerkasHasil.frm*

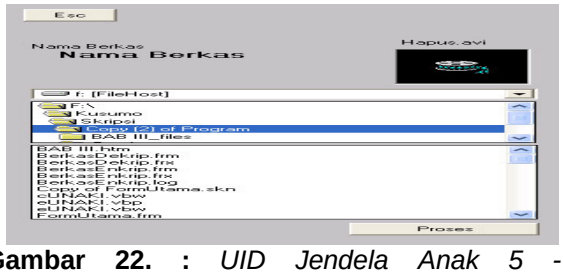
e. Jendela Anak 3 - Sandi.frm

Gambar 20. : *UID Jendela Anak 3 - Sandi.frm*

f. Jendela Anak 4 - Proses.frm

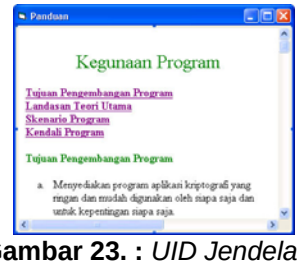
Gambar 21. : *UID Jendela Anak 4 - Proses.frm*

g. Jendela Anak 5 - HapusBerkas.frm



Gambar 22. : *UID Jendela Anak 5 - HapusBerkas.frm*

h. Jendela Anak 6 - Panduan.frm

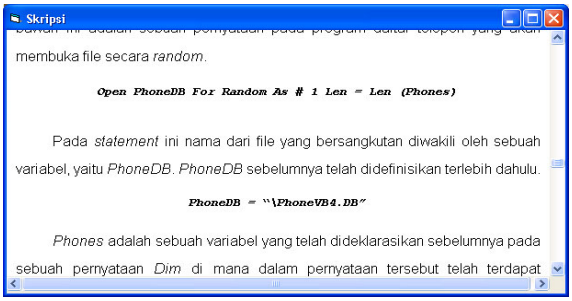


Gambar 23. : *UID Jendela Anak 6 - Panduan.frm*

i. Jendela Anak 7 - Perihal.frm

Gambar 24. : *UID Jendela Anak 7 - Perihal.frm*

j. Jendela Anak 8 - Skripsi.frm



Gambar 25. : *UID Jendela Anak 8 - Skripsi.frm*

3.1. Perangkat Lunak Pengembangan

Agar memenuhi kehendak hasil analisis kebutuhan maka untuk melakukan coding setiap modul yang terlibat pada program ini, penulis menggunakan IDE (*Integrated Development Environment*) bagian dari paket program Microsoft Visual Studio yang bertajuk ***Microsoft Visual Basic 6.0*.**

Secara keseluruhan, perangkat lunak pengembangan yang terlibat dalam kegiatan pengembangan program adalah :

1. Sistem Operasi Win 9x / Windows Xp
2. ActiveSkin Versi 4.25
3. Adobe PhotoShop Versi 7.0
4. Macromedia Flash 6.0
5. Microsoft Visual Basic 6.0

3.2. Perangkat Keras Pengembang

1 unit PC dengan konfigurasi sebagai berikut :

a. Processor Intel Pentium IV 1.7 Ghertz
b. Memory 256 Mbyte
c. Monitor SVGA dengan Resolusi 1024 x 768
d. Harddisk dengan kapasitas 40 Gbyte

2 unit PC untuk coding sesuai spesifikasi kebutuhan perangkat keras pada perangkat lunak pengembangan dengan konfigurasi sebagai berikut :

a. Processor Intel Pentium III 800 Mhz
b. Memory 128 Mbyte
c. Monitor SVGA dengan Resolusi 1024 x 768
d. Harddisk dengan kapasitas 40 Gbyte

1 unit PC atau lebih (dengan spesifikasi berbeda) untuk pengujian. Dalam hal ini penulis mensyaratkan pada responden pengujian untuk menggunakan PC setidaknya sama dengan spesifikasi PC yang digunakan untuk pengembangan (*coding*)

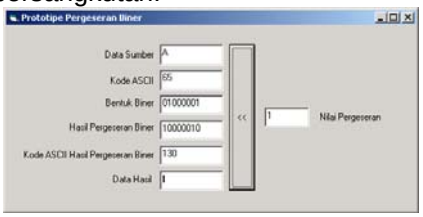
3.3. Perangkat Keras Pengguna

Perangkat keras yang dibutuhkan dalam pengimplementasian metode enkripsi pergeseran biner dengan Visual Basic 6.0 adalah sebagai berikut :

a. Prosessor Intel Pentium 233 MMX.
b. Memory 96 megabyte.
c. Monitor SVGA dengan Resolusi 800 x 600.
d. Mass Storage min 1 Gb.

