

# **Aplikasi Enkripsi Tanda Tangan Di PT.BPR Daya Lambung Asia**

**Radiant Victor Imbar<sup>1)</sup> dan Wendi Chandra<sup>2)</sup>**

Jurusan Sistem Informasi, Fakultas Teknologi Informasi

Universitas Kristen Maranatha, Bandung

Jl. Prof. Drg. Suria Sumantri No. 65 Bandung 40164

Email : [radiant.vi@eng.maranatha.edu](mailto:radiant.vi@eng.maranatha.edu)<sup>1)</sup> , [wen9ball@gmail.com](mailto:wen9ball@gmail.com)<sup>2)</sup>

## *Abstract*

*Today, technology is growth very fast. Many advantage witches can get from the technology, which one is using Encrypted Application, that makes the company can store the data on database with more safely better than application which not using the Encrypted Application. The Encrypted Application can apply within company which have image file, especially .jpg, because file .jpg have a little space that more efficient and effective in space memory. By using the Encrypted Application such as Huffman method in this application, hopefully that can make the company work more efficient and effective in the safety problem.*

*Keywords: Encrypted Application, Image file, Efficient and effective*

## **1. Pendahuluan**

Dalam era globalisasi sekarang ini, maka kebutuhan akan teknologi informasi sudah tak terbendung lagi, mengingat kemajuan teknologi informasi itu sendiri sudah semakin meningkat seiring dengan perkembangan jaman. Seperti yang terjadi pada dunia perbankan di Indonesia pada masa sekarang ini penggunaan teknologi komputer dapat membuat tingkat keakuratannya lebih dapat dipercaya, sehingga kesalahan-kesalahan yang dibuat oleh manusia dapat diminimalisasi.

PT. BPR Daya Lambung Asia yang bergerak dalam bidang bank perkreditan rakyat ini telah memutuskan untuk menggunakan sistem komputerisasi untuk mempermudah pekerjaannya. Di samping itu, dengan adanya sistem komputerisasi di bank ini dapat membantu pula di dalam meningkatkan tingkat keamanan, seperti aplikasi metode enkripsi file tanda tangan yang dapat mencegah penyalahgunaan data nasabah di bank tersebut dari pihak-pihak yang tak bertanggung jawab, khususnya tanda tangan nasabah yang berfungsi sebagai identifikasi ketika nasabah tersebut akan melakukan segala transaksi pada bank ini, agar tidak disalahgunakan oleh pihak yang tak bertanggung jawab.

Aplikasi dengan metode enkripsi ini menggunakan metode pohon Huffman yang tidak hanya berfungsi untuk mengenkripsi file tanda tangan nasabah saja, tetapi juga dapat mengompresi file tersebut, sehingga file yang dihasilkan dari aplikasi ini menjadi lebih efisien di dalam pemakaian memori dan juga aman ketika disimpan di dalam database maupun untuk ditampilkan

## 2. Landasan Teori

### 2.1 Pengertian Sistem

Kata sistem berasal dari bahasa Latin yang disebut *systema* dan juga berasal dari bahasa Yunani yang disebut *sustema* yang berarti suatu kesatuan yang terdiri dari komponen atau elemen yang kemudian dihubungkan bersama untuk memudahkan aliran informasi, materi, ataupun energi (Trevor, 1999). Jadi, sistem dapat diartikan sebagai kumpulan dari komponen yang saling berkaitan untuk secara bersama-sama menghasilkan satu tujuan.

### 2.2. Pengertian Informasi

Informasi diartikan sebagai hasil pengolahan data yang digunakan untuk suatu keperluan, sehingga penerimanya akan mendapatkan suatu rangsangan untuk melakukan tindakan. Data adalah fakta yang jelas lingkup, tempat dan waktunya. Data diperoleh dari 2 sumber data, yaitu sumber data primer dan sumber data sekunder dalam bentuk berita tertulis ataupun sinyal elektronik. Pengertian informasi dan data berlaku sangatlah relative tergantung dari posisinya terhadap lingkup permasalahannya. Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat bagi pengambilan keputusan saat ini atau saat mendatang (Davis, 2003). Informasi merupakan kumpulan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerima. Dengan kata lain, Informasi adalah data yang telah diproses sehingga bentuknya berubah dan nilainya semakin tinggi.

Informasi merupakan data yang telah diproses atau data yang memiliki arti. Sedangkan data itu terdiri dari fakta-fakta dan angka-angka yang secara relative tidak berarti bagi pemakai (McLeod, 2001).

