BAB III

ANALISIS DAN PERANCANGAN

3.1 Analisis sistem

Sistem yang akan dibangun dirancang sebagai sistem keamanan pada *smart card*, karenanya sistem ini diharapkan dapat memenuhi kebutuhan antara lain :

1. Sistem mampu melakukan autentikasi menggunakan *smart card* untuk membatasi penggunaan dari pihak yang tidak berkepentingan.
2. Mampu melakukan pengisian data baru pada *smart card* dengan data yang telah dienkripsi terlebih dahulu.
3. Mampu membaca (melakukan dekripsi) data pada *smart card.*
4. Menggandakan data pada *smart card*.
5. Dapat melakukan cek keamanan sehingga mengetahui keabsahan sebuah *smart card* (apakah asli atau palsu).
6. Memberikan keamanan terhadap *smart card*.

3.1.1 Analisis kebutuhan non-fungsional

Analisis kebutuhan dilakukan untuk mengetahui spesifikasi kebutuhan untuk sistem. Spesifikasi kebutuhan terdiri atas analisis perangkat keras , dan analisis perangkat lunak .

3.1.1.1 Analisis kebutuhan perangkat keras

Perangkat keras yang digunakan dalam mengimplementasikan sistem keamanan pada *smart card* yang akan dibangun menggunakan spesifikasi sebagai berikut :

Tabel 3.1 Spesifikasi Perangkat Keras

Nama perangkat	Spesifikasi
Komputer	Toshiba Satellite® L305-S5946
Processor	Intel® Core 2™ Duo Processor T6400
Memory	DDR2 3 GB
Hardisk	250 GB
Monitor	15.4” TruBrite® Display 1280x800
Smart card reader	ACR38 reader/writer microprocessor dengan protocol T=0/T=1 support ISO-7816 Class A
Smart card	ACOS2,ACOS3 (memiliki sistem operasi tersendiri) dan SLE4428 (tipe memory)

3.1.1.2 Analisis kebutuhan perangkat lunak

Pada analisis kebutuhan perangkat lunak ini menjelaskan tentang perangkat lunak yang digunakan dalam membuat aplikasi yang penulis bangun adalah sebagai berikut :

Tabel 3.2 Spesifikasi Perangkat Lunak

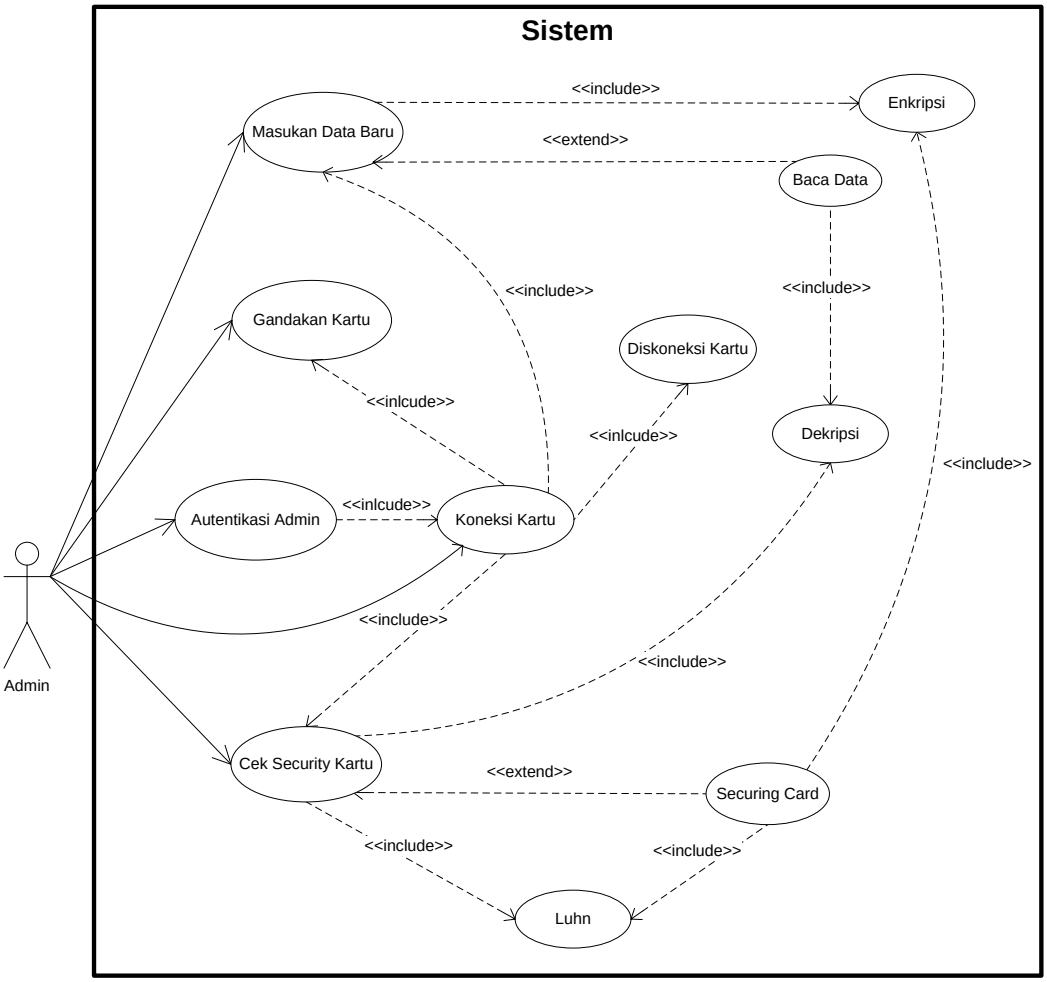
Nama perangkat lunak	Spesifikasi
Sistem operasi	Windows Vista® Home Premium SP1
Compiler	Visual Studio 2008
Tool pendukung	.NET FrameWork 3.5, Card Tool, PCSC learning tool, Scripting Tool, Quick View, Smart card reader monitor

3.1.2 Analisis kebutuhan fungsional

Analisis kebutuhan fungsional merupakan tindak lanjut dari analisis sistem. Tujuanya adalah untuk memberikan gambaran mengenai sistem yang akan dibuat.

3.1.2.1 *Use Case Diagram*

Use Case adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use Case* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem itu digunakan.



Gambar 3.1 ***Use Case Diagram***

3.1.2.1.1 Autentikasi Admin

Skenario :

Identifikasi	
Nama	Autentikasi Admin
Tujuan	Memastikan hak akses aplikasi
Deskripsi	Sistem menerima identifikasi data menggunakan *smart card*. Disini admin tidak dapat menginputkan *password* sebelum *reader* mendeteksi keberadaan kartu.

Aktor		Admin	
Skenario Utama			
Kondisi awal		Aplikasi baru dijalankan	
No	**Aksi Aktor**	**No**	**Reaksi Sistem**
1	Masukan *smart card*	2	Menampilkan *textfield password* kemudian lakukan koneksi ke *smart card* lalu ambil ATR.
2	Masukan *password*	4	Mengecek kebenaran *password*
Skenario Alternatif – Autentikasi Gagal			
No	**Reaksi Sistem**	**No**	**Aksi Aktor**
1	Menampilkan pesan gagal	2	Ulangi autentikasi, masukan *smart card*
Kondisi akhir		Hak akses admin sudah dipastikan	

3.1.2.1.2 Koneksi Kartu

Skenario :

Identifikasi	
Nama	Koneksi Kartu
Tujuan	Melakukan koneksi dari *reader* ke *smart card*
Deskripsi	Koneksi ke kartu dilakukan untuk membangun sebuah komunikasi antara aplikasi dengan kartu agar dapat melakukan operasi-operasi baca, tulis dan transmit APDU

Aktor		Admin	
Skenario Utama			
Kondisi awal		Admin sudah terautentikasi	
No	**Aksi Aktor**	**No**	**Reaksi Sistem**
1	Masukan *smart card*	2	*Reader* mendeteksi adanya kartu
3	Klik button koneksi	4	Membangun koneksi dengan *smart card*
Skenario Alternatif – Koneksi kartu Gagal			
No	**Reaksi Sistem**	**No**	**Aksi Aktor**
1	Menampilkan pesan gagal	2	Periksa *smart card* atau ulangi koneksi.
Kondisi akhir	Sistem dapat berinteraksi dengan *smart card* dan file-file dalam *smart card* tersebut telah terdefinisi.		

3.1.2.1.3 Diskoneksi Kartu

Skenario :

Identifikasi	
Nama	Diskoneksi Kartu
Tujuan	Melakukan pemutusan dari *reader* ke *smart card*
Deskripsi	Pemutusana koneksi dilakukan untuk memutuskan komunikasi antara *reader* dengan *smart card* sehingga sistem sudah tidak lagi dapat berinteraksi atau melakukan operasi-operasi baca, tulis dan sebagainya.

Aktor	Admin		
Skenario Utama			
Kondisi awal	Admin sudah terautentikasi		
No	**Aksi Aktor**	**No**	**Reaksi Sistem**
1	Melakukan pemutusan koneksi	2	Putuskan koneksi dengan kartu
Skenario Alternatif – Diskoneksi kartu Gagal			
No	**Reaksi Sistem**	**No**	**Aksi Aktor**
1	Menampilkan pesan gagal memutuskan koneksi	2	Periksa *smart card* kemungkinan sudah terjadi pemutusan koneksi.
Kondisi akhir	Sistem sudah tidak dapat berinteraksi dengan *smart card*		

3.1.2.1.4 Masukan Data Baru

Skenario :

Identifikasi	
Nama	Masukan Data Baru
Tujuan	Membuat data baru yang akan digunakan pada *smart card*
Deskripsi	Admin membuat data baru kemudian menginputkanya ke kartu, sebelum input data, sistem akan melakukan enkripsi terhadap NIM
Aktor	Admin
Skenario Utama	
Kondisi awal	Admin telah terautentikasi

No	Aksi Aktor	No	Reaksi Sistem
1	Input data baru	2	Merespon penyimpanan data dan mengembalikan pesan (sukses/gagal)
Skenario Alternatif – Masukan data baru gagal			
No	**Reaksi Sistem**	**No**	**Aksi Aktor**
1	Menampilkan pesan gagal	2	Ulangi input data atau cek koneksi kartu atau cek perangkat keras (*reader* atau komponen yang berkaitan).
Kondisi akhir	Tampil status input data baru		

3.1.2.1.5 Baca Data

Skenario :

Identifikasi	
Nama	Baca Data
Tujuan	Membaca data dalam *smart card*
Deskripsi	Admin membaca data yang telah diinputkan ke kartu dengan tujuan memastikan bahwa data telah tersimpan
Aktor	Admin
Skenario Utama	
Kondisi awal	Admin telah terautentikasi

No	Aksi Aktor	No	Reaksi Sistem
1	Klik button baca	2	Baca record file dalam *smart card*, lakukan dekrip pada data yang terenkripsi kemudian

			tampilkan data tersebut pada form.
Skenario Alternatif – Baca Data gagal			
No	**Reaksi Sistem**	**No**	**Aksi Aktor**
1	Menampilkan pesan gagal	2	Ulangi koneksi kartu atau cek kartu.
Kondisi akhir	Tampilkan data pada form		

3.1.2.1.6 Gandakan Kartu

Skenario :

Identifikasi			
Nama	Gandakan Kartu		
Tujuan	Mengganakan inforamsi pada kartu		
Deskripsi	*chosen-chipertext attack*, chiperteks pada kartu digandakan ke kartu lainya. Disini data dari kartu yang pertama masuk akan disalin sepenuhnya lalu dimasukan ke kartu lainya.		
Aktor	Admin		
Skenario Utama			
Kondisi awal	Admin telah terautentikasi		
No	**Aksi Aktor**	**No**	**Reaksi Sistem**
1	Masukan kartu yang akan digandakan	2	Menyalin informasi pada kartu
3	Masukan kartu kosong	4	*Reader* mendeteksi keberadaan kartu
5	Klik button transfer	6	Isi kartu kosong dengan inforamasi

			dari kartu sebelumnya
Skenario Alternatif – Gandakan Katu gagal			
No	**Reaksi Sistem**	**No**	**Aksi Aktor**
1	Menampilkan pesan *Error!*,(Kartu tidak bisa dibaca/tidak valid)	2	Periksa kartu, *reader* atau koneksi ulang.
Kondisi akhir	Data pada kartu asli digandakan ke kartu lain		

3.1.2.1.7 Cek Security Kartu

Skenario :

Identifikasi			
Nama	Cek Security Kartu		
Tujuan	Melakukan pengecekan keamanan pada kartu		
Deskripsi	Sistem menguji keamanan kartu sehingga dapat mengetahui bahwa apakah kartu aman atau tidak. Cek security juga dilakukan untuk menguji ketika data kartu digandakan.		
Aktor	Admin		
Skenario Utama			
Kondisi awal	Admin telah terautentikasi		
No	**Aksi Aktor**	**No**	**Reaksi Sistem**
1	Klik button cek security kartu	2	Ambil cipherteks NIM, CSPRNG RSA dan digit Luhn lakukan dekripsi, ambil MCUID *smart card,* , lalu cek menggunakan Luhn.