3.4. Pembuatan Prototipe

Sebelum seluruh implementasi dilakukan, terlebih dahulu dibuat sebuah aplikasi kecil sebagai alat uji coba atau pembuktian teori pergeseran biner. Aplikasi yang dibuat sebagai prototipe tidak dirancang untuk melakukan proses balik (dekripsi). Berikut adalah tampilan aplikasi dan code list yang bersangkutan.



3.5. Pembuatan Antarmuka Aplikasi

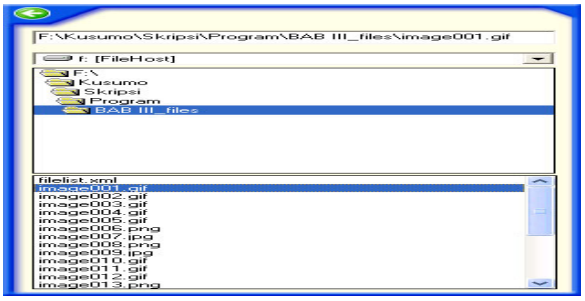
k. Splash Screen - Splash.frm

Gambar 26. : *UI SplashScreen – Splash.frm*

l. Jendela Utama/Penyaji Utilitas - Utama.frm

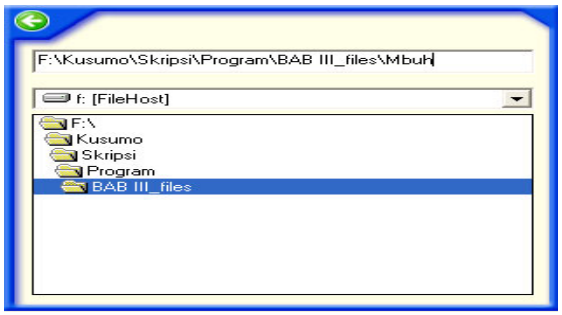
Gambar 27. : *UI Jendela Utama/Penyaji Utilitas – Utama.frm*

m. Jendela Anak 1 - BerkasAsli.frm



Gambar 28. : *UI Jendela Anak 1 - BerkasAsli.frm*

n. Jendela Anak 2 - BerkasHasil.frm



Gambar 29. : *UI Jendela Anak 2 - BerkasHasil.frm*

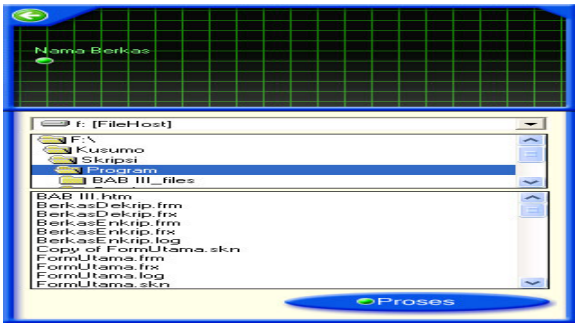
o. Jendela Anak 3 - Sandi.frm

Gambar 29. : *UI Jendela Anak 3 - Sandi.frm*

p. Jendela Anak 4 - Proses.frm

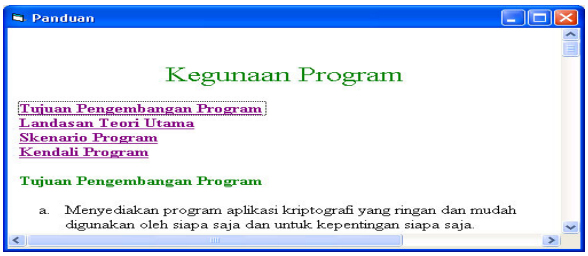
Gambar 30. : *UI Jendela Anak 4 - Proses.frm*

