BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi sistem

Setelah sistem dianalisis dan didesain secara rinci, kemudian dilakukan tahap implementasi. Implementas sistem merupakan tahap meletakkan sistem sehingga siap untuk dioperasikan. Implementasi bertujuan untuk mengkonfirmasi modul-modul perancangan, sehingga pengguna dapat memberi masukkan pada pengembangan sistem.

4.1.1 Lingkungan Implementasi

Untuk mendukung aplikasi yang akan diterapkan pada lingkungan implementasi, maka penulis menggunakan perangkat keras dan perangkat lunak yang menunjang dalam pengembangan sistem keamanan berbasis *smart card.*

4.1.1.1 Perangkat keras yang digunakan

Spesifikasi perangkat keras yang digunakan dalam pengembangan aplikasi ini adalah sebagai berikut :

Tabel 4.1 Spesifikasi perangkat keras yang digunakan

No	Komponen perangkat keras
1	Processor 2.14 GHz
2	RAM 3GB

3	HDD 250GB
4	Smart card reader ACS ACR38U
5	Smart card ACOS2,ACOS3, SLE4428

4.1.1.2 Perangkat lunak yang digunakan

Spesifikasi perangkat lunak yang digunakan dalam pengembangan aplikasi ini adalah sebagai berikut :

Tabel 4.2 Spesifikasi perangkat lunak yang digunakan

No	Perangkat lunak	Keterangan
1	Sistem Operasi	Windows Vista Home Premium SP1
2	IDE	Visual Studio 2008
3	Framework .NET	Versi 4.0
4	UML modeler	Microsoft Visio 2007

4.1.2 Batasan Implementasi

Pembatasan implementasi dimaksudkan agar ruang lingkup implementasi menjadi lebih jelas. Batasan implementasi dari sistem keamanan ini adalah sebagai berikut :

1. Perangkat lunak yang dikembangkan tidak berbasis client-server
2. Tidak menggunakan tipe kartu memory sebagai media penginputan data sehingga sistem hanya menggunakanya sebatas autentikasi admin.
3. Kartu hanya dapat diberi security sebanyak satu kali.
4. Sistem dikembangkan menggunakan bahasa C#.NET sebagai kekas.

5. Fokus implementasi adalah keamanan data yang diterapakan pada *smart card*, yaitu kriptografi dengan ElGamal dan algoritma Luhn (algoritma validasi sebagai solusi untuk menghindari *chosen-chipertext attack*).
6. Perangkat lunak merupakan *prototype*.

4.1.3 Implementasi Program

Pada tahapan ini akan dijelaskan mengenai implementasi antarmuka kriptografi ElGamal dan Algoritma Luhn pada sebuah aplikasi.

4.1.3.1 Implementasi kriptografi ElGamal

Berikut ini adalah implementasi kelas - kelas dengan menggunakan bahasa pemrograman C#.

Tabel 4.3 Implementasi Kelas – kelas

Kelas	Nama File
Kartu	**Kartu.cs**
FormUtama	**Form1.cs**
ReaderKartu	**ReaderKartu.cs**
OperasiFile	**OperasiFile.cs**
FastExponentiation	**FastExponentiation.cs**
ElGamal	**ElGamal.cs**
Ekripsi	**Enkripsi.cs**
Dekripsi	**Dekripsi.cs**
Luhn	**Luhn.cs**
CSPRNG_RSA	**CSPRNG_RSA.cs**

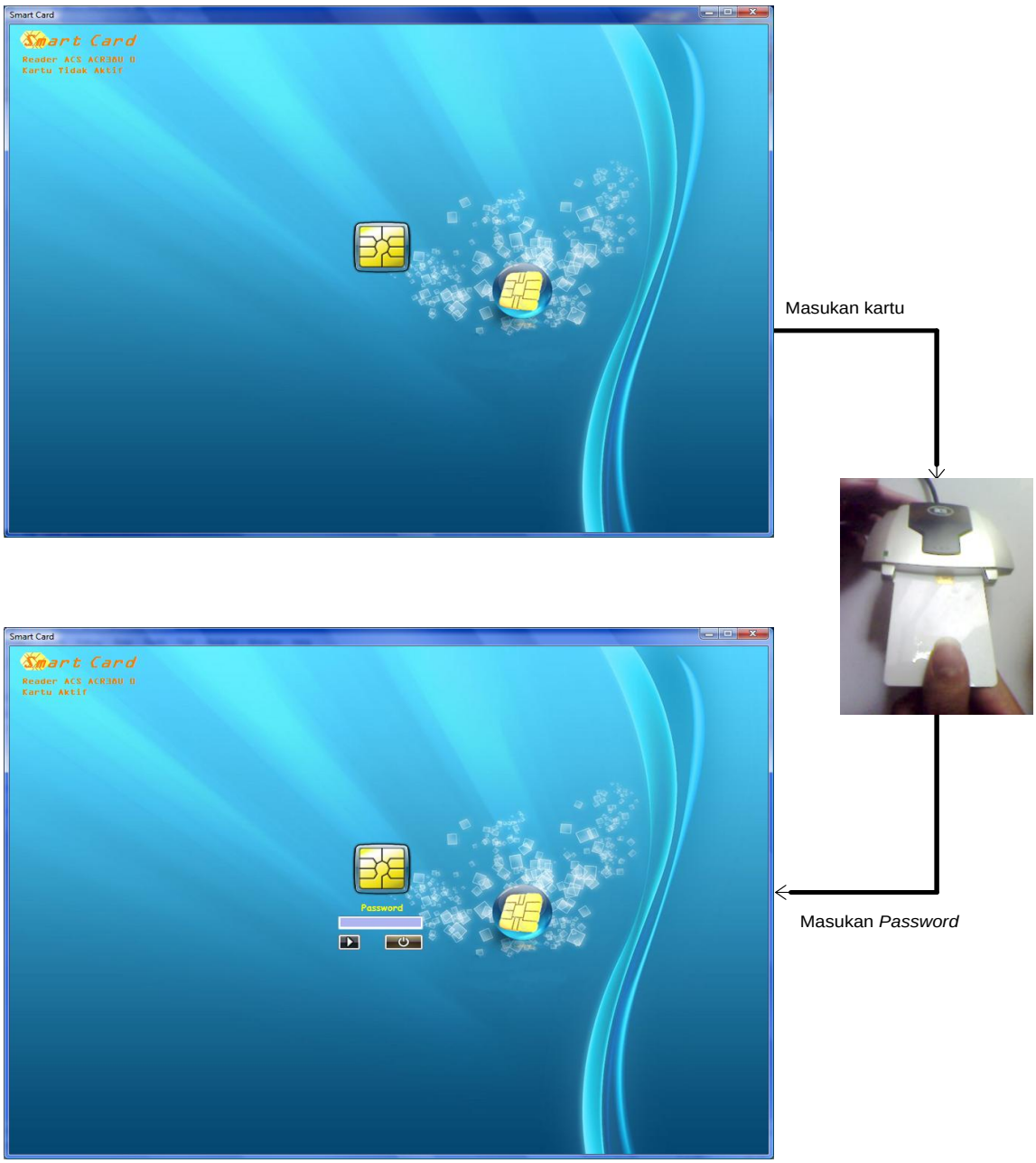
4.1.3.2 Implementasi sistem

Implementasi sistem merupakan hasil dari perancangan yang sebelumnya sudah dilakukan.

Antarmuka yang dibuat disesuaikan dengan alur kerja dan fungsi sesuai dengan perancangan sistem. Berikut adalah gambaran yang telah dibuat berdasarkan hasil perancangan.