Skenario Alternatif – Cek Security Kartu gagal			
No	**Reaksi Sistem**	**No**	**Aksi Aktor**
1	Menampilkan pesan gagal, kartu tidak terbaca..	2	Periksa kartu atau cek koneksi kartu.
Kondisi akhir	Informasi security kartu ditampilkan		

3.1.2.1.8 Securing Card

Skenario :

Identifikasi			
Nama	Securing Card		
Tujuan	Memberikan keamanan terhadap kartu		
Deskripsi	memberikan keamanan dilakukan agar data dalam kartu meskipun berhasil disalin ke kartu, tetap tidak dapat digunakan.		
Aktor	Admin		
Skenario Utama			
Kondisi awal	Admin telah terautentikasi		
No	**Aksi Aktor**	**No**	**Reaksi Sistem**
1	Klik button Securing Card	2	Ambil cipherteks NIM, ambil MCUID *smart card*, buat CSPRNG RSA kemudian buat digit luhn. enkrip CSPRNG RSA dan digit luhn tersebut lalu simpan dalam record file.
Skenario Alternatif – Securing Card gagal			

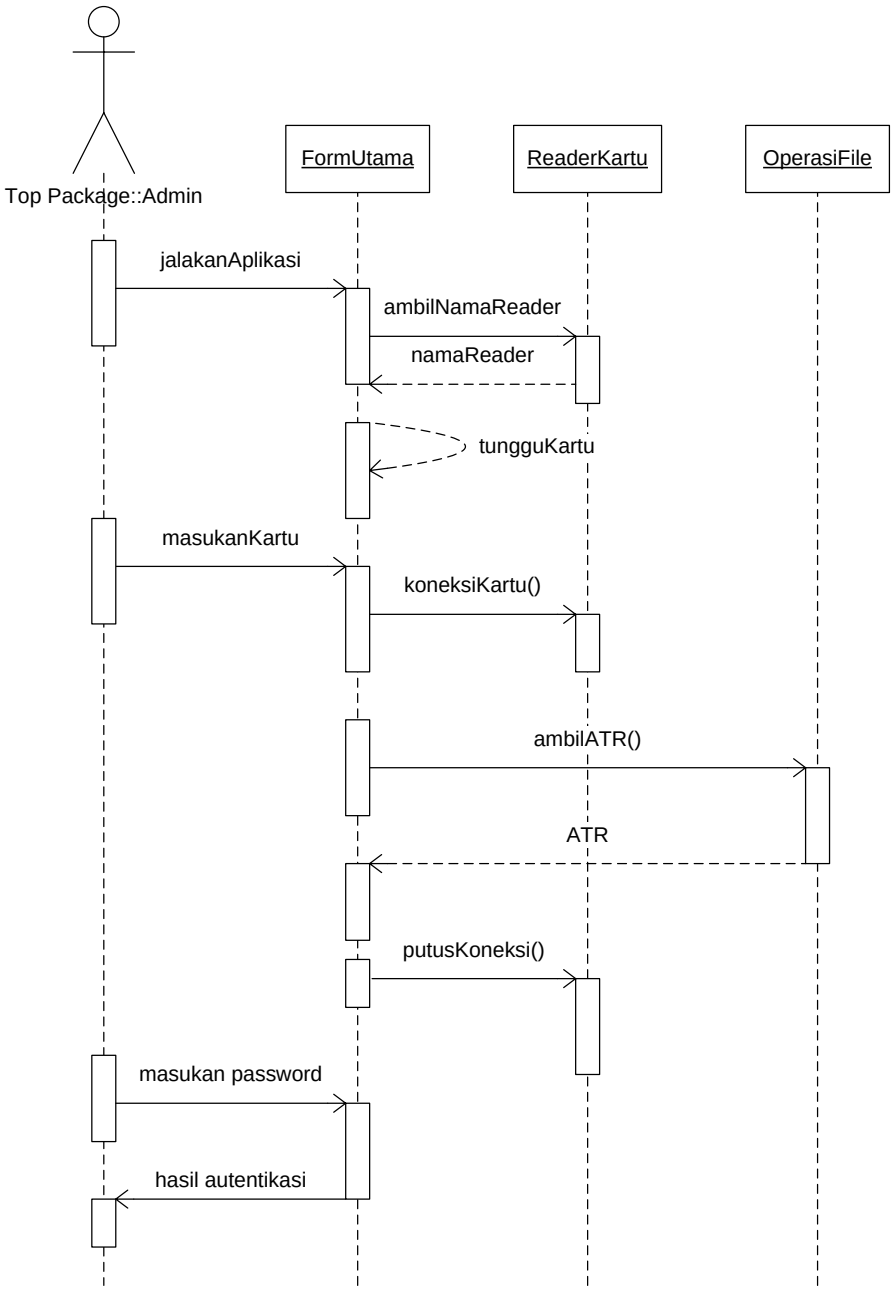
No	Reaksi Sistem	No	Aksi Aktor
1	Manampilkan pesan kegagalan	2	Periksa kartu atau cek koneksi kartu kemudian ulangi securing card.
Kondisi akhir	Informasi kartu yang telah diberi security ditampilkan		

3.1.2.2 *Sequence Diagram*

Sequence diagram digunakan untuk menggambarkan interaksi dan prilaku antara sejumlah *object* dalam rangkaian waktu. Secara khusus, menjabarkan *behavior* skenario tunggal. Diagram tersebut menunjukkan sejumlah objek dan pesan yang melewati objek-objek di dalam *Use Case*.

3.1.2.2.1 Autentikasi admin

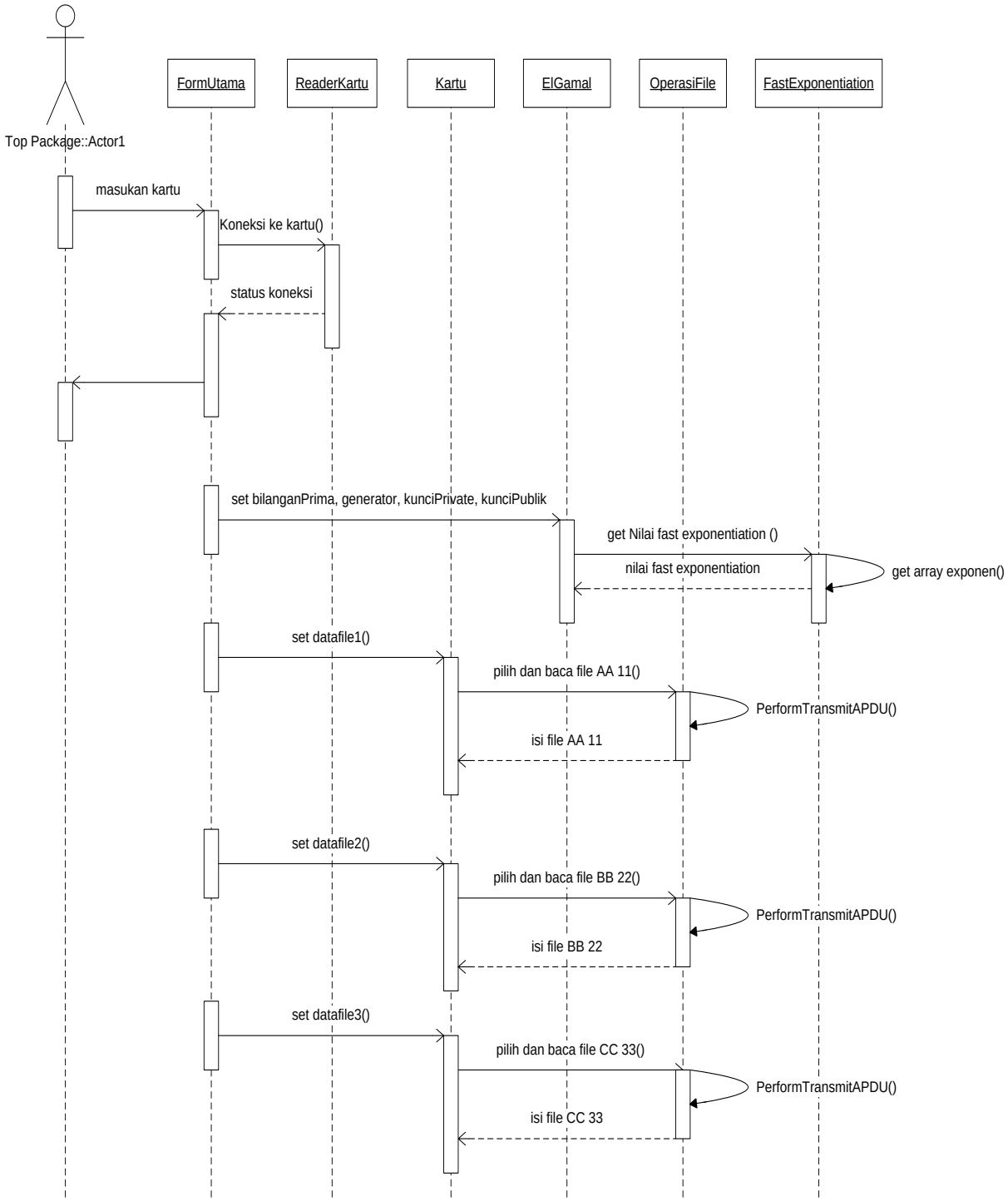
Urutan proses autentikasi admin ketika aplikasi pertama dijalankan. Dalam proses ini juga akan digambarkan bahwa proses autentikasi menggunakan *smart card*.