### 2.3. Pengantar Kriptografi

Di masa sekarang ini, sangatlah diperlukan sebuah mekanisme yang dapat membantu kita menyembunyikan data / informasi dari orang - orang yang tidak berhak untuk menerimanya. Salah satu proses tersebut ialah enkripsi dan dekripsi.. Enkripsi ialah proses merubah sekumpulan data (*message*) menjadi sebuah bentuk tertentu yang tidak dapat dimengerti oleh orang lain, sehingga dapat mencegah dari penyalahgunaan data dari orang yang tak bertanggung jawab yang dapat merugikan kita. Bentuk yang tidak dapat dimengerti oleh orang lain ini sering disebut sebagai *chiper text*. Sedangkan proses deskripsi ialah merubah *chiper text* menjadi **bentuk semula *message*** sebelum dienkripsi yang biasa disebut sebagai *plain text* (Munir, Rinaldi, 2008, Chap 1) .

Sedangkan kriptografi adalah suatu cara yang dapat dilakukan agar pengiriman suatu pesan dapat dilakukan dengan aman, contohnya adalah dengan menggunakan metode enkripsi yang sudah dijelaskan di atas.

Beberapa peran enkripsi antara lain :

1. memproteksi informasi yang disimpan dalam komputer dari akses yang tidak diizinkan (bahkan dari seseorang yang memiliki akses sekalipun).
2. memproteksi informasi yang ditransfer antar komputer dalam suatu jaringan.
3. menentukan dan mendeteksi usaha-usaha modifikasi disengaja terhadap data.

Terdapat 2 buah aplikasi kriptografi, yaitu pengiriman data melalui saluran komunikasi (*data encryption on motion*) dan juga penyimpanan data di dalam *disk storage* (*data encryption at rest*) (Munir, Rinaldi, 2008, Chap 2) .

*Data encryption on motion* merupakan sinyal yang ditransmisikan dalam percakapan dengan *handphone*, seperti juga nomor PIN kartu ATM yang ditransmisikan dari mesin ATM ke komputer bank.

Sedangkan *Data encryption at rest* berupa Dokumen teks, dokumen gambar, dan juga basis data. Berikut ini adalah contoh dari ketiga jenis *Data encryption at rest* tersebut :

#### 1. Dokumen teks

##### **Plainteks (plain.txt):**

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

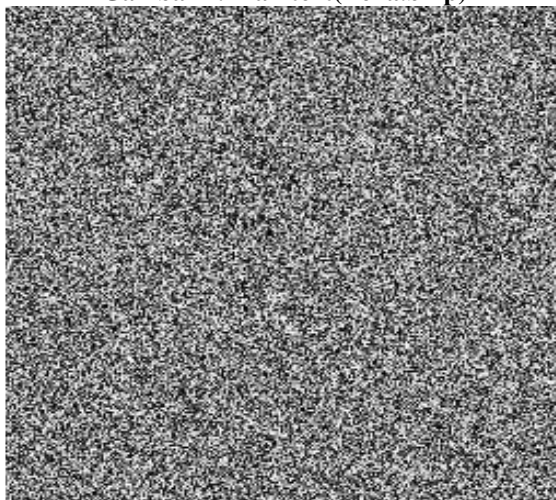
##### **Cipherteks (cipher.txt):**

Ztâxzp/épêp/qtüyp{p}<yp{p}/sx/□p}âpx;  
□□épêp/lt}tläzp}/qp}êpz/étzp{x/zt□xâx  
}v□□ép}v/ltüp}vzp/lt}äyâ/{pää=^tütz  
p□□psp{pw/p}pz<p}pz/zt□xâx}v/ép}  
v/qpüä□□lt}tâpé/spüx/sp{p/□péxü=/  
p{äüx□□lttüzp/lt}vpâpzp}/qpwâp/{pää  
/psp{pw□□âtl□pâ/ztwxsä□p}/ltützp=

#### 2. Dokumen Gambar



**Gambar 1. Plaintext(Lena.bmp)**



**Gambar 2. Ciphertext(Lena2.bmp)**

### 3. Basisdata

**Tabel 1. Plainteks (siswa.dbf)**

<b>NIM</b>	<b>Nama</b>	<b>Tinggi</b>	<b>Berat</b>
<b>000001</b>	<b>Elin Jamilah</b>	<b>160</b>	<b>50</b>
<b>000002</b>	<b>Fariz RM</b>	<b>157</b>	<b>49</b>
<b>000003</b>	<b>Taufik Hidayat</b>	<b>176</b>	<b>65</b>
<b>000004</b>	<b>Siti Nurhaliza</b>	<b>172</b>	<b>67</b>
<b>000005</b>	<b>Oma Irama</b>	<b>171</b>	<b>60</b>
<b>000006</b>	<b>Aziz Burhan</b>	<b>181</b>	<b>54</b>
<b>000007</b>	<b>Santi Nursanti</b>	<b>167</b>	<b>59</b>
<b>000008</b>	<b>Cut Yanti</b>	<b>169</b>	<b>61</b>
<b>000009</b>	<b>Ina Sabarina</b>	<b>171</b>	<b>62</b>