4.1.3.2.1 Antarmuka Autentikasi

Pada antarmuka autentikasi, sistem akan menampilkan form utama untuk melakukan autentikasi. Dalam sekenario analisis sistem, admin dapat memasukan *password* jika *reader* mendeteksi keberadaan kartu. Sehingga admin dapat melakukan autentikasi hanya jika menggunakan *smart card.*

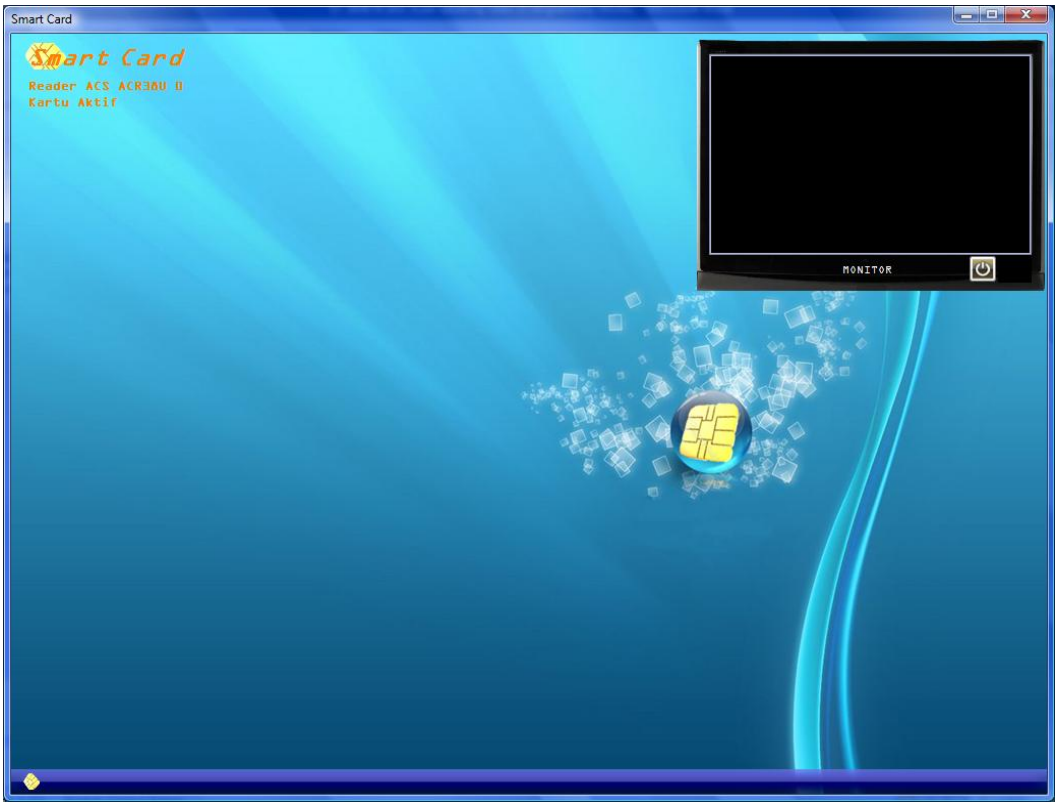


Gambar 4.1 Antarmuka dan proses autentikasi

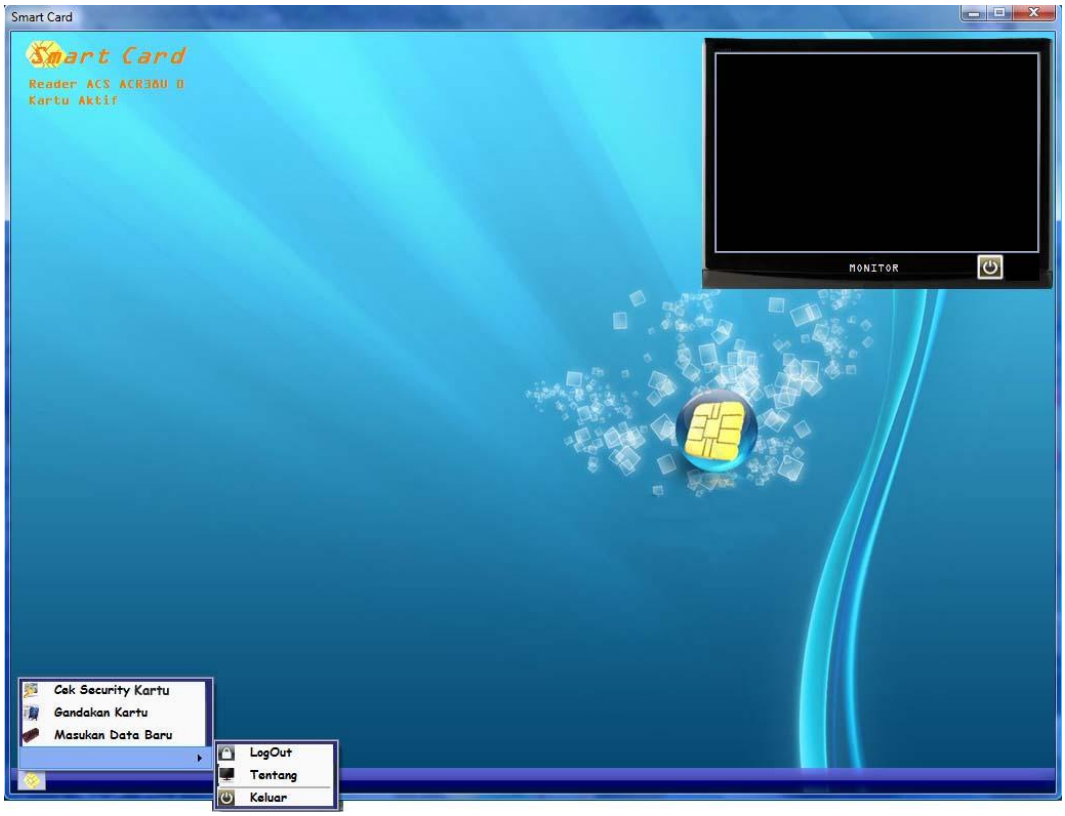
4.1.3.2.2 Antarmuka Menu Utama

Antarmuka Menu Utama terdapat pada form utama. Form utama terdiri dari beberapa komponen diantaranya seperti monitor yang fungsinya memonitor semua aksi terhadap kartu, *button* yang terdapat pada monitor berfungsi sebagai koneksi/diskoneksi antara *reader* ke kartu dan *button* menu

utama yang menyimpan menu utama dimana admin dapat memilih salah satu menu yang disajikan. Lebih jelasnya gambar dibawah ini menunjukan antarmuka form utama dan menu utama.



Gambar 4.2 Antarmuka Form Utama



Gambar 4.3 Tombol Menu Utama Diklik

Pada Menu Utama admin dapat membuka menu seperti Masukan Data Baru, Gandakan Kartu, Cek Security Kartu, Logout, Tentang dan Keluar.