Gambar 3.2 *Sequence diagram* Autentikasi Admin

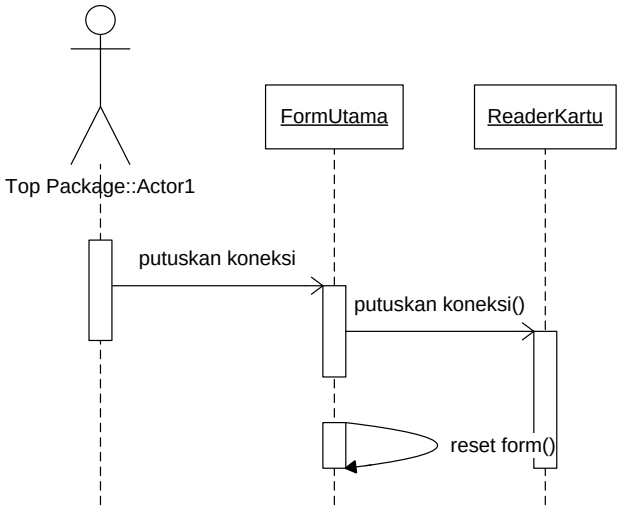
3.1.2.2.2 Koneksi Kartu

Koneksi kartu dilakukan untuk dapat melakukan operasi-operasi yang berhubungan dengan *smart card.*



Gambar 3.3 *Diagram sequence* Koneksi Kartu

3.1.2.2.3 Diskoneksi kartu

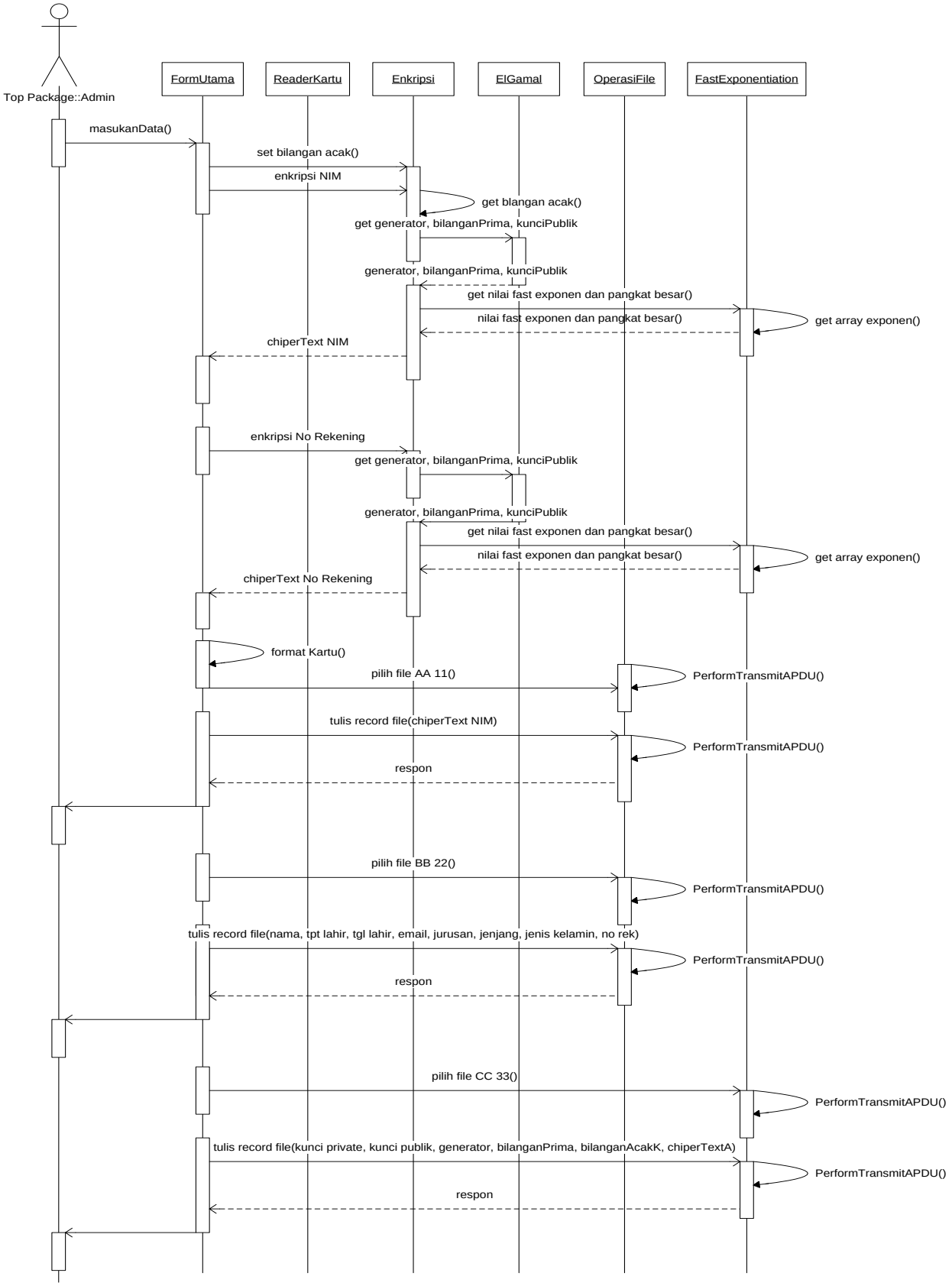


Gambar 3.4 *Sequence Diagram* Diskoneksi kartu

3.1.2.2.4 Masukan Data Baru

Admin memasukan *smart card* yang akan diisikan data kedalamnya, kemudian sistem akan melakukan format dan koneksi pada kartu. Format disini merupakan proses pendefinisian file pada sistem operasi *smart card* yang akan digunakan dalam proses pembuatan data baru. Setelah respon dari koneksi ternyata OK, maka admin dapat memasukan data baru pada kartu.

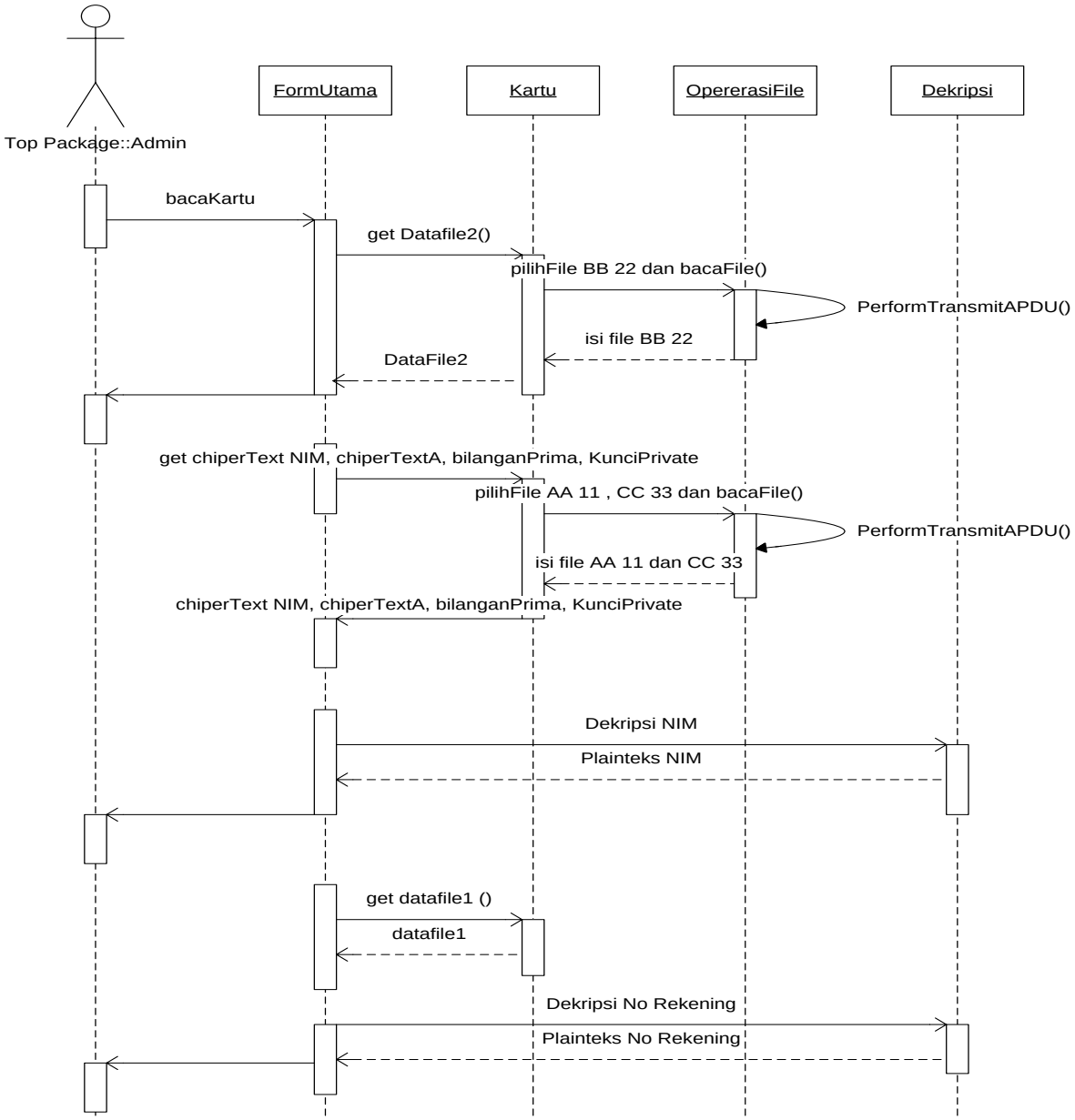
Sebelum input data pada *smart card*, terlebih dahulu sistem melakukan enkripsi terhadap data. Berikut urutan proses input data.



Gambar 3.5 *Sequence diagram* Masukan Data Baru

3.1.2.2.5 Baca Data

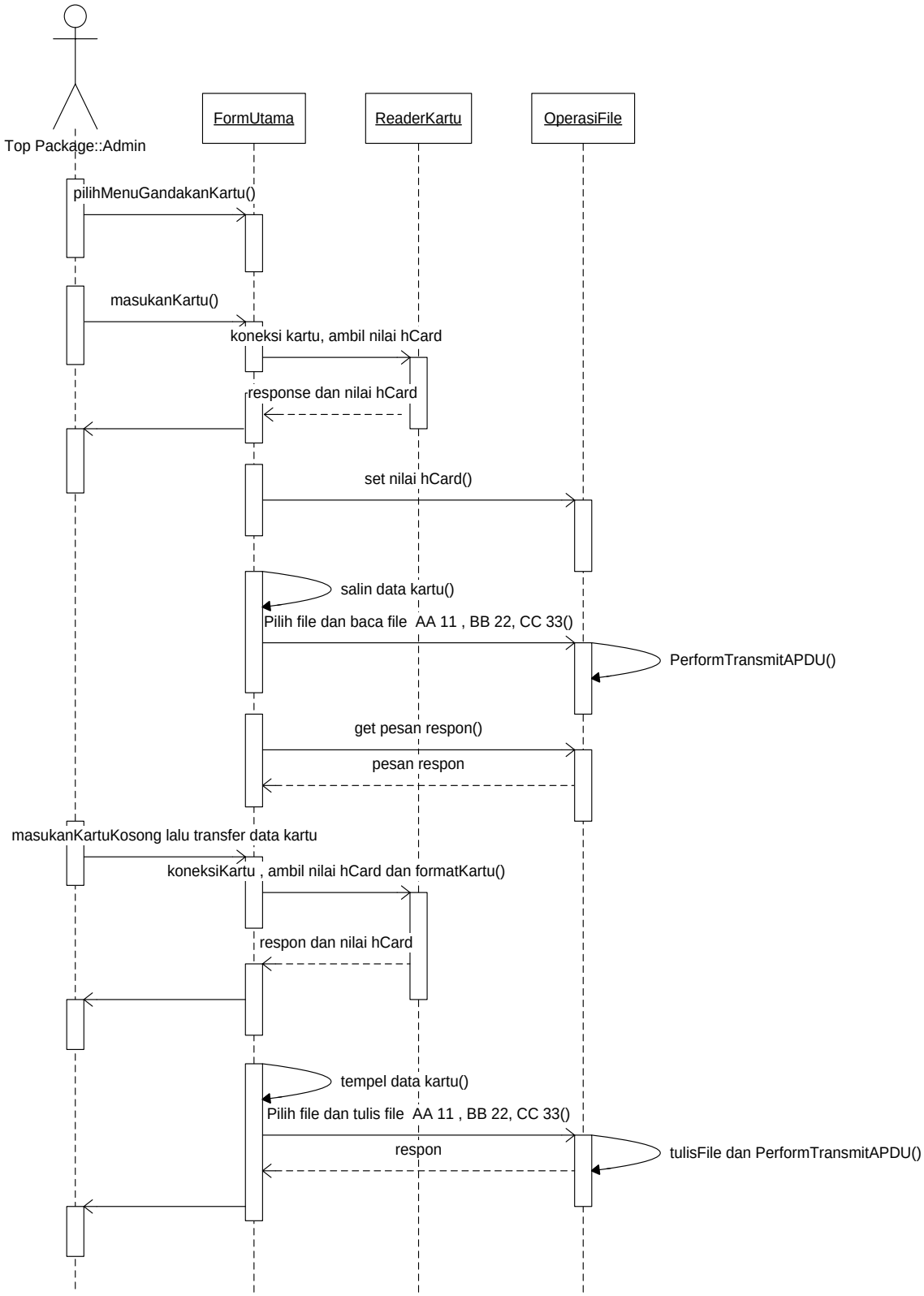
Dalam proses ini digambarkan proses saat melakukan pembacaan data dari *smart card*. Sebelum melakukan pembacaan data dari *smart card*, sistem terlebih dulu memilih file yang akan dibaca. Setelah file terpilih baru kemudian isi file tersebut dibaca kemudian untuk data bersifat privat didekripsi terlibih dahulu, lalu hasil dari pembacaan tersebut ditampilkan pada form.