**Tabel 2. Cipherteks (siswa2.dbf)**

NIM	Nama	Tinggi	Berat
000001	tüp}vzpz/lt}äyâ/{ää	läzp}	épêp
000002	□□lt}tâpé/spüx/sp	péxü=	ztwxsä□
000003	□□âtl□pâ/ztwxsä□p}/l	}/ltü	spüx/
000004	épêp/lt}tläzp}/qpêpz	qp}êpz	wxsä
000005	étzp{x/zt□xâx}v□□êp}	pää/psp	étzp{
000006	spüx/sp{pl/□péxü=}	xâx}v	ttüzp/l
000007	Ztâxzp/épêp/qtüypp}<	äzp}	}äyâ/{
000008	qpwâp/{pää/psp{pw□	Ztwxs	xâx}v□□
000009	}tläzp}/qp}êpz/ép{	qp}êp	äzp}/qp

Keterangan: hanya *field* Nama, Berat, dan Tinggi yang dienkripsi.

#### 2.4. Algoritma pohon *Huffman*

Kode *Huffman* selalu memiliki atribut prefik yang unik, artinya bahwa setiap kode harus dapat didekodekan tanpa adanya kesalahan walaupun panjang kodenya bervariasi. Hal ini dimungkinkan karena adanya pohon *Huffman*. Pohon *Huffman* pada dasarnya merupakan pohon biner. Pada algoritma lain, seperti *Shannon-Fano* (di sini tidak dibahas), pembuatan pohon dimulai dari *root* lalu bergerak ke bawah menuju daun, sedangkan pada metode pohon *Huffman* dibuat mulai dari daun dan bergerak keatas menuju akar (*root*).

Adapun ketentuan membuat pohon *Huffman* adalah sebagai berikut :

1. Tentukan 2 buah *node* dengan jumlah kemunculan paling jarang.
2. Bentuk sebuah *node* induk dari kedua *node* tersebut. Probabilitasnya adalah jumlah dari kedua *node* tersebut.
3. *Node* induk ditambahkan ke dalam list *node* bebas, kemudian 2 *node* tersebut dihilangkan dari list.
4. *Node* dengan frekuensi kemunculan lebih tinggi akan dikodekan ke bit 0 dan *node* yang lainnya dikodekan ke bit 1.
5. Langkah sebelumnya diulangi hingga tersisa sebuah *node* bebas saja. *Node* bebas itu merupakan akar dari pohon.

Contoh : Terdapat data awal sebagai berikut :

Simbol A : memiliki jumlah kemunculan sebanyak 15 kali.

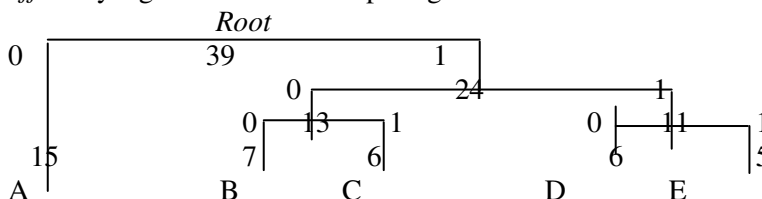
Simbol B : memiliki jumlah kemunculan sebanyak 7 kali.

Simbol C : memiliki jumlah kemunculan sebanyak 6 kali.

Simbol D : memiliki jumlah kemunculan sebanyak 6 kali.

Simbol E : memiliki jumlah kemunculan sebanyak 5 kali.

Maka pohon *Huffman* yang terbentuk akan seperti gambar di bawah ini :



**Gambar 3. Pohon hasil kompresi pohon *huffman***

Maka didapatkan kode hasil kompresi seperti pada tabel 3 di bawah ini :

**Tabel 3. Tabel kode hasil kompresi menggunakan *static Huffman***

Simbol	Kemunculan	Kode
A	15	0
B	7	100
C	6	101
D	6	110
E	5	111

**Tabel 4. Tabel jumlah bit yang dibutuhkan untuk menyimpan informasi dengan menggunakan algoritma pohon *Huffman***

Simbol	Kemunculan	Ukuran <i>Huffman</i>	Jumlah bit <i>Huffman</i>
A	15	1	1 * 15 = 15
B	7	3	3 * 7 = 21
C	6	3	3 * 6 = 18
D	6	3	3 * 6 = 18
E	5	3	3 * 5 = 15
<b>Total</b>	<b>39</b>		<b>87 bit</b>

Dari hasil tabel 4 dapat dilihat bahwa dengan *static Huffman* untuk menyimpan 85.25 bit informasi hanya dibutuhkan 87 bit saja. Hal ini lebih baik dibandingkan Shannon-Fano yang menyimpan dengan 89 bit.