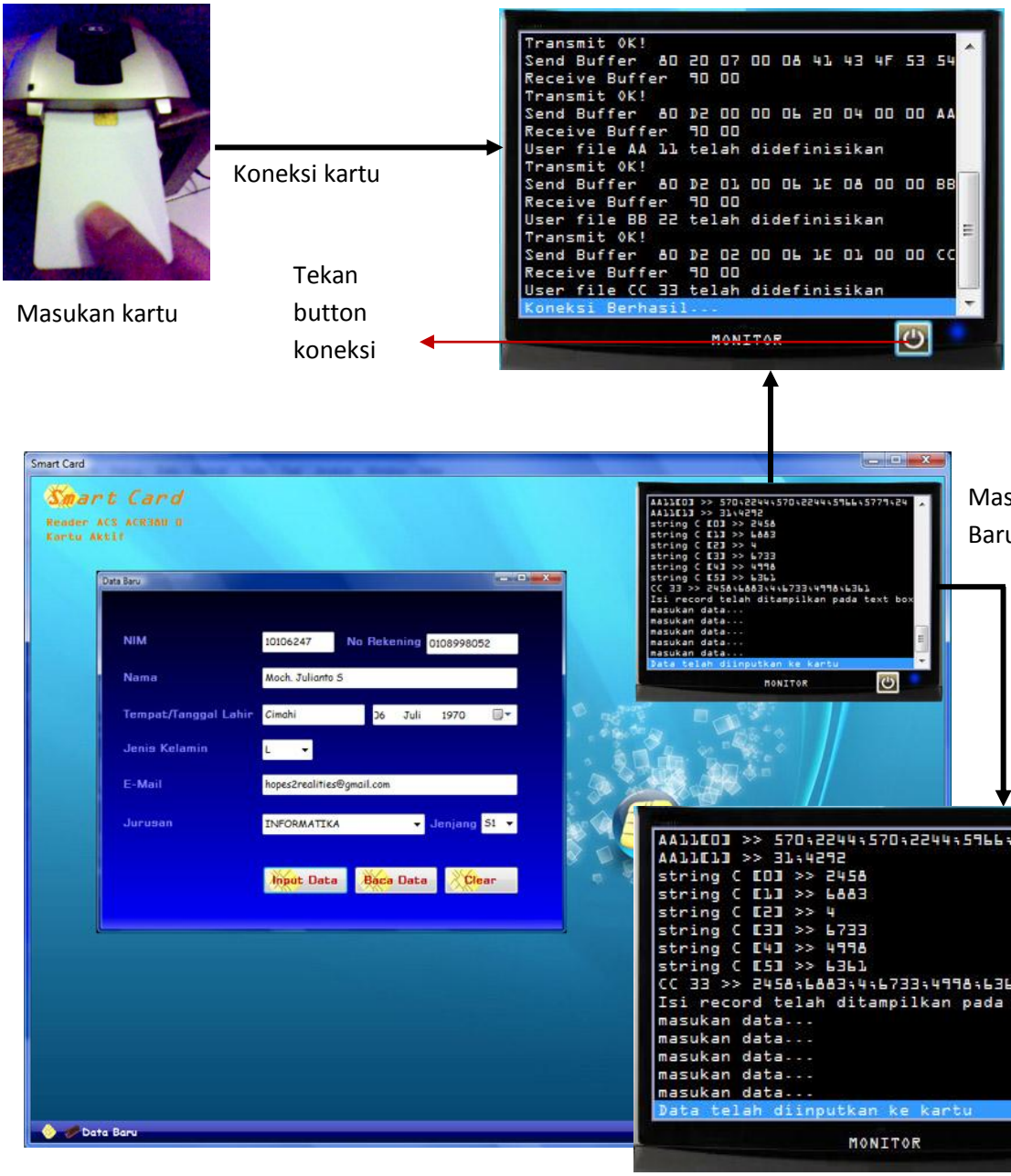
4.1.3.2.3 Masukan Data Baru

Pada antarmuka masukan data baru, form utama menampilkan panel dengan label “Data Baru” disini admin dapat menginputkan data ke kartu. Sebelum menginputkan data ke kartu admin terlebih dahulu memasukan kartu ke slot *smart card reader*. Setelah kartu terbaca oleh *reader* lakukan koneksi, baru kemudian admin dapat menginputkan data ke kartu. Hasil dari semua aksi proses input dan baca data ditampilkan di monitor.

Berikut langkah-langkah proses masukan data baru :

1. Masukan kartu ke slot *reader*.
2. Lakukan koneksi ke kartu dengan menekan *button* koneksi, selanjutnya monitor akan menampilkan status koneksi.
3. Tulis data yang akan dimasukan ke kartu.
4. Tekan *button* input data, selanjutnya monitor akan menampilkan status input data.

Lebih jelasnya, proses masukan data baru dapat dilihat pada gambar 4.4.



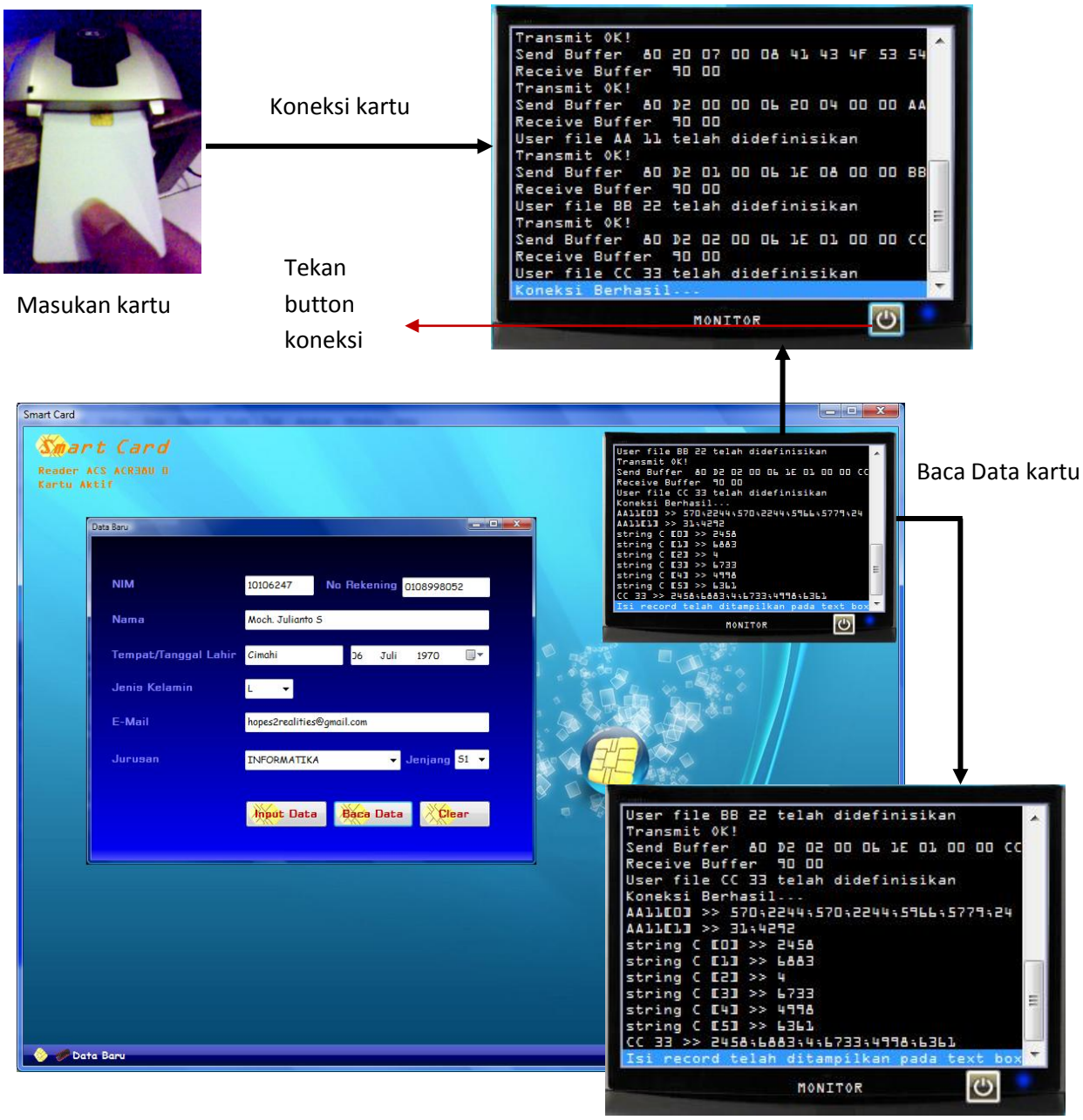
Gambar 4.4 Antarmuka dan aksi Masukan Data Baru

Aksi masukan data juga merupakan aksi enkrip data NIM. Terlihat dalam monitor baris 1 dan 2 untuk nama file “AA11[0]” dan “AA11[1]” terdapat *chipertext* yang merupakan hasil enkripsi NIM.

Selanjutnya untuk langkah-langkah proses baca data dapat dilihat sebagai berikut :

1. Masukan kartu ke slot *reader*.
2. Lakukan koneksi ke kartu dengan menekan *button* koneksi, selanjutnya monitor akan menampilkan status koneksi kartu.
3. Tekan *button* baca data, monitor akan menampilkan "Isi record telah ditampilkan pada *text box*", selanjutnya data akan ditampilkan.

Lebih jelasnya, proses baca data dapat dilihat pada gambar 4.5.



Gambar 4.5 Proses dan aksi Baca Data

Proses baca data merupakan kebalikan dari input data. Disini data dalam kartu didekrip kemudian ditampilkan dalam *textbox*.