Gambar 3.6 *Sequence Diagram* Baca Data

3.1.2.2.6 Gandakan Kartu

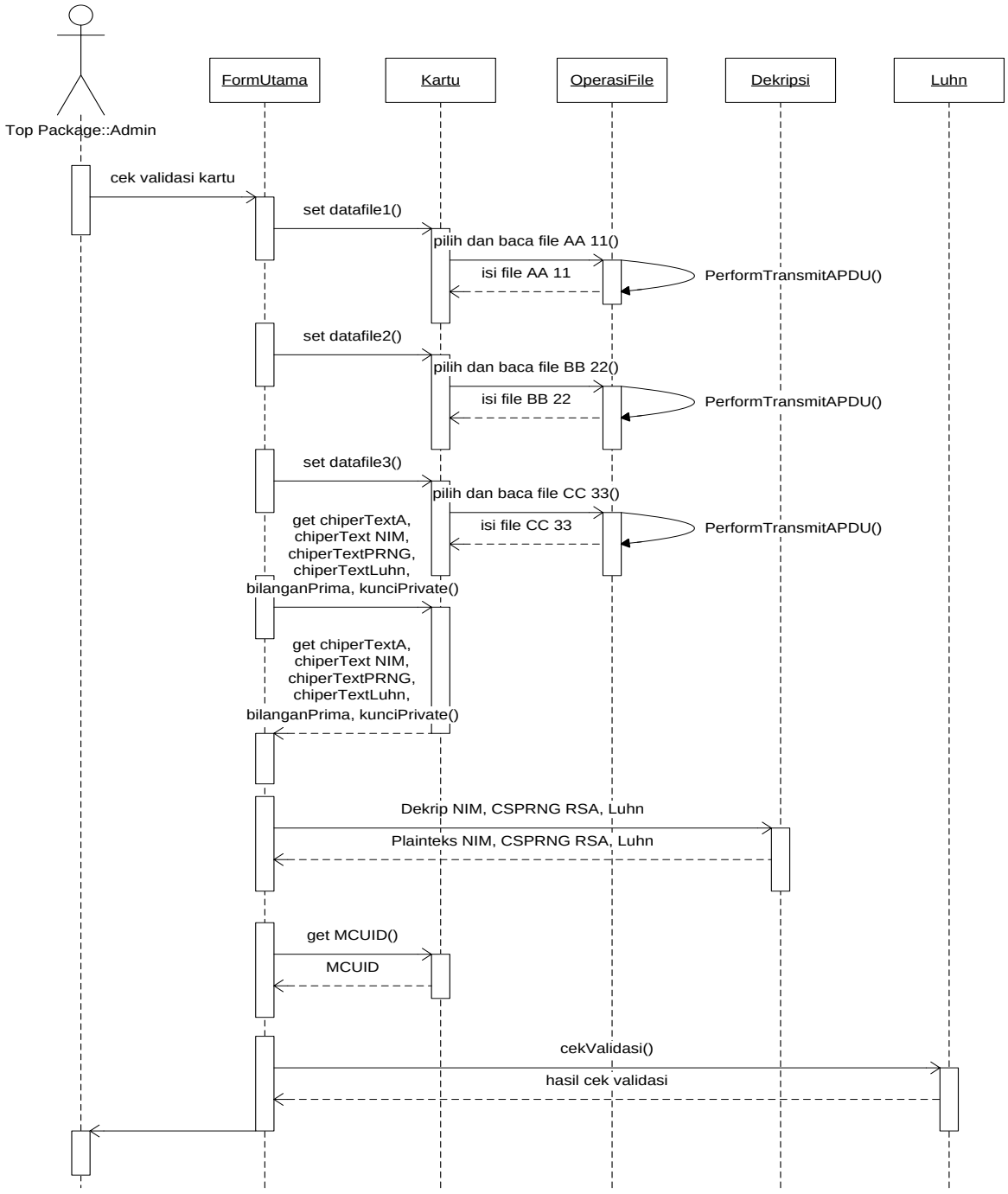
Dalam proses gandakan kartu akan digambarkan bagaimana informasi pada *smart card* dapat digandakan (sehingga *hacker* tidak perlu mendekripsi informasi). Informsai pada *smart card* yang dimasukan ke sistem akan secara otomatis dibaca kemudian data tersebut ditransefer ke kartu kosong.



Gambar 3.7 *Sequence diagram* Gandakan Kartu

3.1.2.2.7 Cek Security Kartu

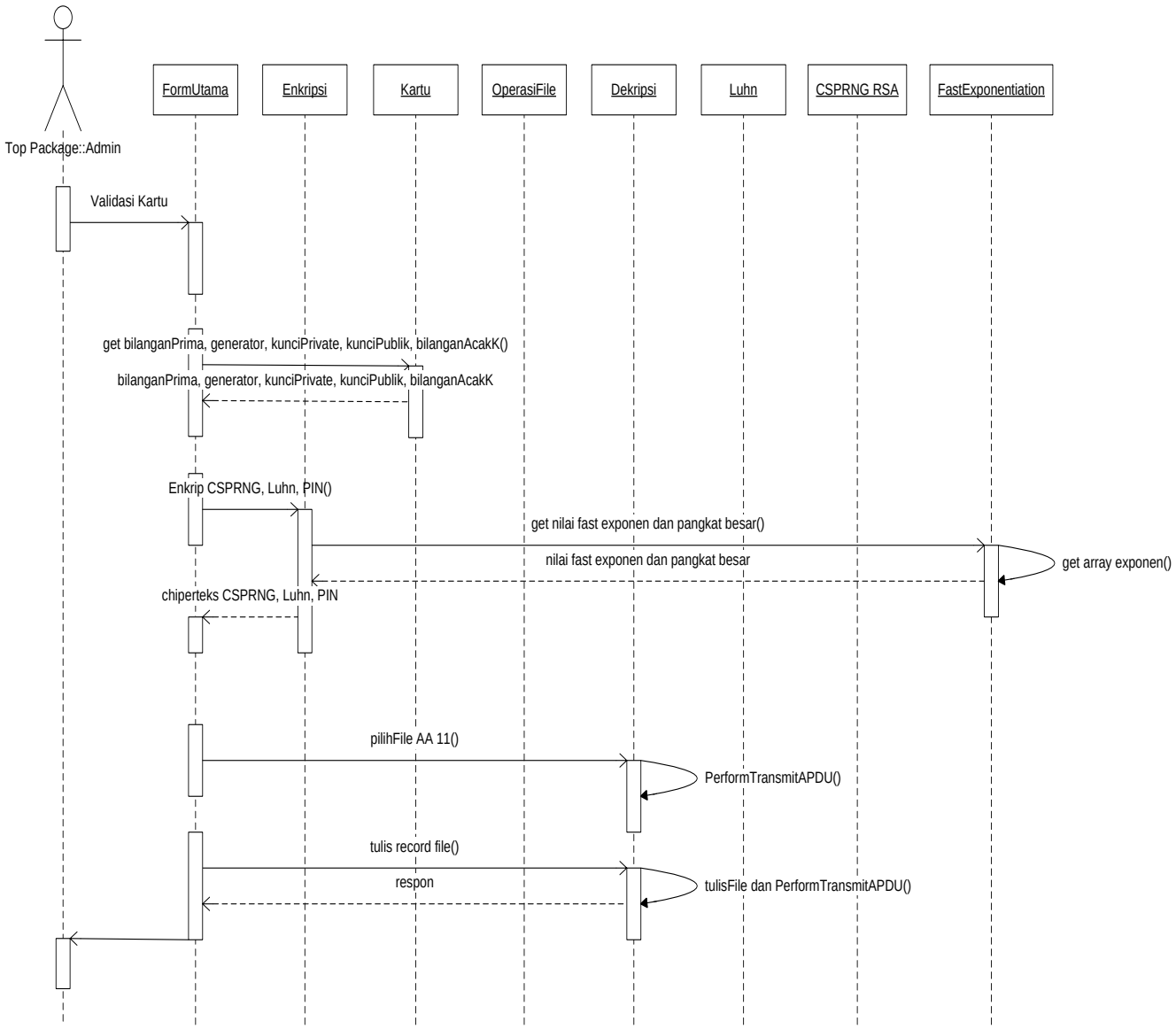
Dalam proses ini digambarkan bagaimana melakukan cek secutiry sebuah kartu.