Untuk mengetahui lebih lanjut cara perhitungan dengan metode *Shannon-Fano*, berikut penjelasannya:

Simbol	Kemunculan		
A	15	0	total bagian atas = 15 + 7 = 22
B	7	0	
pembagian pertama			
C	6	1	total bagian bawah = 6 + 6 + 5 = 17
D	6	1	
E	5	1	

**Gambar 4. Pembagian pertama pada Shannon-Fano**

Maka langkah selanjutnya adalah membagi kedua bagian yang terbentuk pada gambar 5. secara rekursif.

Simbol	Kemunculan		
A	15	0 0	Pembagian kedua
B	7	0 1	
Pembagian pertama			
C	6	1 0	Pembagian ketiga

D	6	1 1 0	Pembagian keempat
E	5	1 1 1	

**Gambar 5. Pembagian secara rekursif pada Shannon-Fano**

Secara teoretis perhitungan jumlah bit informasi adalah sebagai pada tabel 5 di bawah ini :

**Tabel 5. Tabel jumlah bit informasi.**

Simbol	Kemunculan	Jumlah bit informasi
A	15	$15 * (-^2\log(15/39)) = 20.68$
B	7	$7 * (-^2\log(7/39)) = 17.35$
C	6	$6 * (-^2\log(6/39)) = 16.20$
D	6	$6 * (-^2\log(6/39)) = 16.20$
E	5	$5 * (-^2\log(5/39)) = 14.82$
Total	39	85,25 bit

Jika *file* asal dikodekan berdasarkan standar ASCII, maka ukuran *file* asal adalah 312 bit (39 \* 8 bit). Jadi dapat dikatakan bahwa dalam 312 bit terdapat informasi 85.25 bit.

**Tabel 6. Tabel jumlah bit yang dibutuhkan untuk menyimpan informasi dengan menggunakan algoritma Shannon-Fano.**

Simbol	Kemunculan	Ukuran SF	Jumlah bit SF
A	15	2	$2 * 15 = 30$
B	7	2	$2 * 7 = 14$
C	6	2	$2 * 6 = 12$
D	6	3	$3 * 6 = 18$
E	5	3	$3 * 5 = 15$
Total	39		89 bit

Dari tabel 6 dapat diambil kesimpulan bahwa dengan standar ASCII untuk menyimpan informasi 85.25 bit dibutuhkan 312 bit, tetapi apabila digunakan algoritma Shannon-Fano untuk menyimpan 85.25 bit informasi dibutuhkan 89 bit.

### 3. Analisis dan Perancangan

#### 3.1 Identifikasi Kebutuhan Sistem

Pada bagian ini akan berisi tentang identifikasi dari aplikasi pengamanan data nasabah yang akan dikembangkan. Dilakukan dengan penelitian dari lingkungan-lingkungan sekitar. Di dalam tahap identifikasi ini diperoleh informasi yaitu :

1. Membuat bagaimana user( di dalam aplikasi ini yaitu *teller*) untuk memahami fungsi dari enkripsi dan juga seluruh proses yang ada di dalam aplikasi ini.
2. User yang bertindak sebagai *teller* hanya dapat melakukan pengecekan data nasabah (*view*) saja, sedangkan admin yang dapat melakukan perubahan data nasabah, seperti menambah data nasabah (*add*), mengedit data nasabah (*update*), menghapus data nasabah (*delete*).

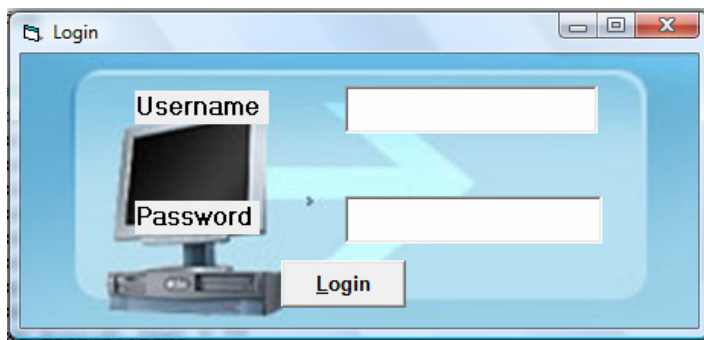
### 3.2 Overview Sistem

Pada bagian ini berisi tentang sistem-sistem yang akan dikembangkan untuk aplikasi enkripsi data nasabah ini setelah dilakukan identifikasi kebutuhan sistem adalah sebagai berikut:

1. Aplikasi ini dapat mencari data-data nasabah berdasarkan nomor rekening nasabah.
2. Aplikasi ini dapat dengan cepat melakukan pengamanan data berupa enkripsi pada saat data nasabah sudah terisi dan tanda tangan nasabah sudah berbentuk file gambar(.jpg), kemudian keseluruhan data tersebut disimpan ke dalam database.
3. Terdapat fungsi agar file tanda tangan nasabah yang sudah terenkripsi tersebut dapat dilihat dengan melakukan dekripsi data pada menu *view* di dalam aplikasi tersebut, dan proses dekripsi data hanya dapat dilakukan di dalam aplikasi ini, sehingga tingkat keamanan data nasabah pun lebih tinggi.
4. Aplikasi ini juga memiliki fungsi melihat proses apa saja yang terjadi selama ada kegiatan yang dilakukan di dalam aplikasi ini, dan semua data tersebut dapat dilihat di dalam menu *history*.

### 3.3 Analisa Proses Aplikasi

Berikut adalah gambar dari proses aplikasi enkripsi tandatangan ini :

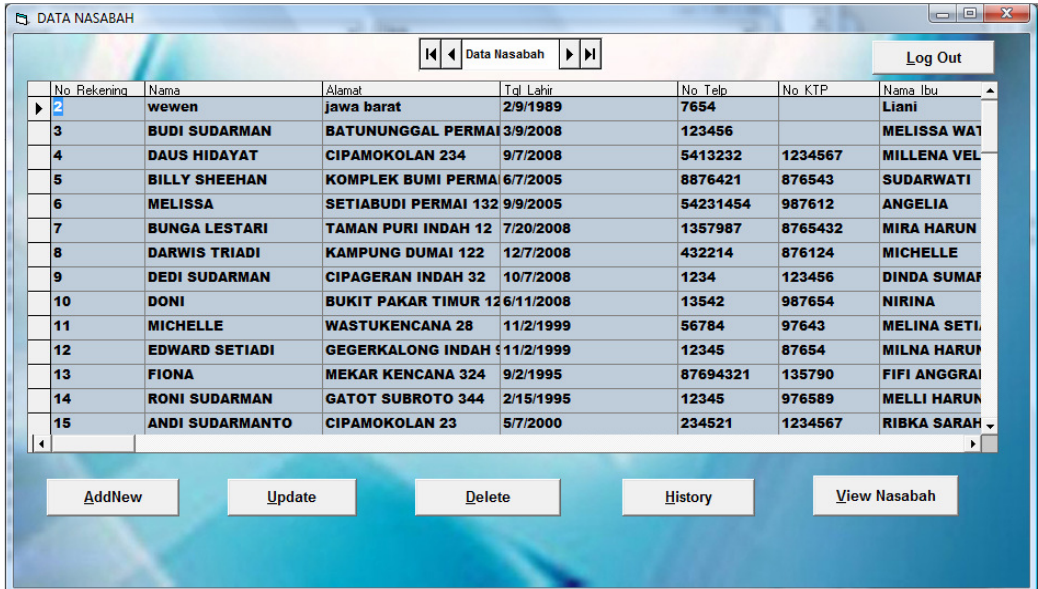


**Gambar 3. Form Login**

Ini adalah form login, dimana terdapat 2 *user*, yaitu administrator dan teller.

Jika kita memasuki aplikasi sebagai administrator, maka form utamanya adalah seperti ini :