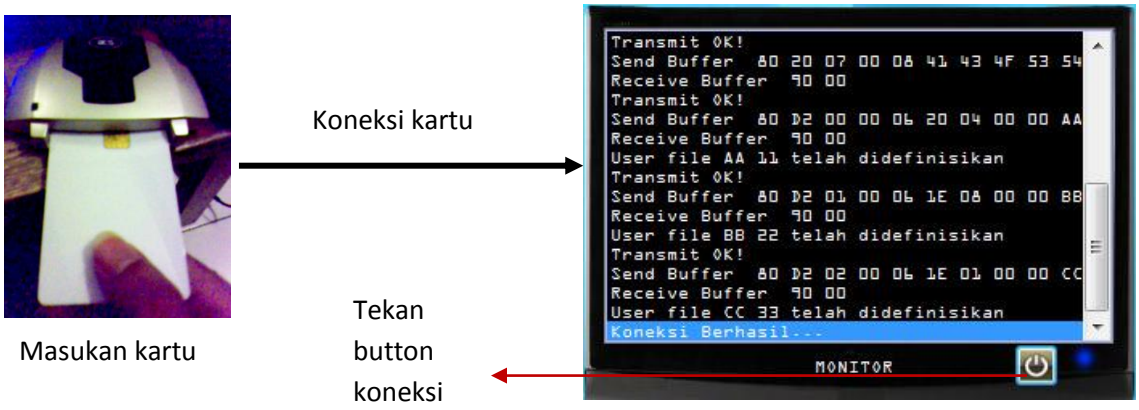
4.1.3.2.4 Securing Card

Antarmuka dan aksi untuk memberikan keamanan pada kartu dalam hal ini digambarkan sebuah kartu dari penginputan data secara sah diberikan keamanan untuk menghindari terjadinya penyalahgunaan kartu oleh pihak selain pengguna yang tidak memiliki hak terhadap kartu tersebut.

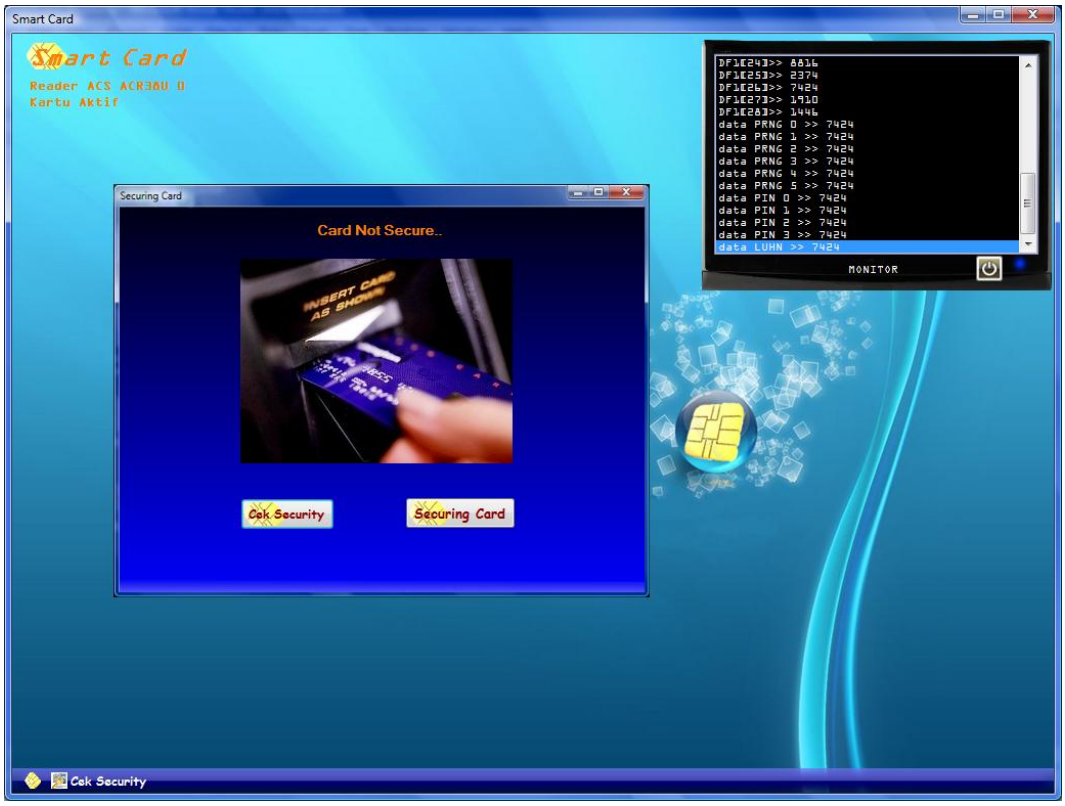
Langkah-langkah pada proses pemberian keamanan kartu adalah sebagai berikut :

1. Masukan kartu ke slot reader.
2. Lakukan koneksi ke ke kartu dengan menekan *button* koneksi, selanjutnya monitor akan menampilkan status koneksi.
3. Tekan *button* cek security, selanjutnya sistem akan menampilkan status keamanan kartu (*Card Secure/Card Not Secure..)*.
4. Tekan *button* securing card, selanjutnya aplikasi akan memberikan keamanan pada kartu dan memberikan PIN default pada kartu.

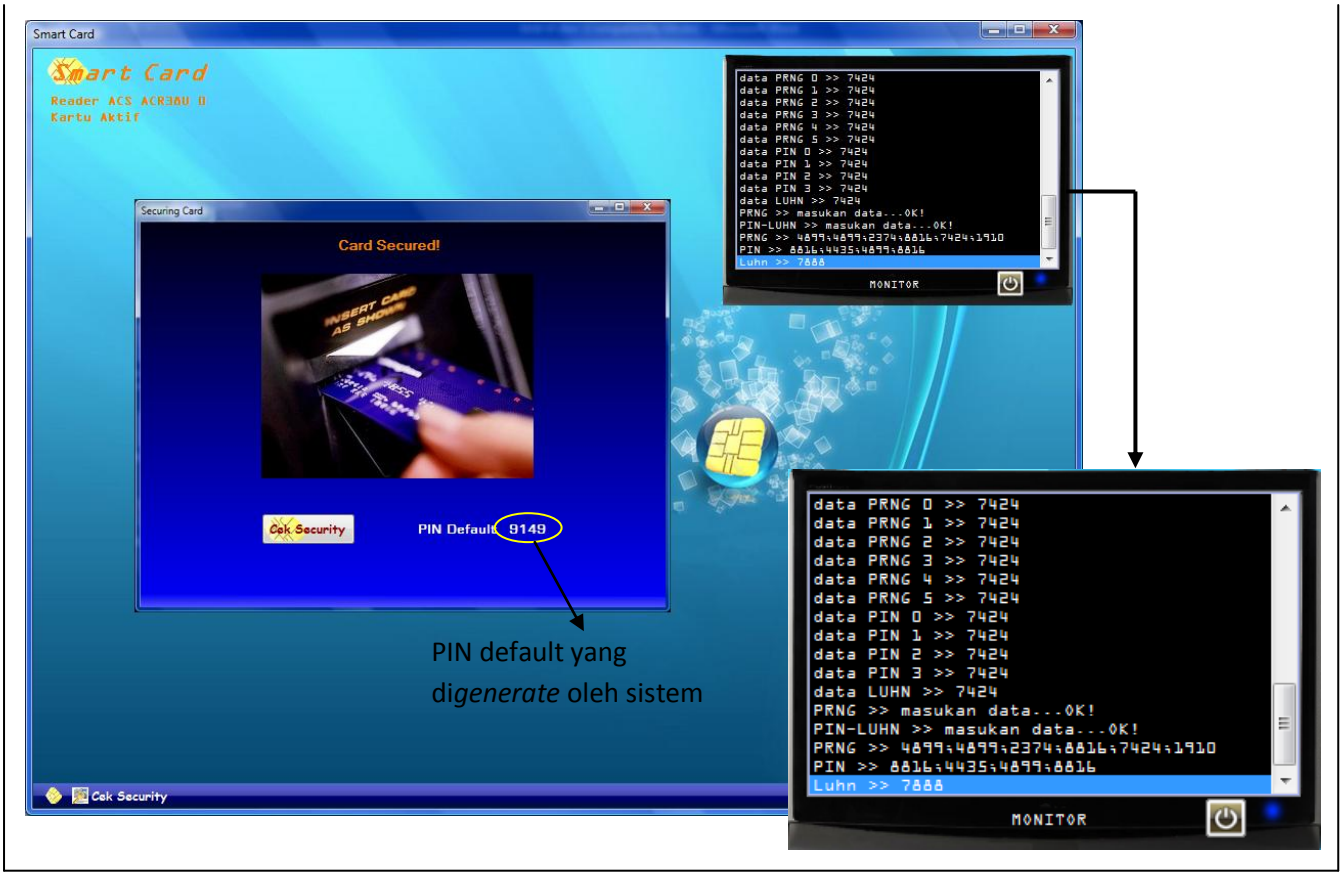
Agar lebih mudah dipahami, perhatikan gambar 4.6.



Selanjutnya tekan *button* Cek Security. Maka akan tampil status keamanan kartu. Serperti berikut.