Gambar 3.8 *Sequence diagram* Cek Security Kartu

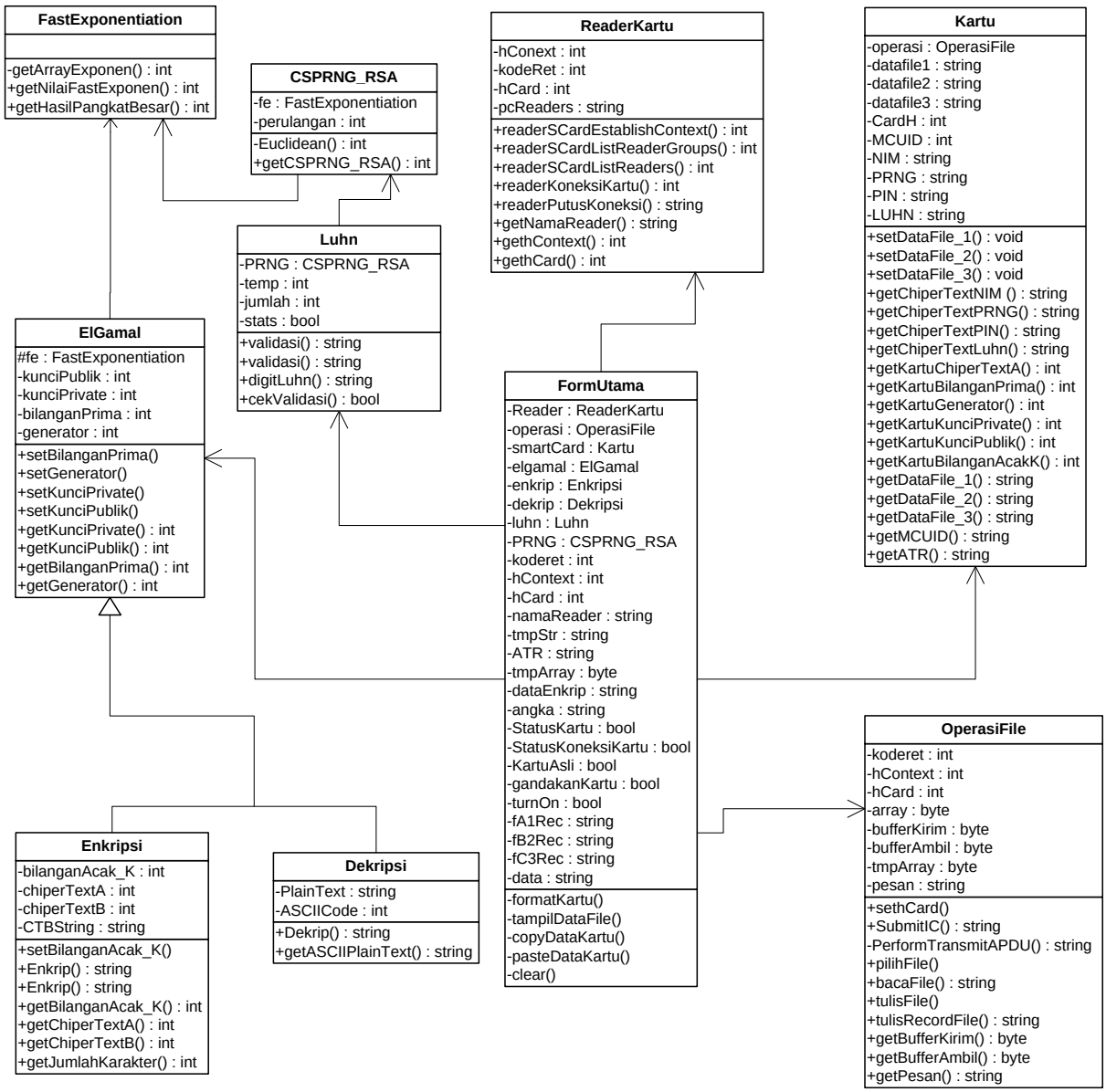
3.1.2.2.8 Securing Card

Berikut digambarkan urutan proses securing card. Disini akan diberikan security menggunakan *Luhn*. Konsep security yang akan digunakan adalah (*NIM*)(*MCUID*)(*CSPRNG_RSA*)(*LUHN*). Proses pertama-tama dilakukan dekripsi terhadap NIM kemudian ambila MCUID dari kartu, generate bilangan acak menggunakan CSPRNG RSA kemudian validasi menggunakan Luhn.



Gambar 3.9 *Sequence Diagram* Securing Card

3.1.2.3 *Class Diagram*



Gambar 3.10 *Class Diagram*

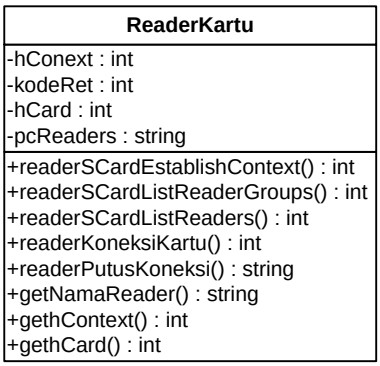
3.1.2.4 Spesifikasi kelas

Tabel 3.3 Spesifikasi Kelas

Nama kelas	Tipe		Nama kelas	Tipe
Kartu	Entitas		Elgamal	Kontrol
FormUtama	Interface		Enkripsi	Kontrol
ReaderKartu	Kontrol		Dekripsi	Kontrol
OperasiFile	Kontrol		Luhn	Kontrol
FastExponentiation	Kontrol		CSPRNG_RSA	Kontrol

3.1.2.4.1 Kelas ReaderKartu

Kelas ini digunakan untuk melakukan koneksi termasuk pemutusan koneksi pada *reader* dan *smart card.*



1. Deskripsi atribut

Nama	Tipe	Panjang	Range	Nilai Awal	Sifat
koderet	Int			0	Private
hContext	Int			0	Private
hCard	Int			0	Private
pcReaders	String		’a’..’Z’,’0’..’9’,-,/,spasi	null	private

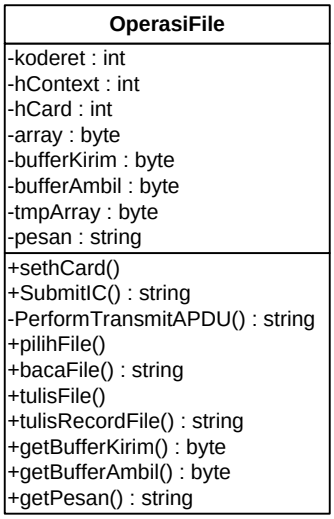
2. Deskripsi layanan

Operasi-operasi yang terdapat dalam kelas ini adalah :

1. Method berikut merupakan inisialisasi pengaturan *reader smart card.*
   1.1 readerSCardEstablishContext() berfungsi untuk membangun konteks
   1.2 readerSCardListReaderGroups() berfungsi untuk membuat gruplist reader yang akan digunakan.
   1.3 readerSCardListReaders() berfungsi untuk mengambil list PCSC card yang terinstall.
2. Method readerKoneksiKartu() berfungsi sebagai koneksi dari reader ke *smart card.*
3. Method readerPutusKoneksi() digunakan untuk memutuskan hubungan dengan kartu ketika selesai menggunakan kartu.
4. Method getNamaReader() digunakan untuk mengambil nama *reader smart card* yang terkoneksi dengan sistem.
5. Method gethContext() berfungsi untuk mengembalikan nilai pada *reader* yang terkoneksi.
6. Method gethCard() berfungsi untuk mengembalikan nilai pada *smart card* yang terkoneksi.

3.1.2.4.2 Kelas OperasiFile

Kelas ini digunakan untuk melakukan operasi-operasi dasar terutama juga yang berhubungan dengan file pada *smart card*.



1. Deskripsi atribut

Nama	Tipe	Panjang	Range	Nilai Awal	Sifat
koderet	Int			0	Private
hContext	Int			0	Private
hCard	Int			0	Private
array	byte	262	00 – FE		Private
bufferKurim	byte	262	00 – FE		Private
bufferAmbil	byte	262	00 – FE		Private
tmpArray	byte	56	00 – FE		Private
pesan	string	3	’a’..’Z’,’0’..’9’,spasi		Private

2. Deskripsi layanan

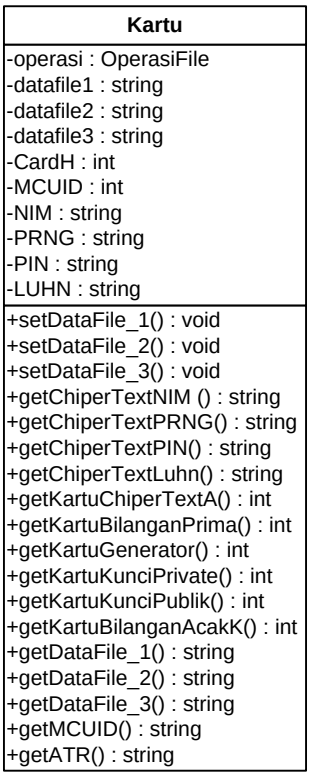
Operasi-operasi yang terdapat dalam kelas ini adalah :

1. Method sethCard(Hcard:int) berfungsi sebagai setter nilai variabel hCard dalam class ini.
2. Method SubmitIC() berfungsi sebagai pengirim *issuer code* ke *smart card* sebagai *protocol* dasar dalam melakukan komunikasi.

3. Method PerformTransmitAPDU(apdu:WinSCard) digunakan untuk mengirimkan protocol APDU agar dapat melakukan komunikasi dengan kartu.
4. Method pilihFile(AlAtas, AlBawah:byte) digunakan untuk memilih file yang akan digunakan dalam *smart card*.
5. Method bacaFile(posisiRecord, pjngData:byte) berfungsi untuk membaca file pada yang dipilih.
6. Method tulisFile(tipeCase:int, posisiRecord, pjngMax, pjngData, ApduIn:byte) berfungsi untuk menulis data pada file yang dipilih.
7. Method tulisRecordFile(dataText:string, recordNo, pjngData:byte) digunakan untuk menulis record pada sebuah file.
8. Method getBufferKirim() mengambil nilai buffer yang dikimkan.
9. Method getBufferAmbil() mengambil nilai buffer yang diterima.
10. Methos getPesan() berfungsi mengambil string pesan respon.

3.1.2.4.3 Kelas Kartu

Kelas ini digunakan untuk menampung data record dalam file, MCU ID dan ATR pada *smart card*.



1. Deskripsi atribut

Nama	Tipe	Range	Nilai Awal	Sifat
operasi	OperasiFile			Private
datafile1	String	’a’..’Z’,’0’..’9’,-,/,spasi		Private
datafile2	String	’a’..’Z’,’0’..’9’,-,/,spasi		Private
datafile3	String	’a’..’Z’,’0’..’9’,-,/,spasi		Private
CardH	Int		0	Private
MCUID	String	’a’..’Z’,’0’..’9’,-,/,spasi		Private
NIM	String	’a’..’Z’,’0’..’9’,-,/,spasi		Private
PRNG	String	’a’..’Z’,’0’..’9’,-,/,spasi		Private
PIN	String	’a’..’Z’,’0’..’9’,-,/,spasi		Private
LUHN	String	’a’..’Z’,’0’..’9’,-,/,spasi		Private

2. Deskripsi layanan

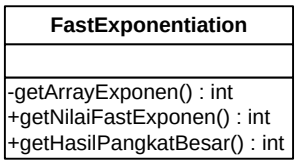
Operasi-operasi yang ada dalam kelas ini adalah :

1. Method setDatafile1() berfungsi untuk menampung data file 1.

2. Method setDatafile1()berfungsi untuk menampung data file 2.
3. Method setDatafile1()berfungsi untuk menampung data file 3.
4. Method getChiperTextNIM() mengembalikan string chiperteks NIM.
5. Method getChiperTextPRNG()mengembalikan string chiperteks PRNG.
6. Method getChiperTextPIN()mengembalikan string chiperteks PIN.
7. Method getChiperTextLuhn()mengembalikan string chiperteks Luhn.
8. Method getChiperTextA()mengembalikan string chiperteks A.
9. Method getBilanganPrima()mengembalikan string bilangan prima.
10. Method getGenerator()mengembalikan string generator.
11. Method getKunciPrivate()mengembalikan string kunci private.
12. Method getKunciPublik()mengembalikan string kunci publik.
13. Method getBilanganAcakK()mengembalikan string bilangan acak
14. Method getDataFile_1() mengembalikan array string file data.
15. Method getDataFile_2() mengembalikan array string file data.
16. Method getDataFile_3() mengembalikan array string file data.
17. Method getMCUID() mengembalikan nilai MCU ID *smart card*.
18. Method getATR()mengembalikan nilai ATR *smart card*.

3.1.2.4.4 Kelas FastExponentiation

Kelas ini dimanfaatkan sebagai metode dalam menghitung bilangan pangkat besar menggunakan metode *Fast Exponentation* ketika melakukan perhitungan pembentukan pasangan kunci dan sebagainya.



1. Deskripsi atribut

Tidak ada atribut dalam kelas ini.

2. Deskripsi layanan

Operasi-operasi yang ada dalam kelas ini adalah :

1. Method berikut digunakan untuk mengembalikan nilai dalam perhitungan diantaranya :

   1.1 getArrayExponen(target : int) digunakan untuk mendapatkan nilai array.

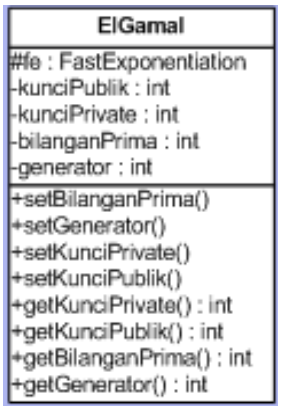
   1.2 getNilaiFastExponen(bilangan, pangkat, modulo:int) digunakan untuk mendapatkan nilai hasil perhitungan fast exponentiation.

   1.3 getHasilPangkatBesar(KunciPublic, BilanganPrima, k : int) digunakan untuk mendapatkan nilai hasil pemangkatan besar.

Setiap method tersebut menggunakan parameter, sehingga langsung menghasilkan nilai tanpa perlu sebelumnya melakukan set bilangan.