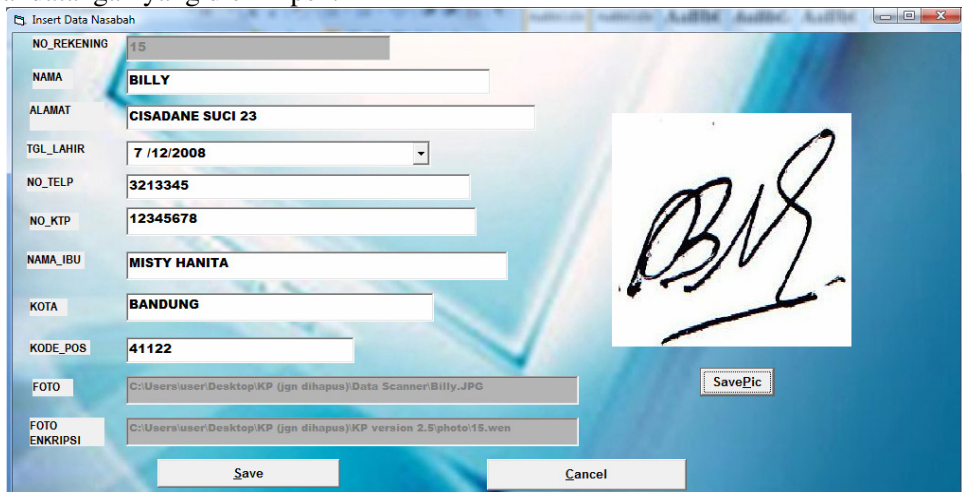


The screenshot shows a window titled "DATA NASABAH" with a table of customer records. The table has columns for No Rekening, Nama, Alamat, Tgl Lahir, No Telp, No KTP, and Nama Ibu. Below the table are buttons for "AddNew", "Update", "Delete", "History", and "View Nasabah".

No Rekening	Nama	Alamat	Tgl Lahir	No Telp	No KTP	Nama Ibu
2	wewen	jawa barat	2/9/1989	7654		Liani
3	BUDI SUDARMAN	BATUNUNGGAL PERMAI	3/9/2008	123456		MELISSA WAT
4	DAUS HIDAYAT	CIPAMOKOLAN 234	9/7/2008	5413232	1234567	MILLENA VEL
5	BILLY SHEEHAN	KOMPLEK BUMI PERMA	6/7/2005	8876421	876543	SUDARWATI
6	MELISSA	SETIABUDI PERMAI 132	9/9/2005	54231454	987612	ANGELIA
7	BUNGA LESTARI	TAMAN PURI INDAH 12	7/20/2008	1357987	8765432	MIRLA HARUN
8	DARWIS TRIADI	KAMPUNG DUMAI 122	12/7/2008	432214	876124	MICHELLE
9	DEDI SUDARMAN	CIPAGERAN INDAH 32	10/7/2008	1234	123456	DINDA SUMAR
10	DONI	BUKIT PAKAR TIMUR 12	6/11/2008	13542	987654	NIRINA
11	MICHELLE	WASTUKENCANA 28	11/2/1999	56784	97643	MELINA SETI
12	EDWARD SETIADI	GEGERKALONG INDAH	11/2/1999	12345	87654	MILNA HARUN
13	FIONA	MEKAR KENCANA 324	9/2/1995	87694321	135790	FIFI ANGGRAI
14	RONI SUDARMAN	GATOT SUBROTO 344	2/15/1995	12345	976589	MELLI HARUN
15	ANDI SUDARMANTO	CIPAMOKOLAN 23	5/7/2000	234521	1234567	RIBKA SARAH

Gambar 4. Form Administrator

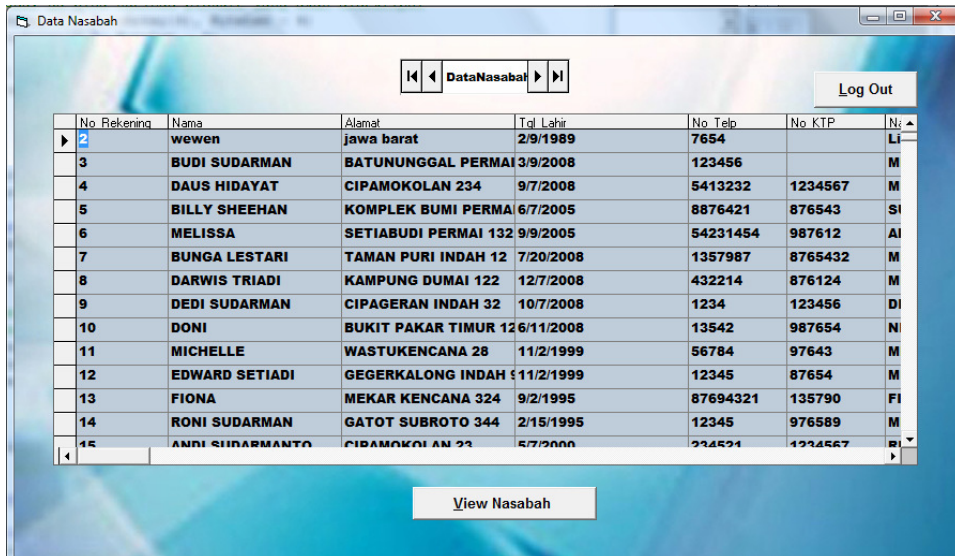
Di dalam form administrator, kita dapat membuat file nasabah baru serta file tandatangan yang dienkripsi :