Setelah terlihat status kartu "Card Not Secure..", kemudian tekan *button* "Securing Card", maka sistem akan menampilkan PIN yang digenerate oleh sistem dan menampilkan status "Card Secured!" seperti yang terlihat pada gambar berikut.



Gambar 4.6 Antarmuka dan aksi securing card

Aksi security diatas diasumsikan bahwa kartu telah terkoneksi ke *reader* dan data baru telah diinputkan ke kartu, kemudian admin tinggal melakukan securing terhadap kartu tersebut selanjutnya sistem akan langsung melakukan proses validasi luhn dan membuatkan (men-*generate*) nomor PIN.

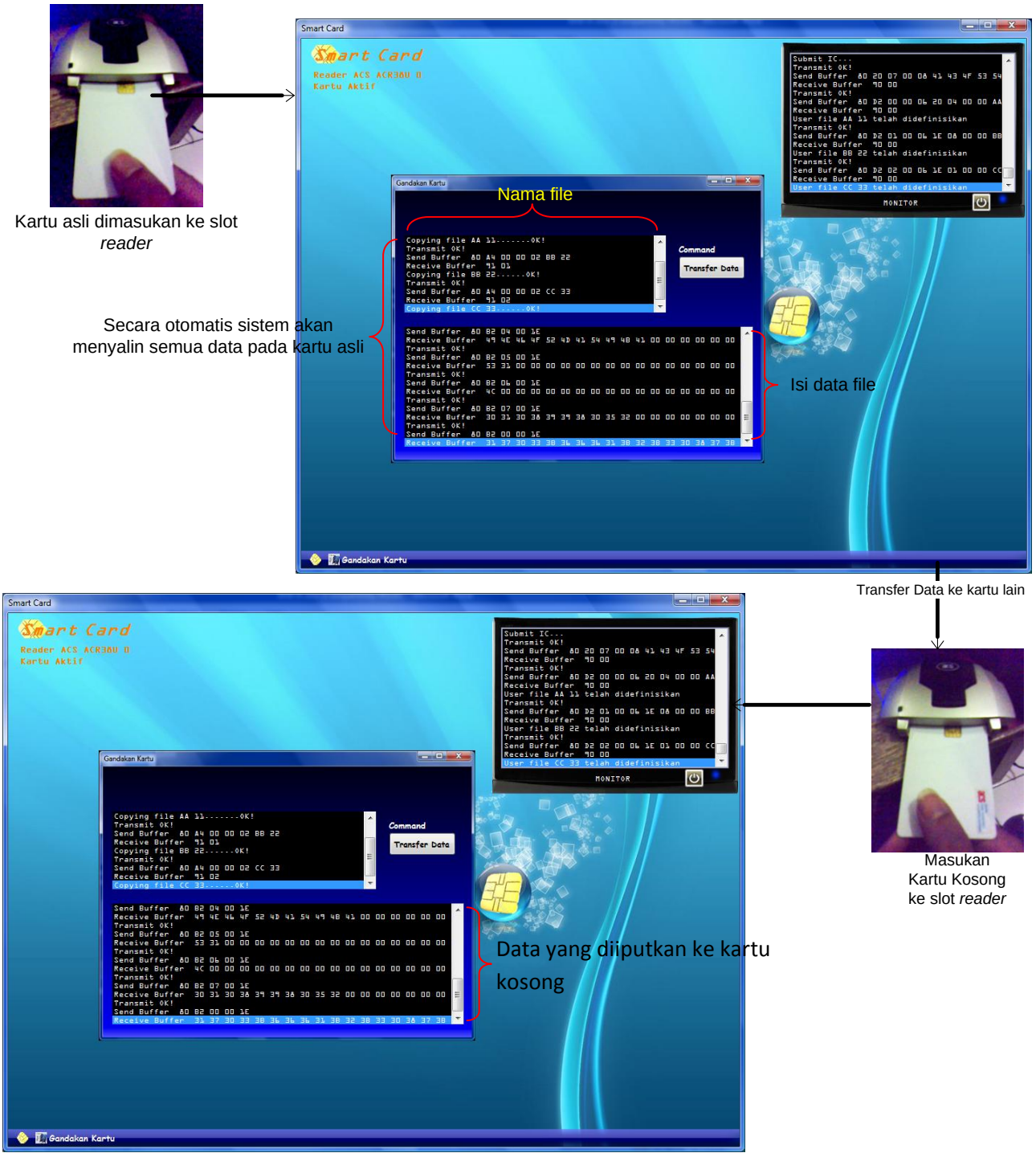
4.1.3.2.5 Gandakan Kartu

Antarmuka dan aksi gandakan kartu merupakan aksi nyata penggandaan data pada kartu. Dalam kasus ini *chipertext* tidak perlu didekripsi, semua data langsung disalin kemudian ditransfer ke kartu lainya. Aksi seperti ini merupakan contoh tipe serangan *chosen-chipertext attack.*

Langkah-langkah yang terjadi pada aksi gandakan kartu dapat dilihat sebagai berikut :

1. Masukan kartu asli yang akan digandakan ke slot *reader*, selanjutnya program secara otomatis akan melakukan penyalinan data pada kartu tersebut.
2. Masukan kartu kosong ke slot *reader* yang akan digunakan untuk menyimpan data hasil salin dari kartu asli.
3. Tekan *button* transfer data, maka data hasil salin akan dimasukan ke kartu kosong.

Lebih jelasnya, proses gandakan kartu dapat dilihat pada gambar 4.7.



Gambar 4.7 Antarmuka dan Aksi Gandakan Kartu

4.1.3.2.6 Cek Security Kartu

Antarmuka dan aksi Cek Security Kartu, merupakan proses pengecekan keabsahan sebuat kartu. Dalam kasus ini kartu asli dan kartu hasil

salinan kartu asli dicek. Hal ini juga dilakukan untuk menguji apakah algoritma validasi mampu membedakan kedua kartu tersebut.

Langkah-langkah proses cek security kartu dapat dilihat sebagai berikut :

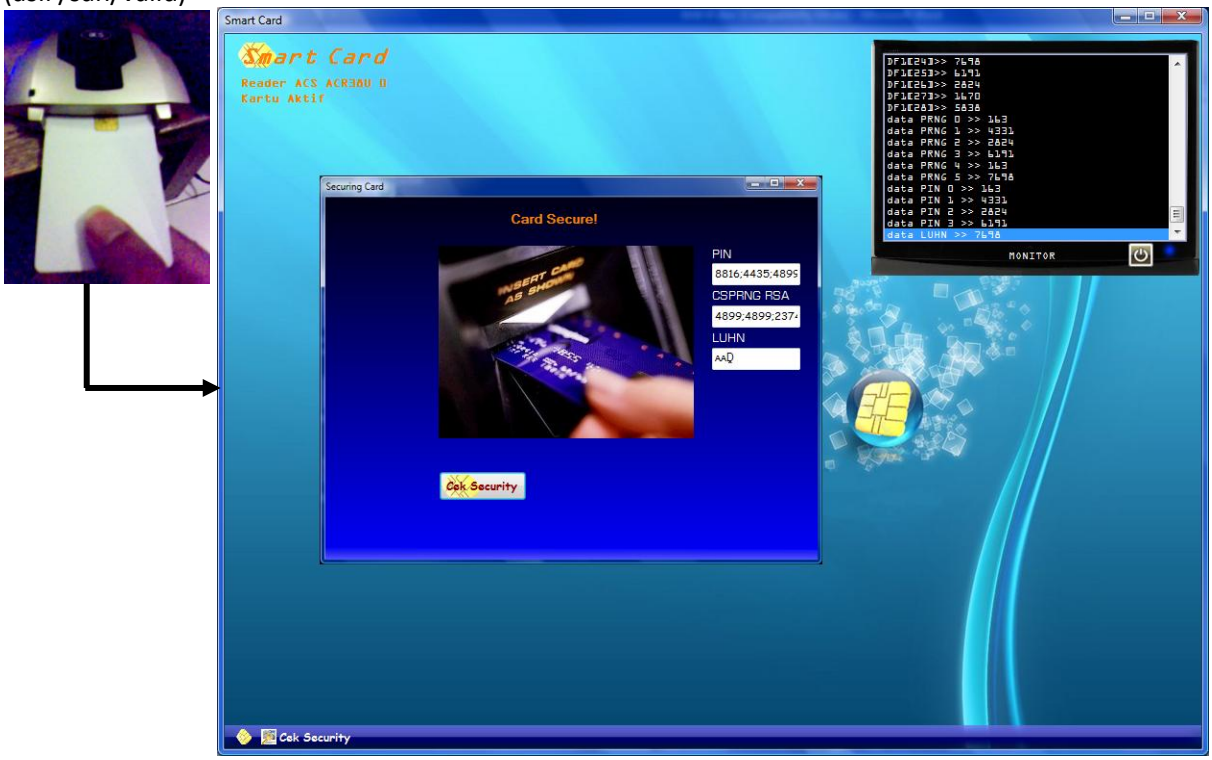
1. Masukan kartu ke slot *reader*.
2. Lakukan koneksi ke kartu dengan menekan *button* koneksi, selanjutnya monitor akan menampilkan status koneksi kartu.
3. Tekan *button* cek security, selanjutnya sistem akan menampilkan status security pada kartu.