3.1.2.4.5 Kelas ElGamal

Kelas ini digunakan sebagai inisilasisasi pembentukan pasangan kunci dan beberapa elemen yang digunakan dalam kriptografi.



1. Deskripsi atribut

Nama	Tipe	Panjang	Range	Nilai Awal	Sifat
fe	Fast Exponentiation				Protected
kunciPublik	Int	4		0	Private
kunciPrivate	Int	4	1 s.d. (bilanganPrima – 1)	0	Private
bilangan Prima	Int	4	10000	0	Private
generator	Int	1	2-5	0	Private

2. Deskripsi layanan

Operasi-operasi yang terdapat dalam kelas ini adalah :

1. Method berikut digunakan sebagai inisialisasi bilangan :

   1.1 setBilanganPrima()

   1.2 setGenerator()

   1.3 setKunciPrivate()

   1.4 setKunciPublik()

2. method berikut digunakan untuk mengembalikan nilai hasil inisialisasi :

   2.1 getKunciPrivate()

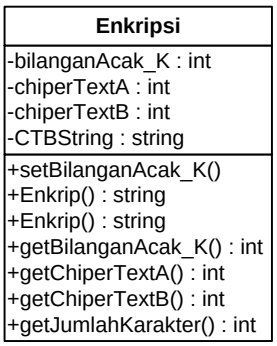
   2.2 getKunciPublik()

   2.3 getBilanganPrima()

2.4 getGenerator()

3.1.2.4.6 Kelas Enkripsi

Kelas ini digunakan untuk melakukan operasi enkripsi.



1. Deskripsi atribut

Nama	Tipe	Panjang	Range	Nilai Awal	Sifat
bilanganAcak_K	Int	4	0 – (bilanganPrima – 1)	0	Private
chiperTextA	Int			0	Private
chiperTextB	Int			null	Private
CTBString	String		’0’..’9’,’;’		Private

2. Deskripsi layanan

Operasi-operasi yang terdapat dalam kelas ini adalah :

1. Method setbilanganAcak_K(bilanganprima:int) digunakan untuk membuat bilangan acak yang digunakan dalam operasi enkripsi.
2. Method Enkrip(plaintext:string) merupakan method yang digunakan untuk melakukan enkripsi.
3. Method Enkrip(plaintext:string, generator, bilanganPrima, kunciPublic, bilanganAcak:int) merupakan method overload yang digunakan untuk melakukan enkripsi.

4. Method berikut digunakan untuk mengembalikan nilai dalam operasi enkripsi diantaranya :

   3.1 getBilanganAcak_K()

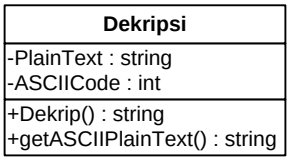
   3.2 getChiperTextA()

   3.3 getChiperTextB()

   3.4 getJumlahKarakter()

3.1.2.4.7 Kelas Dekripsi

Kelas ini digunakan untuk melakukan operasi dekripsi.



1. Deskripsi atribut

Nama	Tipe	Panjang	Range	Nilai Awal	Sifat
PlainText	String		’a’..’Z’,’0’..’9’ ,spasi		Private
ASCIICode	Int			0	Private

2. Deskripsi layanan

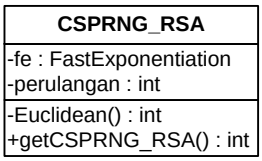
Operasi-operasi yang terdapat dalam kelas ini adalah :

1. Method Dekrip(chiperTextA:int, chiperTextB:string, bilanganPrima, kunciPrivate:int) digunakan untuk melakukan operasi dekripsi.
2. Method berikut digunakan untuk mengembalikan nilai hasil operasi dekripsi diantaranya :

   2.1 getASCIIPlainText() mengembalikan nilai string plainteks dalam bentuk ASCII code.

3.1.2.4.8 Kelas CSPRNG RSA

Kelas ini digunakan sebagai generator bilangan acak yang aman untuk kriptografi dalam operasi pembuatan nomor kartu.



1. Deskripsi atribut

Nama	Tipe	Panjang	Range	Nilai Awal	Sifat
fe	FastExponentiation				Private
Perulangan	Int			0	Private

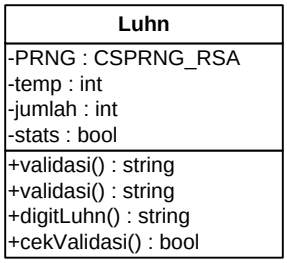
2. Deskripsi layanan

Operasi-operasi yang terdapat dalam kelas ini adalah :

1. Method Euclidean(aa,bb:int) digunakan untuk menentukan faktor persekutuan terbesar (*greatest common divisor*) dalam menentukan *pseudorandom generator* menggunakan metode Euclide.
2. Method getCSPRNG_RSA(jmlDigit:int) digunakan untuk mengembalikan nilai *random number* yang telah dibuat.

3.1.2.4.9 Kelas Luhn

Kelas ini digunakan untuk melakukan pembuatan bilangan luhn dan operasi validasi pada *smart card*.



1. Deskripsi atribut

Nama	Tipe	Panjang	Range	Nilai Awal	Sifat
PRNG	CSPRNG RSA				Private
temp	Int			0	Private
jumlah	Int			0	Private
stats	bool			true	

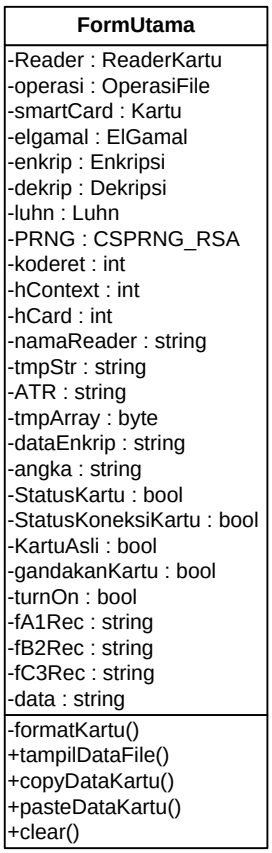
2. Deskripsi layanan

Operasi-operasi yang terdapat dalam kelas ini adalah :

1. Method validasi(jmlDigitAngka:int) digunakan untuk membuat no kartu dengan jumlah yang ditetapkan dalam parameter.
2. Method validasi(digitAngka:string) digunakan untuk membuat nomor kartu baru.
3. Method digitLuhn(noKartu:string) berfungsi untuk membuat digit Luhn.
4. Method cekValidasi(angka:string) berfungsi untuk melakukan cek security kartu pada *smart card.*

3.1.2.4.10 Kelas FormUtama

Kelas ini digunakan sebagai pengendali utama program



1. Deskripsi atribut

Nama	Tipe	Panjang	Range	Nilai Awal	Sifat
Reader	ReaderKartu				Private
operasi	OperasiFile				Private
smartCard	Kartu				Private
elgamal	ElGamal				Private
enkrip	Enkripsi				Private
dekrip	Dekripsi				Private
luhn	Luhn				Private
PRNG	CSPRNG_RSA				Private
koderet	Int			0	Private
hContext	Int			0	Private
hCard	Int			0	Private
namaReader	String		’a’..’Z’ , spasi		Private
tmpStr	String		’a’..’Z’ , spasi		Private
ATR	String		’a’..’Z’ , spasi		Private
tmpArray	Byte	56		Null	Private

dataEnkrip	String		'a'..'Z' , spasi		Private
angka	String		'a'..'Z' , spasi		Private
statusKartu	Bool			True	Private
StatusKoneksi Kartu	Bool			True	Private
KartuAsli	Bool			True	Private
gandakanKartu	Bool			True	Private
turnOn	Bool			True	Private
fA1Rec	String	7	'a'..'Z' , spasi		Private
fB2Rec	String	7	'a'..'Z' , spasi		Private
fC3Rec	String		'a'..'Z' , spasi		Private
Data	String		'a'..'Z' , spasi		Private

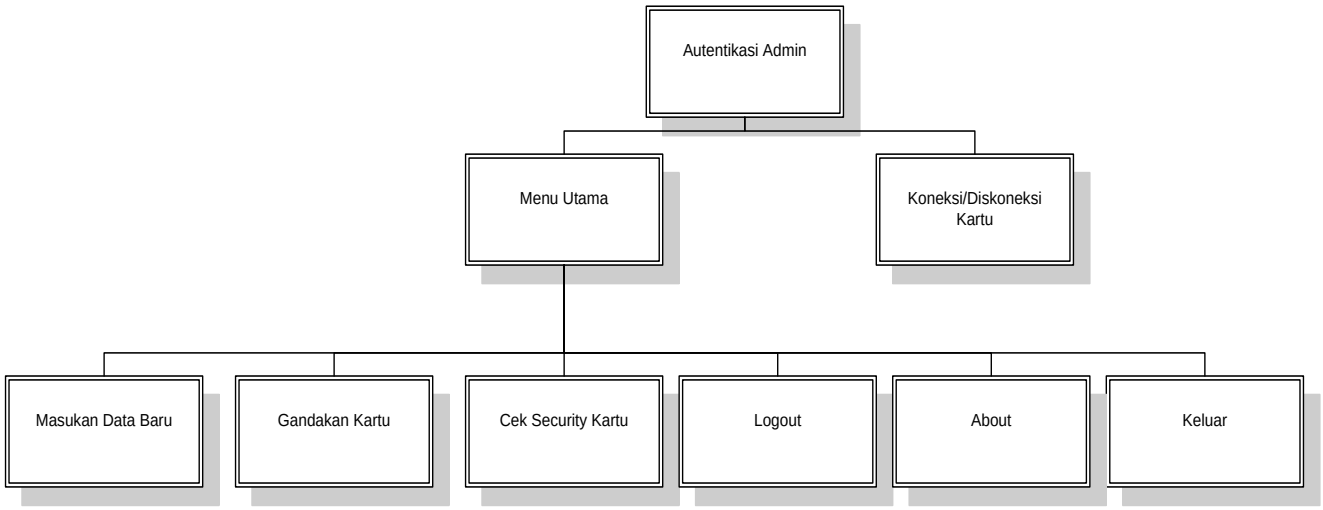
2. Deskripsi layanan

Operasi-operasi yang terdapat dalam kelas ini adalah :

1. Method formatKartu() berfungsi sebagai fase *personalization bit* pada *smart card.*
2. Method tampilDataFile() berfungsi untuk menampilkan dari file dalam smart card pada textfield.
3. Method copyDataKartu() menyalin isi record dalam file pada *smart card*
4. Method pasteDataKartu() melakukan paste (tempel) data dari file record yang telah dicopy dari kartu asli ke kartu kosong.
5. Method Clear() melakukan pembersihan textfield.

3.2 Perancangan Antarmuka Sistem

Gambar dibawah ini menunjukan perancangan arsitektur menu dari pembangunan sistem keamanan *smart card*.



Gambar 3.11 Perancangan Arsitektur Menu

Autentikasi adalah proses untuk memastikan hak akses kepada admin agar dapat mengoperasikan sistem. Disini admin memasukan *smart card* lalu kemudian masukan password. Jika benar, maka sistem dapat digunakan.

Menu Utama terdiri dari :

1. ***Masukan Data Baru,*** adalah submenu yang berfungsi sebagai proses penginputan data baru pada kartu. Ketika input data baru, maka secara otomatis sistem akan melakukan enkripsi terhadapt NIM.

2. ***Gandakan Kartu,*** adalah submenu yang berfungsi untuk melakukan penggandaan kartu. Ini dilakukan sebagai contoh ketika suatu data dalam kartu berhasil digandakan.

3. ***Cek Security Kartu,*** adalah submenu yang berfungsi untuk melakukan pengecekan security kartu dan securing card. Securing card dilakukan untuk menghindari tipe serangan *chosen-chipertext attack* oleh kriptanalis.