The screenshot shows the "Insert Data Nasabah" form with the following fields filled out: NO\_REKENING: 15, NAMA: BILLY, ALAMAT: CISADANE SUCI 23, TGL\_LAHIR: 7 /12/2008, NO\_TELP: 3213345, NO\_KTP: 12345678, NAMA\_IBU: MISTY HANITA, KOTA: BANDUNG, KODE\_POS: 41122. There are two photo upload fields: FOTO and FOTO ENKRIPSI. A signature image is displayed on the right side of the form. Buttons for "Save" and "Cancel" are at the bottom.

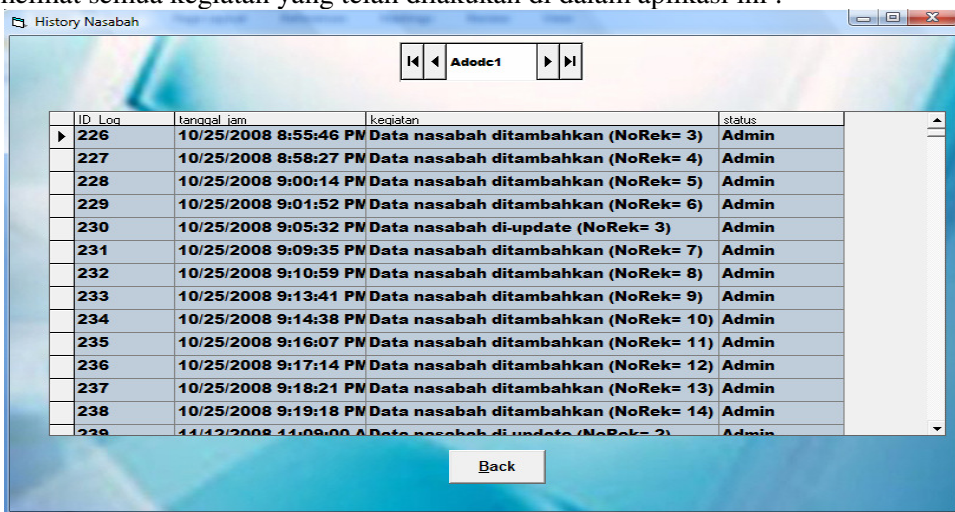
Gambar 5. Form Insert Data Nasabah

Berikut ini adalah gambar form jika kita memasuki aplikasi ini sebagai teller :



Gambar 6. Form utama Teller

Dan, berikut ini adalah form history, dimana di dalam form ini kita dapat melihat semua kegiatan yang telah dilakukan di dalam aplikasi ini :



Gambar 7. Form History

Aplikasi ini menekankan kepada metode enkripsi file tanda tangannya. Dan berikut adalah proses algoritma Huffman yang dipakai untuk melakukan proses enkripsi file tanda tangan :

1. Membuat pohon Huffman ( Huffman tree) dengan membuat node kiri, node kanan, node parent, frekuensi, dan juga nilai-nya(value).
2. Sesudah membuat pohon Huffman, langkah selanjutnya pemberian nilai(value) kepada setiap leaf-nya yang maksimal berukuran 255.

3. Setelah memberikan *value* kepada setiap *leaf*-nya, maka langkah selanjutnya adalah membandingkan antara node-node yang frekuensinya kemunculannya paling kecil.
4. Kemudian, jika sudah didapatkan 2 node yang kemunculannya paling kecil, maka 2 frekuensi tersebut dijumlahkan, dan 2 node tersebut digabung menjadi node *parent*, dan seterusnya sampai habis sampai membentuk sebuah pohon *Huffman*.
5. Setelah pohon *Huffman* terbentuk, maka kita bisa menghitung panjang *length* bitnya dengan menjumlahkan setiap bit yang sudah terbentuk dari setiap karakternya, dan menentukan *prefix*(sebagai identifikasi unik untuk setiap karakter-nya).
6. Dengan semua proses di atas, maka *file* tandatangan sudah terenkripsi. Sedangkan untuk proses dekripsi data, maka langkahnya adalah membalikkan kode dari *prefix* yang sudah dibuat pada proses enkripsi data-nya, sehingga *file* tandatangan tersebut sudah dapat terdekripsi.