q. Jendela Anak 5 - HapusBerkas.frm



Gambar 31. : *UI Jendela Anak 5 - HapusBerkas.frm*

r. Jendela Anak 6 - Panduan.frm

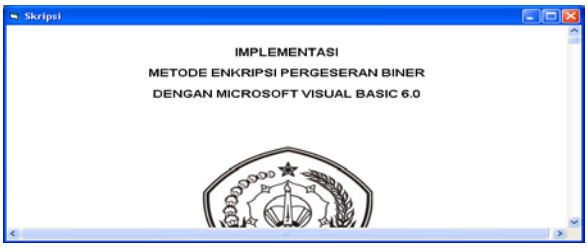


Gambar 32. : *UI Jendela Anak 6 - Panduan.frm*

s. Jendela Anak 7 - Perihal.frm

Gambar 33. : *UI Jendela Anak 7 - Perihal.frm*

t. Jendela Anak 8 - Skripsi.frm



Gambar 34. : *UI Jendela Anak 8 - Skripsi.frm*

3.6. Pengujian

Pengujian atau evaluasi terhadap aplikasi ini merupakan tahapan akhir yang tidak dapat diabaikan begitu saja. Keberhasilan aplikasi ini

sangat tergantung dari keberhasilan dari pengujian yang dilakukan. Karena pengujian merupakan suatu kejadian untuk mengidentifikasikan keberhasilan kelengkapan, keamanan dan kualitas pada aplikasi yang bersangkutan (Joko Nurjadi – PC Media, 2006). Berikut adalah jenis-jenis testing yang dilakukan atas aplikasi yang dihasilkan : (1) *Crash* (2) *Anomali* (3) *Fault* dan (4) *Mistake*. Sedangkan tahapan pengujian dimulai dari (1) *Unit/Component Testing* (2) *Integration Testing* (3) *System Testing* (4) *Acceptance Testing* dan (5) *Regression Testing*.

3.6.1. Unit/Component Testing

Unit/Component Testing sebenarnya telah dilakukan secara otomatis pada saat pengembangan aplikasi karena perangkat pengembangan yang digunakan merupakan sebuah IDE (*Integrated Development Environment*) yang sangat terpadu. Yaitu Microsoft Visual Basic versi ke 6.0. Oleh karena Tahap ini tidak dilaporkan kecuali jika tahap selanjutnya ternyata ditemui adanya *bug*.

3.6.2. Integration Testing

Pengujian dilakukan terhadap fungsi-fungsi sesuai *SRS* dan dengan modus penilai ber-skala 5.

Tabel 4.1. : *Hasil Integration Testing*

No	Responden	Uji1	Uji 2	Uji3	Uji 4
1.	Muhammad Ibrahim	4,00	3,00	4,00	3,80
2.	Fajar Widodo	3,90	3,20	4,00	2,90
3.	Joko Wandyatmono	3,80	3,20	3,80	3,10
4.	Yudi Wahyudi	4,00	3,50	4,00	3,50
5.	Erik Yuli	4,00	3,40	3,80	3,80
	Hasil Rata-Rata Uji	3,94	3,26	3,92	3,42

Uji 1. Dapat melakukan proses enkripsi dengan metode pergeseran biner dengan nilai pergeseran yang bersifat variable atau dapat dirubah oleh pengguna

Uji 2. Dapat digunakan untuk komunitas tertutup (eksklusif)

Uji 3. Menggunakan metode sertifikasi untuk proteksi berkas

Uji 4. Memiliki perangkat untuk manajemen berkas

3.6.3. System Testing

Sebenarnya pengujian harus pula dilakukan terhadap Sistem Operasi akan tetapi dengan asumsi bahwa Sistem Operasi telah teruji sebelumnya oleh pihak *vendor* maka pengujian hanya dilakukan terhadap komponen-komponen yang berupa *Dynamic Link Library* dan *ActiveX* kecuali yang standard (bawaan Sistem Operasi). Yang *intrinsic* (standard) maupun yang berupa *plug-in*. Pengujian dilakukan dilakukan oleh tim yang terdiri dari para responden *Integration Test.*Hasilnya terlihat pada tabel berikut ini.