4. ***Keluar,*** adalah submenu yang berfungsi untuk keluar dari aplikasi.

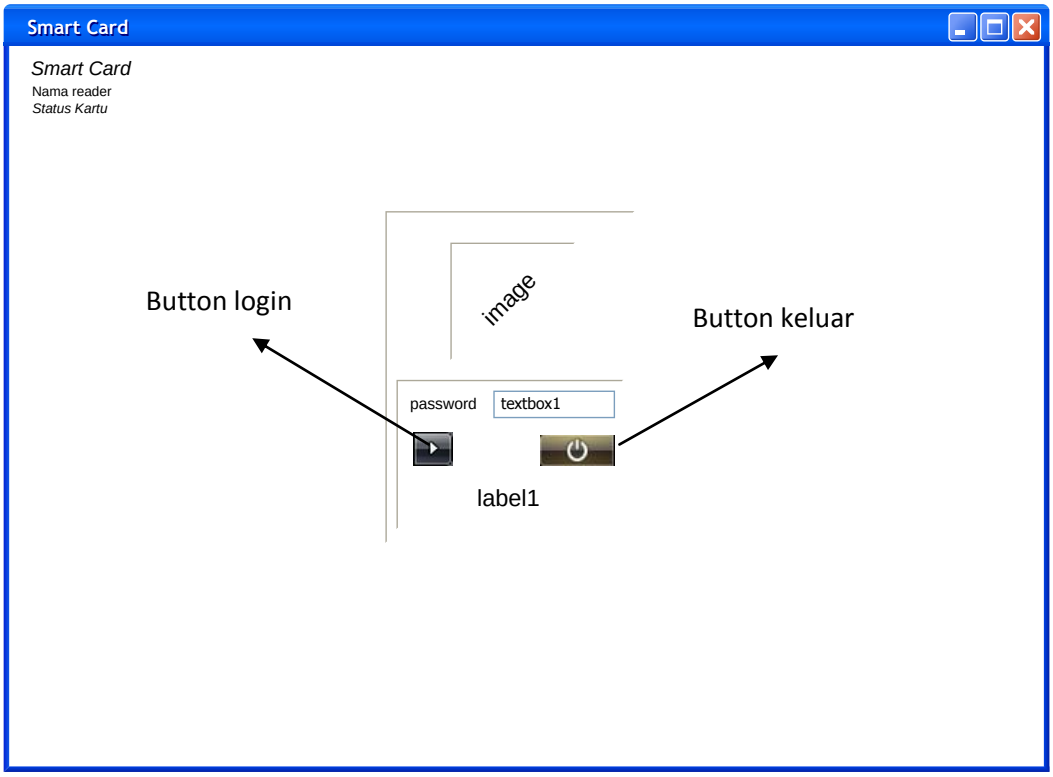
5. ***Logout,*** adalah submenu yang berfungsi untuk keluar dari sesi log setelah admin melakukan autentikasi.

Koneksi/Diskoneksi Kartu, berfungsi sebagai pengatur koneksi dan diskoneksi pada kartu yang terdeteksi oleh *reader*. Ini dilakukan ketika akan melakukan operasi-operasi dasar seperti melakukan *read/write* data pada kartu.

Adapun beberapa layar (*screen*) dan informasi pesan yang terlibat dalam pembangunan aplikasi adalah sebagai berikut :

1. Autentikasi
2. Menu utama
3. Masukan Data Baru
4. Cek Security Kartu
5. Gandakan Kartu
6. Tentang

3.2.1 Autentikasi



Gambar 3.12 Antarmuka Autentikasi

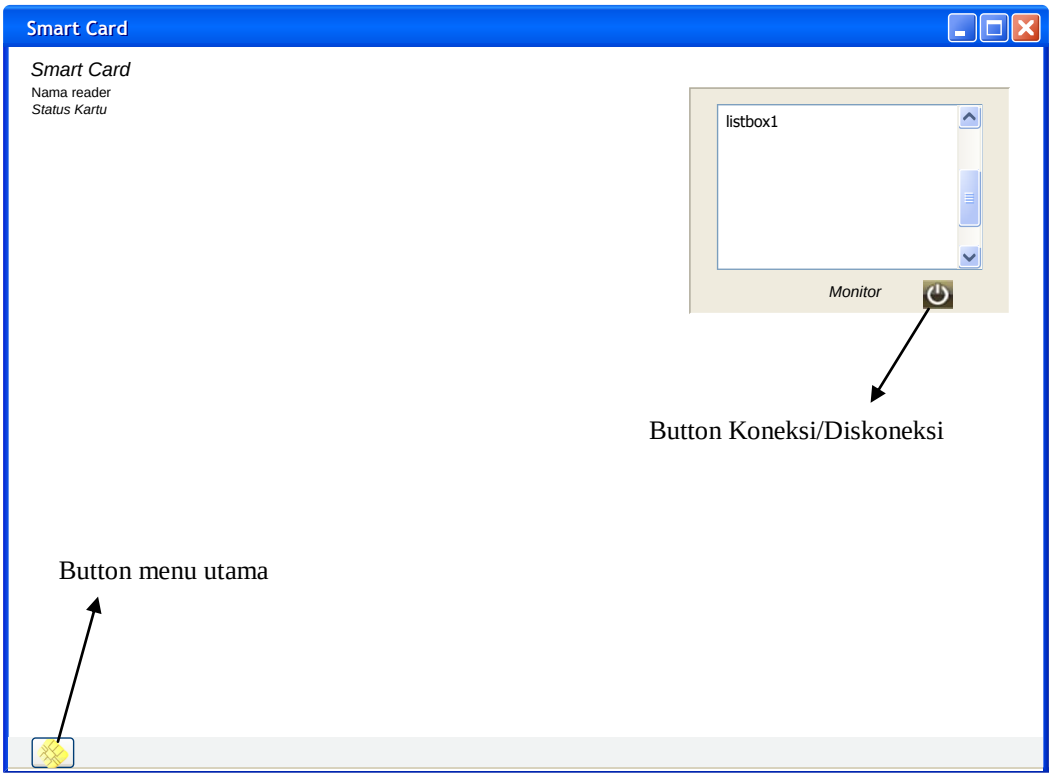
Keterangan nama dan fungsi tombol pada rancangan antarmuka autentikasi

Tabel 3.4 Keterangan Autentikasi

No	Nama komponen	Fungsi
1	Textbox1	Menampung string *password*
2	Button login	Tombol proses untuk password jika berhasil menuju menu utama, jika gagal kembali pada antarmuka autentikasi
3	Button keluar	Tombol untuk keluar dari aplikasi
4	Label1	Menampilkan pesan autentikasi gagal
2	Label nama reader	Menampilkan *reader* yang terkoneksi/terinstall
3	Label status kartu	Menampilkan status kartu yang terdeteksi

		oleh *reader*

3.2.2 Menu Utama



Gambar 3.13 Atarmuka Menu Utama

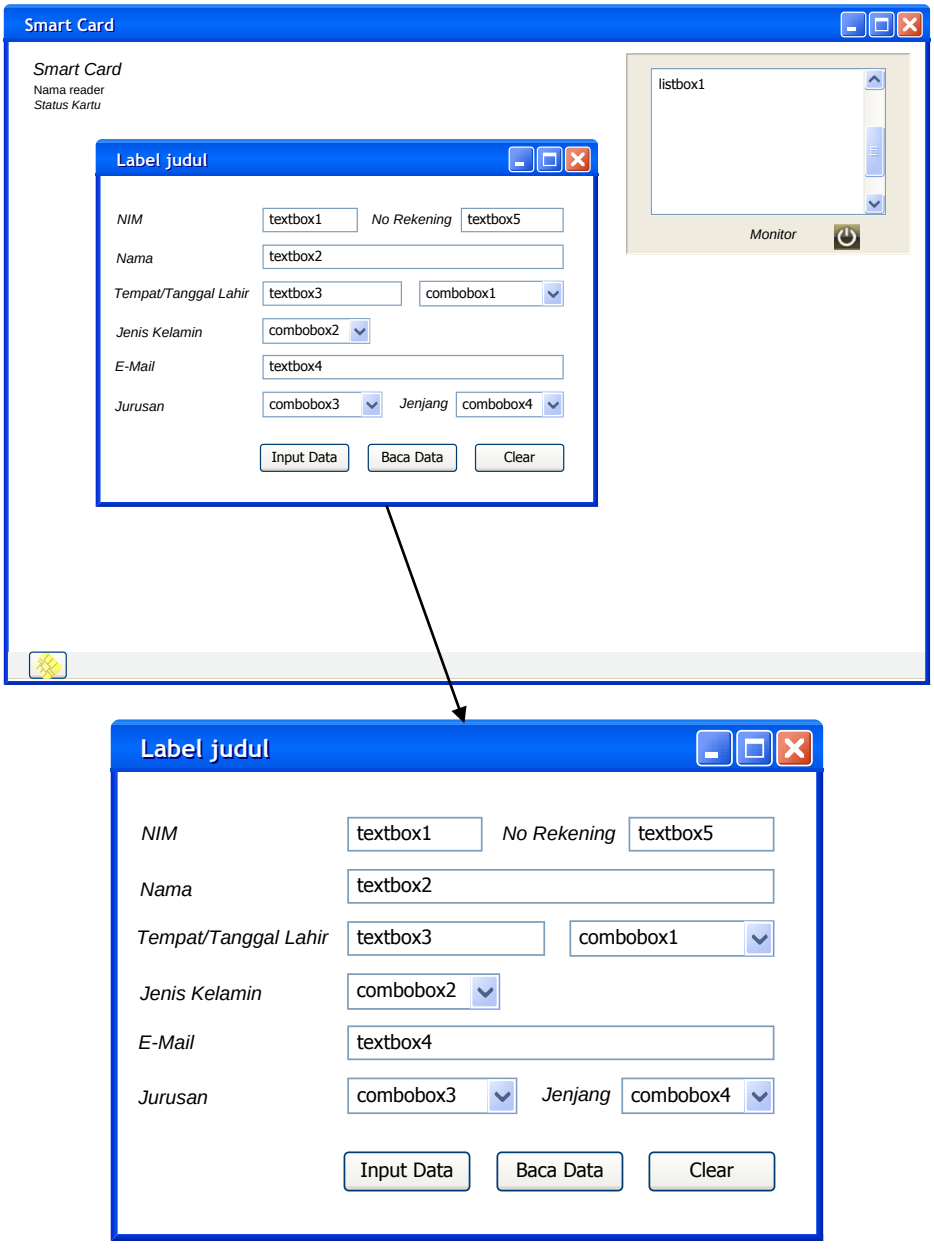
Keterangan nama dan fungsi tombol pada rancangan antarmuka Menu Utama

Tabel 3.5 Keterangan Menu Utama

No	Nama komponen	Fungsi
1	Button Menu Utama	Menampilkan Menu Utama
2	Button Koneksi/Diskoneksi	Melakukan proses koneksi/diskoneksi dari *reader* ke kartu.
3	Listbox1	Menampilkan aksi-aksi yang dilakukan terhadapt *smart card*, data hasil enkripsi, data yang tersimpan dalam *smart card*, respon terhadap aksi-aksi tersebut, dll.