    3.1 Jika kartu telah diberi security, maka akan tampil PIN, CSPRNG RSA dan LUHN yang telah dienkripsi.

    3.2 Jika kartu belum melalui tahap securing, maka akan tampil *button* Securing Card yang berfungsi untuk melakukan proses security pada kartu.

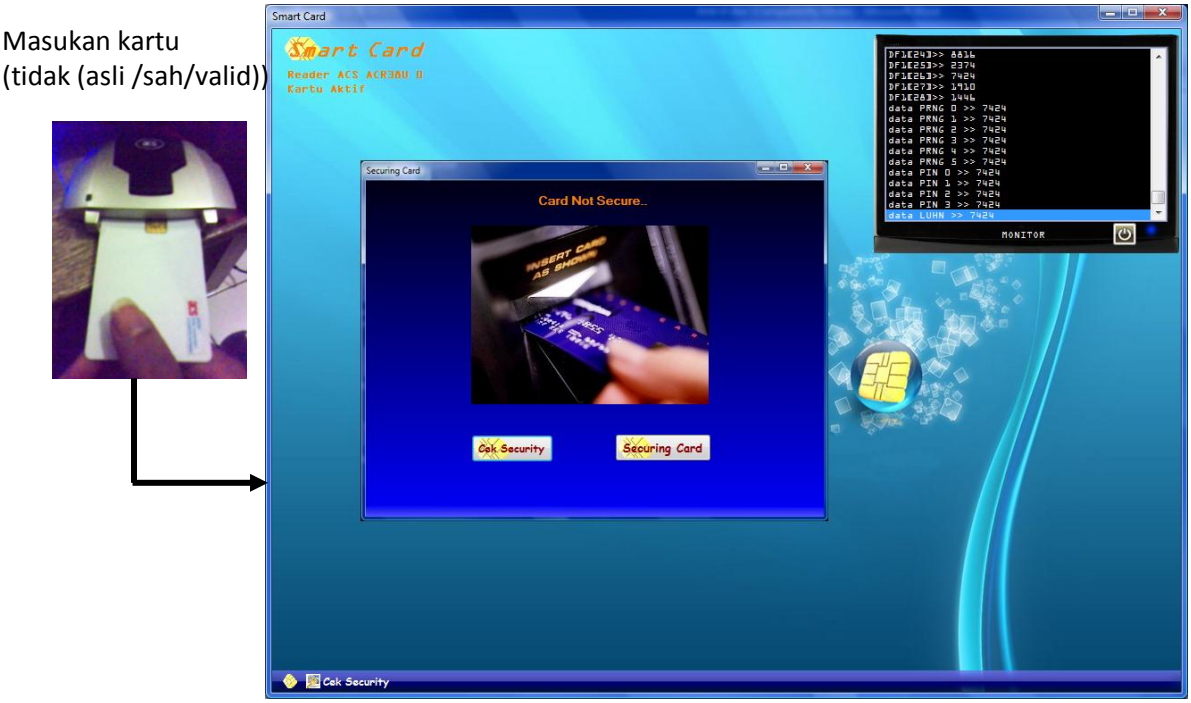
Lebih jelasnya perhatikan gambar 4.8 dan 4.9.

Masukan kartu
(asli /sah/valid)



Gambar 4.8 Antarmuka dan Aksi cek security kartu (Card Secure)

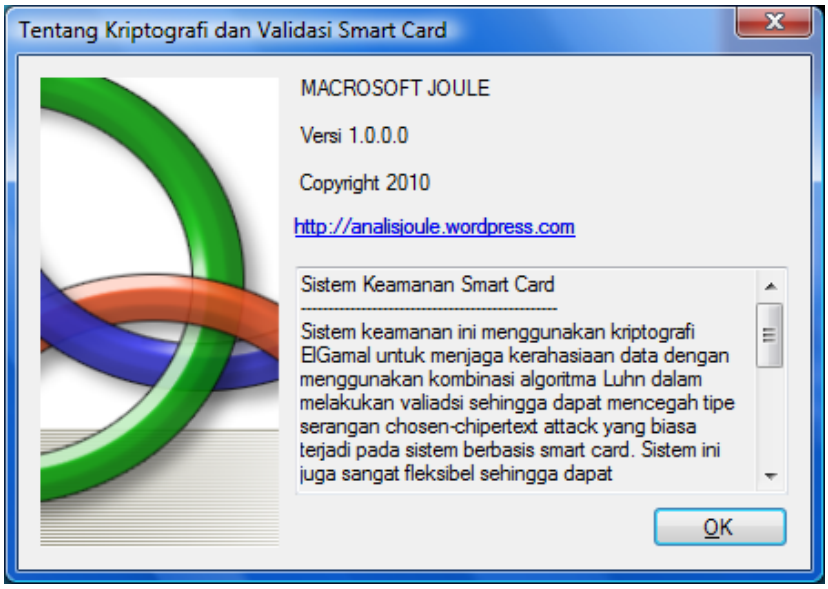
Masukan kartu
(tidak (asli /sah/valid))



Gambar 4.9 Tampilan cek security kartu (Card Not Secure)

Gambar diatas membuktikan bahwa sistem mampu membedakan kartu asli dan kartu hasil salinan kartu asli. Sehingga keamanan data masih terjamin.

4.1.3.2.7 Tentang



Gambar 4.10 Tampilan Form Tentang

4.2 Pengujian Sistem

Pengujian merupakan metode yang dilakukan untuk menjelaskan mengenai pengoprasian perangkat lunak yang terdiri dari metode pengujian, rencana pengujian dan pengujian alpha.

4.2.1 Metode Pengujian

Pengujian program ini menggunakan metode *Black-box*. Tujuan dari metode *Black Box* ini adalah untuk menemukan kesalahan fungsi pada program. Pengujian dengan metode *black box* dilakukan dengan cara mengimplenentasikan program yang kemudian diproses sesuai dengan tujuan

dibangunya program tersebut untuk melihat apakah menghasilkan keluaran yang diinginkan dan sesuai dengan fungsi dari program tersebut. Apabila dari masukan yang diberikan proses menghasilkan keluaran yang sesuai, maka program yang bersangkutan telah lulus uji, tetapi jika keluaran yang dihasilkan tidak sesuai dengan hasil analisis dan perancangan, maka masih terdapat kesalahan pada program yang dibangun.

4.2.2 Rencana Pengujian

Melakukan perencanaan dalam jenis-jenis proses yang akan diujikan dari implementasi sistem keamanan aplikasi ini, dan rencana hasil yang diinginkan dari proses pengujian nantinya.

Pengujian perangkat lunak sistem keamanan ini dilaksanakan dengan tujuan agar sistem yang dibangun memiliki kualitas yang lebih baik. Untuk rencana pengujian selengkapnya dapat dilihat pada tabel berikut :

Tabel 4.4 Rencana pengujian aplikasi sistem keamanan pada *smart card*

***Item* Uji**	**Butir Uji**	**Jenis Pengujian**
Autentikasi	Masukan kartu dan *password*	*Black Box*
Masukan Data Baru	*Input* data ke kartu	*Black Box*
	Baca data	*Black Box*
Cek Security Kartu	Cek security	*Black Box*
	Securing Card	*Black Box*
Gandakan Kartu	Mengambil salinan data dari kartu	*Black Box*
	Mentransfer data hasil salinan dari kartu lain	*Black Box*

4.2.3 Pengujian

Pengujian yang dilakukan menitikberatkan pada hasil *output* dari kendali *input*. Pada kendali *input,* pengujian dikatakan berhasil apabila *output* sesuai dengan kendali *input*.