Tabel 4.2. : *Hasil System Testing*

Komponen	Hasil Uji
ActiveSkin 4.0 Type Library	<OK>
CDlg/Common Dialog	<OK>
Microsoft HTML Object Library	<NA>
Microsoft Windows Common Control s 5.0 (SP2)	<OK>
Microsoft Windows Common Control s – 2 5.0 (SP2)	<OK>
Shockwave Flash ActiveX Component	<NA>

3.6.4. Acceptance Testing

Acceptance Testing dilakukan dengan responden yang sama pada saat *Integration Testing*.

Tabel 4.3. : *Hasil Acceptance Testing*

Responden	Keluwesan	Kemudahan	Muatan	Tampilan	Manfaat
Ibrahim	3,00	3,20	3,80	3,90	3,50
F. Widodo	3,60	3,30	3,90	4,00	3,50
Joko W.	3,50	3,40	3,60	3,30	3,50
Yudi W.	3,20	3,70	3,70	3,80	3,80
Erik Yuli	3,20	3,50	3,80	4,00	3,80
Rata-Rata Uji	3,30	3,42	3,76	3,80	3,62

3.6.5. Regression Testing

Karena metode yang digunakan adalah metode *Waterfall* maka untuk setiap tahapan pengujian dilakukan berkali-kali dan hasil pengujian yang diterakan pada masing-masing tahapan adalah resultan dari setiap pengujian yang dilakukan. Sehingga dari fakta tersebut maka secara teoritis pengujian pada tahapan ini tidak perlu dilakukan.

Hanya saja, pada *System Testing* untuk komponen uji 3 (Microsoft HTML Object Library) dan 5 (Shockwave Flash ActiveX Component) yang semula masih bernilai NA (*not available*) telah diperbaiki dan telah bernilai OK.

4. KESIMPULAN

1. Dalam implementasi metode enkripsi pergeseran biner dengan visual basic 6.0

penulis hanya menggunakan metode sederhana yaitu dengan menggeser per bit karakter yang dirotasikan dari arah kanan ke kiri. Akan tetapi walau hanya sederhana berdasarkan hasil pengujian jika ada pihak yang mencoba untuk mencoba informasi yang telah dienkripsi akan membutuhkan waktu yang lama karena pengambilan karakter dilakukan satu persatu sehingga proses enkripsi dilakukan dengan sangat teliti.

2. Walaupun metode enkripsi yang gunakan sangat sederhana tetapi dapat digunakan sebagaimana program lain yang lebih canggih. Variasi enkripsi dapat dilakukan dengan hanya merekonstruksi formula pergeseran biner yang penulis gunakan dengan memperbesar jumlah bit yang dijadikan obyek pergeseran maupun dengan memperbanyak jumlah variasi formulasi pergeseran.

5. SARAN-SARAN

1. Upaya keamanan apapun yang dilakukan oleh berbagai pihak sebetulnya hanya solusi sementara pada satu satuan waktu tertentu. Metode keamanan harus selalu berkembang, variasi pada formulasi enkripsi sebaiknya terus dikembangkan
2. Program ini sebaiknya dikembangkan dengan memberikan fasilitas yang berguna untuk memilih metode enkripsi apa yang akan digunakan oleh pengguna dan atau memasukan metode enkripsi dengan formulasinya sendiri. Oleh karena itu disarankan kepada publik agar selalu mengembangkan ide-ide formulasi enkripsi. Dari yang canggih (rumit) sampai dengan yang paling sederhana, karena yang penting adalah perubahan dari perkembangan itu sendiri.

DAFTAR PUSTAKA

[1] **ANDI YOGYA,** Wahana Komputer, *Memahami Model Enkripsi & Security Data* Semarang, 2003.

[2] **KARTONO. 1999,** *Metodologi Riset,* Jakarta : Penerbit Elek Media Komputindo, Jakarta, 1999.

[3] **PAMUNGKAS, IR,** "*Trip & Trik Microsoft Visual Basic 6.0*", Elek Media Komputindo, Jakarta, 2003.