3	Label nama reader	Menampilkan *reader* yang terkoneksi/terinstall
4	Label status kartu	Menampilkan status kartu yang terdeteksi oleh *reader*

3.2.3 Masukan Data Baru



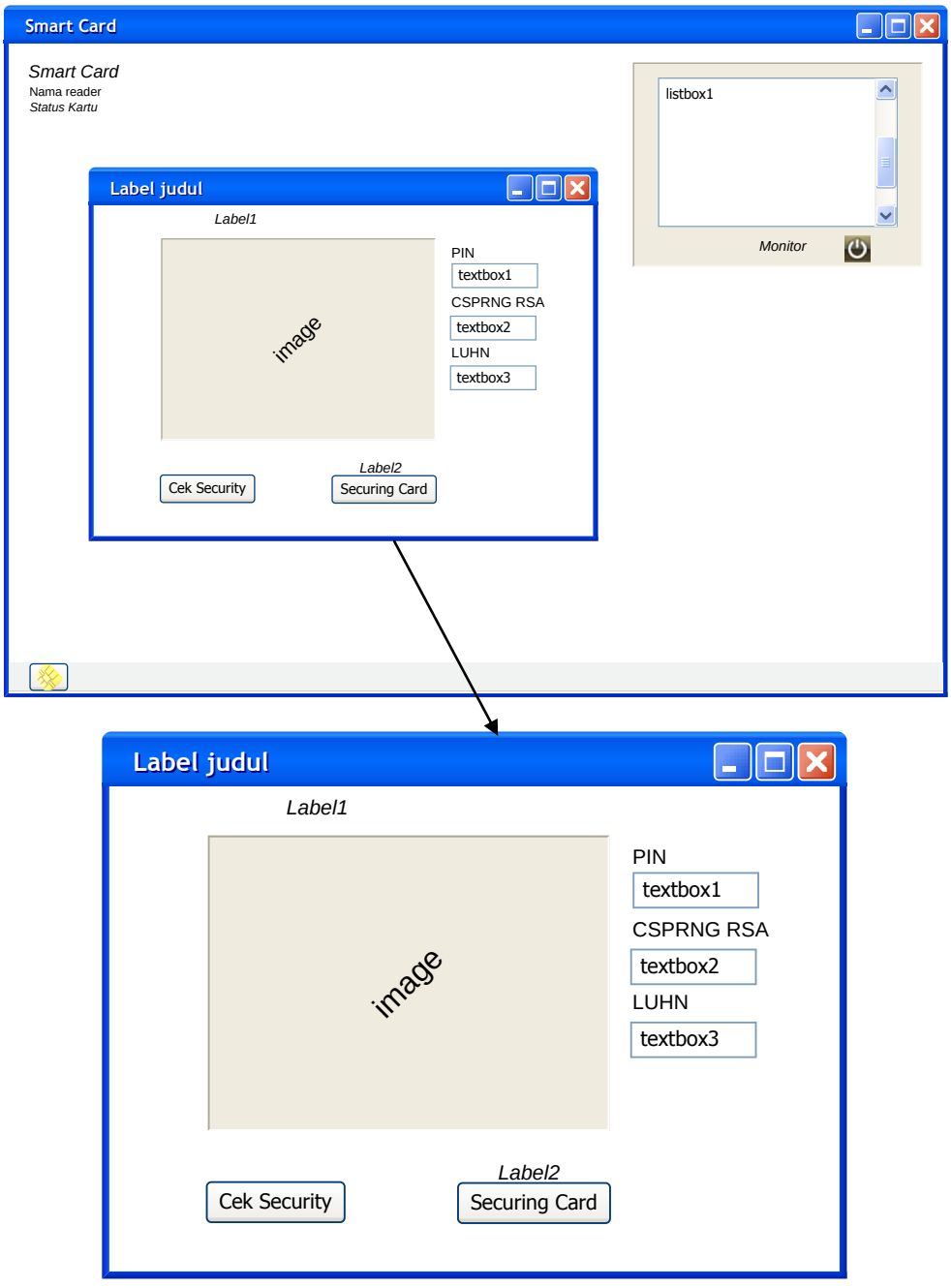
Gambar 3.14 Antarmuka Masukan Data Baru

Keterangan nama dan fungsi tombol pada rancangan antarmuka Masukan Data Baru.

Tabel 3.6 Keterangan Masukan Data Baru

No	Nama komponen	Fungsi
1	Textbox1	Menampung string **NIM**
2	Textbox2	Menampung string **Nama**
3	Textbox3	Menampung string **tempat lahir**
4	Combobox1	Combobox untuk memilih **tanggal**,**bulan** dan **tahun lahir**
5	Combobox2	Combobox untuk memilih jenis kelamin
6	Textbox4	Menampung **email**
7	Combobox3	Combobox untuk memilih **jurusan**
8	Combobox4	Combobox untuk memilih **jenjang**
9	Textbox5	Menampung string **no rekening**
10	Button Input data	Tombol proses untuk menginputkan data pada kartu
11	Button Baca data	Tombol proses untuk membaca data dari kartu
12	Button Clear	Tombol untuk mengembalikan/ mengosongkan textbox dan combobox ke kondisi awal

3.2.4 Cek Security Kartu



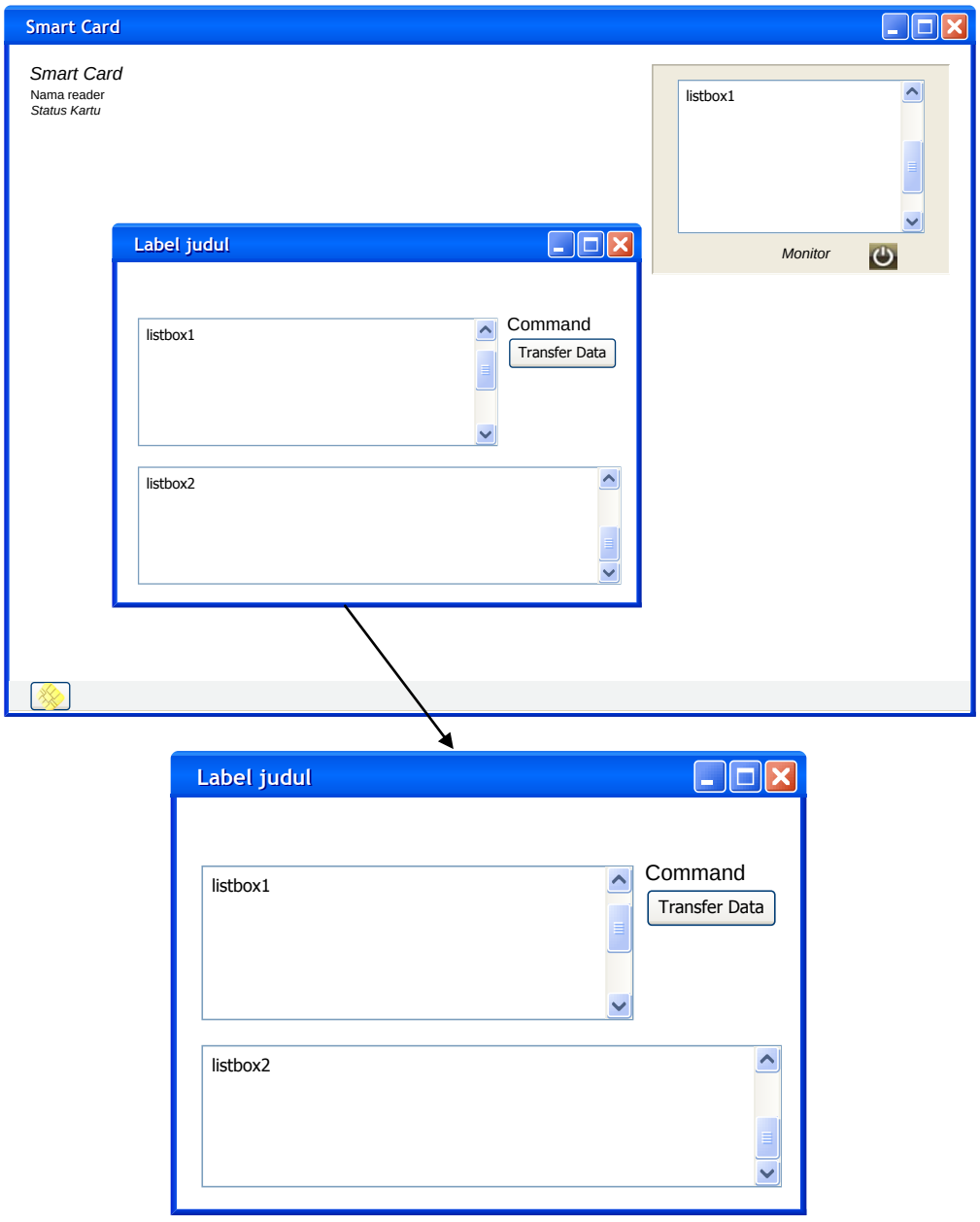
Gambar 3.15 Antarmuka Cek Security Kartu

Keterangan nama dan fungsi tombol pada rancangan antarmuka Cek security Kartu.

Tabel 3.7 Keterangan Cek Security Kartu

No	Nama komponen	Fungsi
1	Textbox1	Menampung string **NIM** yang telah dienkripsi yang diambil dari kartu
2	Textbox2	Menampung string **CSPRNG RSA** yang telah dienkripsi yang diambil dari kartu
3	Textbox3	Menampung string **LUHN** yang telah dienkripsi yang diambil dari kartu
4	Button cek security	Tombol untuk menjalankan proses menguji security dari kartu
5	Button securing card	Tombol untuk melakukan proses pemberian keamanan pada kartu
6	Label1	Menampilkan status cek security, apakah kartu secure atau tidak.
7	Label2	Menampilkan **PIN** default yang digenerate secara acak oleh sistem setelah dilakukan security terhadap kartu.

3.2.5 Gandakan Kartu



Gambar 3.16 Antarmuka Gandakan Kartu

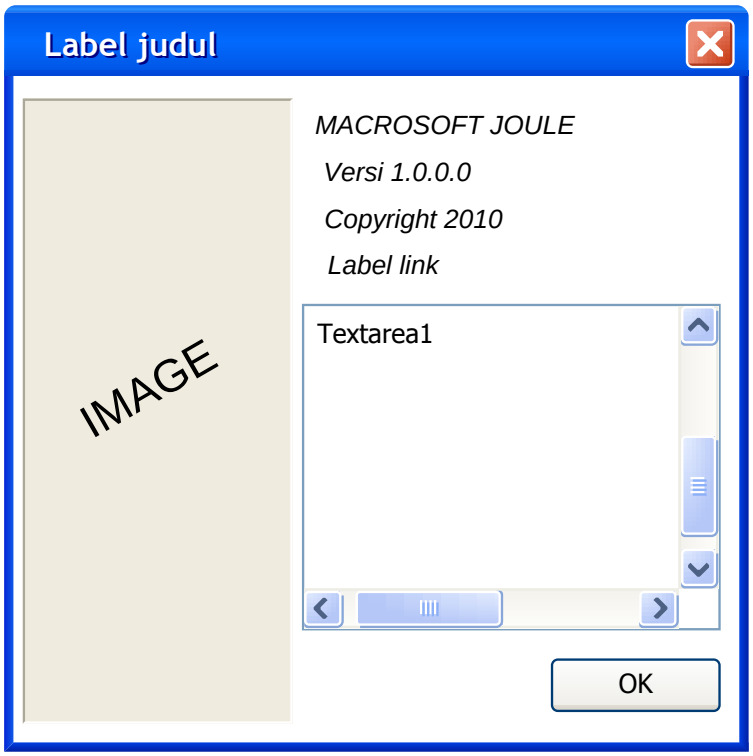
Keterangan nama dan fungsi tombol pada rancangan antarmuka Gandakan Kartu

Tabel 3.8 Keterangan Gandakan Kartu

No	Nama komponen	Fungsi

1	Listbox1	Menampilkan proses pemilihan file, buffer yang dikirimkan dan buffer yang diterima
2	Listbox2	Menampilkan isi record file
3	Button Transfer Data	Tombol perintah untuk melakukan proses transfer data ke kartu kosong/lain.

3.2.6 Tentang



Gambar 3.17 Atarmuka Tentang

Keterangan nama dan fungsi tombol pada rancangan antarmuka Tentang

Tabel 3.9 Keterangan Tentang

No	Nama komponen	Fungsi
1	Button OK	Keluar dari form about
2	Textarea1	Menampilkan deskripsi aplikasi