Berdasarkan rencana pengujian aplikasi sistem keamanan ini telah disusun, maka dapat dilakukan pengujian sebagai berikut.

4.2.3.1 Pengujian Autentikasi

Hasil pengujian saat melakukan autentikasi menggunakan *smart card* antara lain sebagai berikut :

Tabel 4.5 Pengujian terhadap butir uji Autentikasi

Kasus dan Hasil Uji [Menggunakan Data Normal]			
Butir uji	**Hasil yang diharapkan**	**Hasil pengamatan**	**Kesimpulan**
Masukan kartu	Sistem mampu mendetekasi keberadaan kartu dan menampilkan *textbox password, button* login dan keluar	*textbox password, button* login dan *button* keluar muncul setelah sistem mendeteksi keberadaan kartu pada *reader*	[√] sukses [ ] gagal
Masukan password (kartu salah password salah)	Menampilkan pesan gagal	Sistem menampilkan pesan gagal	[√] sukses [ ] gagal
Masukan password (kartu benar password salah)	Menampilkan pesan gagal	Sistem menampilkan pesan gagal	[√] sukses [ ] gagal
Masukan	Menampilkan	Sistem	[√] sukses

password (kartu salah password benar)	pesan gagal	menampilkan pesan gagal	[ ] gagal
Masukan password (kartu benar password benar)	Login ke menu utama	Login ke menu utama	[√] sukses [ ] gagal

4.2.3.2 Pengujian Masukan Data Baru

Hasil pengujian saat menginputkan dan membaca data *smart card* antara lain sebagai berikut :

Tabel 4.6 Pengujian terhadap butir uji Masukan Data Baru

Kasus dan Hasil Uji [Menggunakan Data Normal]			
Butir uji	**Hasil yang diharapkan**	**Hasil pengamatan**	**Kesimpulan**
Input data ke kartu	- Enkripsi NIM sebelum menginputkan data ke kartu - Menampilkan data hasil enkripsi	- Data berhasil diinputkan ke kartu dan menampilkan data hasil enkripsi	[√] sukses [ ] gagal
Baca data	- Melakukan dekripsi terhadap *chipertext* NIM dan tampilkan pada *textbox*	- Data NIM berhasil didekrip dan ditampilkan pada *textbox*	[√] sukses [ ] gagal

4.2.3.3 Pengujian Cek Security Kartu

Hasil pengujian saat pengecekan keamanan kartu dan melakukan security terhadap kartu tersebut antara lain sebagai berikut :

Tabel 4.7 Pengujian terhadap butir uji Cek Security

Kasus dan Hasil Uji [Menggunakan Data Normal]			
Butir uji	**Hasil yang diharapkan**	**Hasil pengamatan**	**Kesimpulan**
Cek Security (menggunakan kartu tidak valid)	Sistem dapat menampilkan status “Card Not Secure” dan *button* “Securing Card”	- Sistem menampilkan status “Card Not Secure” dan *button* “Securing Card”	[√] sukses [ ] gagal
Cek Security (menggunakan kartu valid)	Sistem dapat menampilkan status “Card Secure!” dan hasil enkrip NIM, CSPRNG RSA, LUHN	- Sistem menampilkan status “Card Secure!” dan hasil enkrip NIM, CSPRNG RSA, LUHN	[√] sukses [ ] gagal
Securing Card	Sistem melakukan securing pada kartu serta menampilkan status “Card Secured!” dan menampilkan PIN	- Kartu dapat diberikan security serta menampilkan status “Card Secured!” dan menampilkan PIN	[√] sukses [ ] gagal

4.2.3.4 Pengujian Gandakan Kartu

Hasil pengujian saat melakukan penggandaan kartu, dimana kartu yang telah memiliki data disalin ke kartu kosong antara lain sebagai berikut :

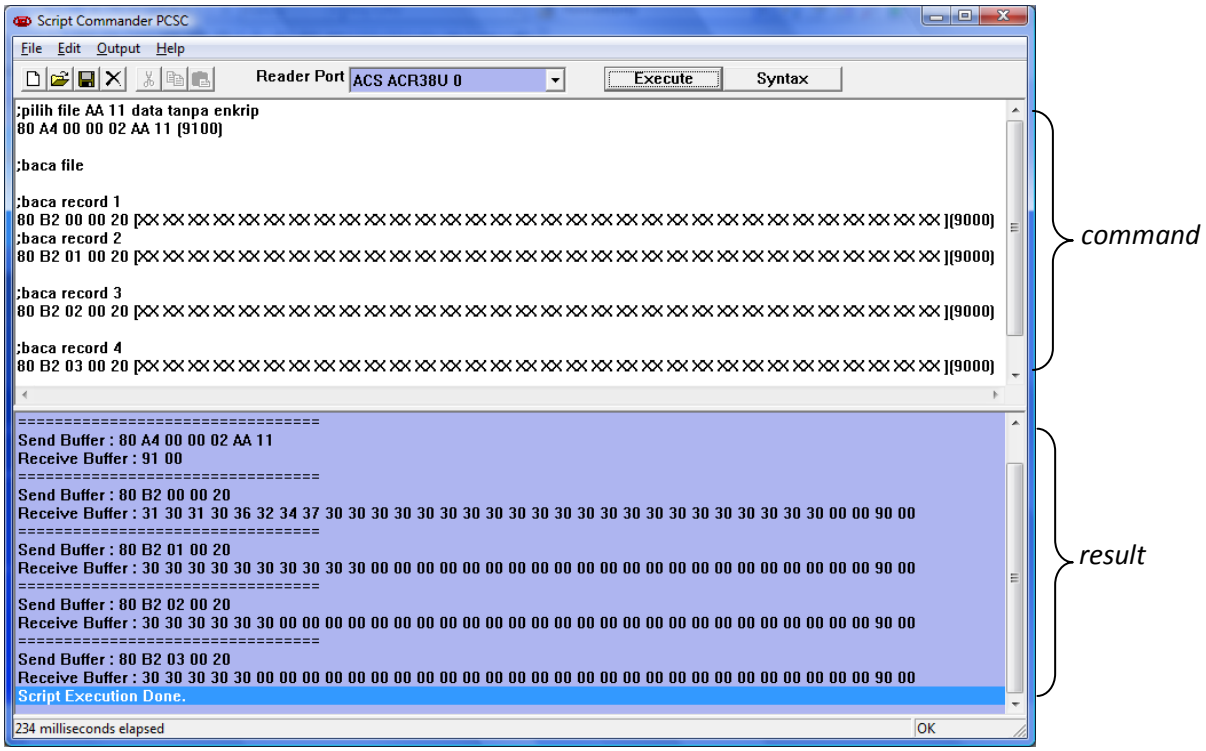
Tabel 4.8 Pengujian terhadap butir uji Gandakan Kartu

Kasus dan Hasil Uji [Menggunakan Data Normal]			
Butir uji	**Hasil yang diharapkan**	**Hasil pengamatan**	**Kesimpulan**
Mengambil salinan data dari kartu	Menyalin data dari kartu	Melakukan penyalinan data kartu	[√] sukses [ ] gagal
Mentransfer data hasil salinan dari kartu lain	Mentransfer data hasil salinan dari sebuah kartu ke kartu lain/kosong	Data hasil salinan berhasil ditransfer ke kartu lain/kosong	[√] sukses [ ] gagal

4.2.4 Pengujian Menggunakan *Software Scripting Tool*

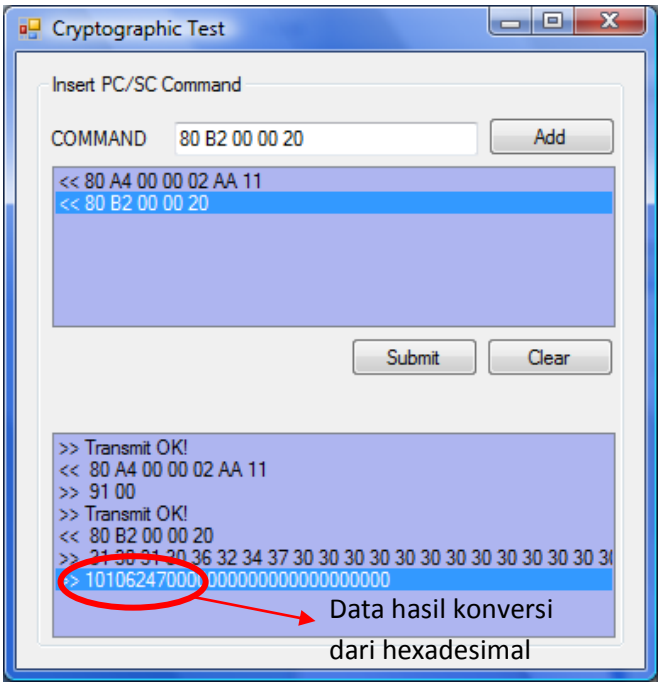
Dalam hal ini pengujian dilakukan menggunakan software *scripting tool*. Software ini merupakan *script commander* manual untuk melihat data file dalam ruang-ruang record pada *smart card*. Dengan menggunakan software ini data akan terlihat dalam bentuk asli(hexadesimal) sehingga cocok untuk menguji data hasil enkripsi.