Algoritma ini diterapkan di tombol *Save* ketika hendak menyimpan data nasabah baru. Jadi, prosesnya ketika tombol *Save* disimpan, data nasabah disimpan ke dalam *database*, dan file tanda tangan-nya dienkripsi dengan proses algoritma *Huffman* di atas. Begitu juga ketika melakukan *Update* data, file tanda tangan baru yang akan dienkripsi akan dienkripsi setelah user menekan tombol *Update*. Dengan algoritma pohon *Huffman*, memiliki kode *prefix* yang unik, sehingga tidak dapat disamakan dengan kode lainnya. Metode pohon *Huffman* sebetulnya merupakan metode untuk kompresi data, tetapi sekaligus dapat juga melakukan proses *enkripsi* data, sehingga pada akhir proses pengamanan data nasabah ini, *file* tanda tangan nasabah telah dikompresi sekaligus *enkripsi* juga.

#### **4. Kesimpulan dan Saran**

Dengan dibuatnya aplikasi enkripsi tanda tangan ini dapat menjawab beberapa kebutuhan yang ada di PT.BPR Daya Lumbang Asia dalam hal pengurusan penyimpanan data nasabah beserta dengan sistem keamanannya.

Adapun hal yang dapat terjawab dengan adanya aplikasi ini adalah:

1. Pengolahan data yang telah dienkripsi dapat lebih mudah ditentukan tempat penyimpanannya sesuai dengan kehendak kita sehingga lebih efisien dan dapat mencegah data tersebut dari perbuatan yang tidak bertanggung jawab.
2. Pembuatan file enkripsi tanda tangan yang mewajibkan file aslinya bertipe *.jpg* membuat aplikasi ini lebih efektif, karena membuat ruang penyimpanan data lebih efisien serta mampu menciptakan file enkripsi yang bertipe tertentu( di dalam aplikasi ini file enkripsi bertipe *.wen*). Untuk hasil kompresi *image*, tidak akan mempengaruhi perubahan pada *image* itu sendiri secara kasat mata, karena perubahan *pixel* hanya mempengaruhi sedikit saja *image* yang tertampil sesudah proses *enkripsi* terjadi.
3. Aplikasi ini lebih difokuskan untuk penyimpanan data yang lebih aman. Jadi, data yang sudah terenkripsi tersebut hanya dapat didekripsi di dalam aplikasi

ini saja, selain di dalam aplikasi ini tidak dapat melakukan proses dekripsi data. Dan seperti yang sudah dijelaskan sebelumnya bahwa aplikasi ini melakukan proses kompresi data sekaligus *enkripsi* data, sehingga ruang penyimpanan dapat lebih efisien sekaligus file tanda tangan nasabah pun aman.

4. Pengolahan data enkripsi dapat lebih efektif dengan dicantumkan no rekening nasabah yang berfungsi sebagai identifikasi dari setiap data yang telah dienkripsi, yang berarti nama dari *file* tanda tangan nasabah hasil *enkripsi* menjadi nomor rekening nasabah tersebut, karena nomor rekening merupakan *primary key*, sehingga tidak akan ada yang sama, sedangkan jika menggunakan nama, ataupun data lainnya, maka besar kemungkinan terdapat data yang sama, sehingga dapat membingungkan *user* untuk mengolah data nasabah.

Sebaiknya aplikasi ini dapat terus berkembang dengan adanya metode baru yang tentunya lebih efektif di dalam proses pengamanan data, karena metode pohon *Huffman* merupakan metode yang sudah sangat lama dipakai dan sudah cukup banyak dipakai oleh banyak orang untuk proses kompresi data sekaligus *enkripsi* data, sehingga dibutuhkan metode yang lebih baru dan yang lebih efektif digunakan di dalam proses pengamanan data.

## **V. Daftar Pustaka**

- [Cam07] Campos, A. (2007, November 24). *Introduction to Static Huffman*. Retrieved October 20, 2008, from <http://www.arturocampos.com/static-huffman>.
- [Dar07] Handoyo, Erico Darmawan.(2007).*Materi Perkuliahan Pemrosesan Citra Digital Teori*. Bandung : Universitas Kristen Maranatha.
- [Rin06] Munir, Rinaldi.(2006).*Materi Perkuliahan Kriptografi*. Bandung. Universitas Kristen Maranatha
- [Yun06] Supardi, Yuniar.(2006).*Microsoft Visual Basic 6.0 untuk Segala Tingkat*. Jakarta:PT Elex Media Komputindo.
- [Yus05] Yuswanto.(2005).*Pemrograman Client-Server Microsoft Visual Basic 6.0 Jilid 2*.Jakarta : Prestasi Pustakaraya.
- [Zak08] Zakaria, Teddy Marcus.(2008).*Materi Perkuliahan Basis Data Lanjut Praktikum*. Bandung : Universitas Kristen Maranatha.