Pada kasus ini akan dilakukan pengujian data hasil enkripsi terhadap NIM yang telah di *input* pada kartu. Sebagai pembanding, disini disediakan kartu lain dengan menggunakan NIM yang sama namun tidak melalui tahap enkripsi. Perhatikan gambar berikut :



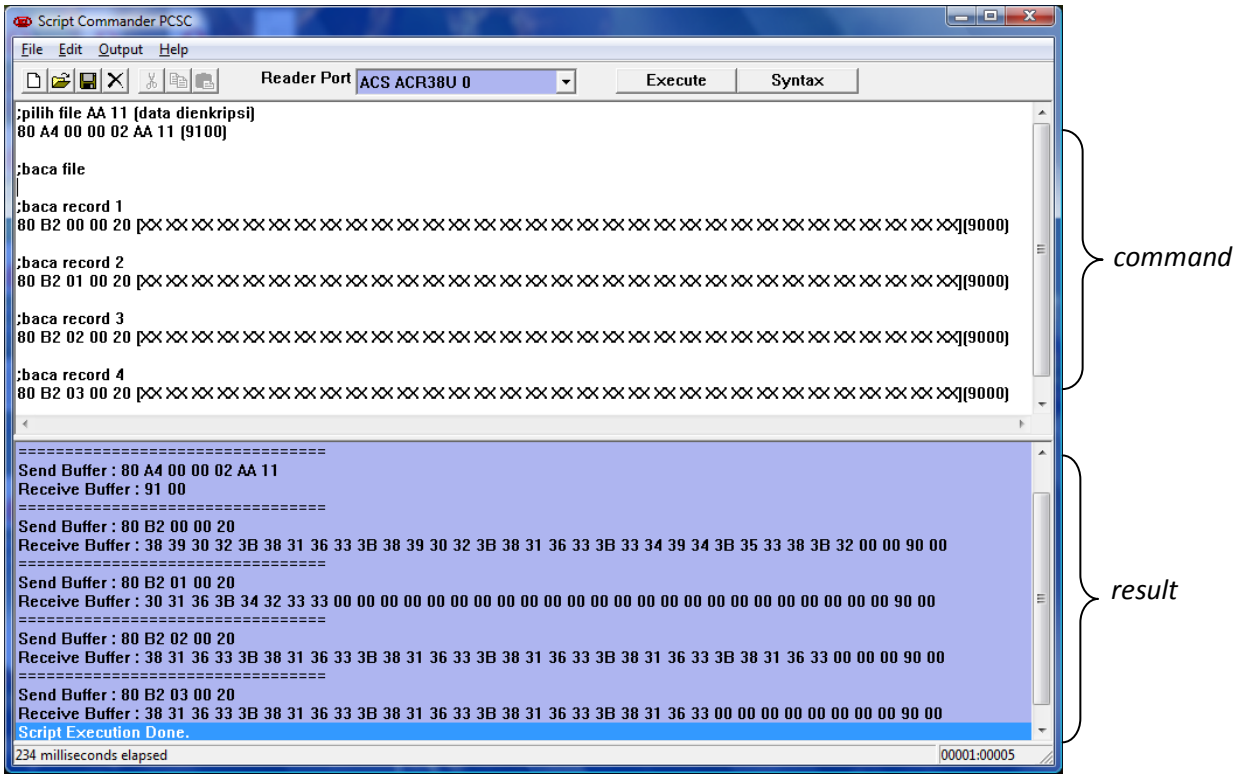
Gambar 4.11 Pengujian data NIM (tanpa enkripsi)

Pada gambar 4.10 dalam *field* "*command"* terdapat perintah pilih file dan baca *record* record. *Command* pilih file dilakukan untuk memilih sebuah file yang terdiri dari beberapa data didalamnya. Kemudian *command* baca record dilakukan untuk membaca record yang terdapat pada file tersebut. Selanjutnya pada *field "result"* (baris 6 : *Receive Buffer*) terlihat bahwa pada kasus ini *smart card* menyimpan data NIM (dalam bentuk hexadesimal) tanpa dienkripsi sebagai berikut :



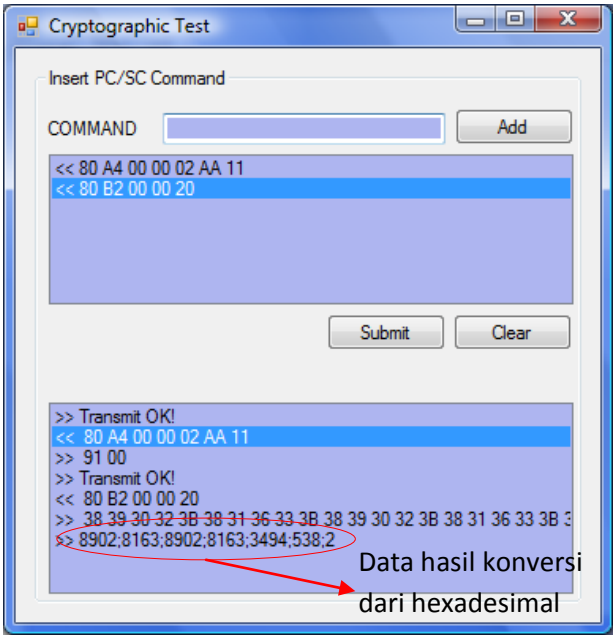
Gambar 4.12 Data hasil konversi(tanpa enkripsi)

Selanjutnya dengan menggunakan kartu yang berbeda namun dengan data yang sama (NIM : 10106247) hasil enkripsi, maka akan tampil seperti berikut :



Gambar 4.13 Pengujian data NIM (hasil enkripsi)

Pada gambar 4.12 dalam *field "result"* (baris 6 : *Receive Buffer*) terlihat bahwa pada kasus ini *smart card* menyimpan data hasil enkripsi NIM (dalam bentuk hexadesimal).



Gambar 4.14 Data hasil konversi (hasil enkripsi)

Tabel 4.9 Pengujian menggunakan *Scripting Tool*

Kasus dan Hasil Uji [Menggunakan Data Normal]			
Butir uji	**Hasil yang diharapkan**	**Hasil pengamatan**	**Kesimpulan**
Data NIM hasil enkrip	Menampilkan data enkripsi dan panjang data (hexadesimal) > 8	Tampil data hasil enkripsi dengan panjang data hexadesimal > 8	[√] sukses [ ] gagal
Data NIM tanpa enkrip	Menampilkan data NIM	Tampil data NIM	[√] sukses [ ] gagal

4.2.5 Kesimpulan Hasil Pengujian

Berdasarkan hasil pengujian dengan kasus uji *sample* diatas dapat ditarik kesimpulan bahwa perangkat lunak bebas dari kesalahan sintaks dan secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan. Beberapa hal lain dari hasil pengujian yang juga menjadi sorotan diantaranya antara lain :

1. Hasil enkripsi dapat secara kuat mempertahankan kerahasiaan data dan sukses ketika diimplementasikan pada *smart card* (termasuk dekrip data).
2. Pengecekan security pada *smart card* berjalan sesuai dengan hasil analisis. Sehingga sistem mampu mengetahui perbedaan antar kartu.
3. Sistem mampu memerikan security pada *smart card* sehingga kartu yang bersangkutan (secara teori dan hasil implementasi) aman dari aksi penggandaan kartu.