# Perancangan Aplikasi Enkripsi Dan Dekripsi SMS Dengan Algoritma Simetri AES Pada Telepone Seluler

**Naskah Publikasi**


**Diajukan oleh:**

**Miftahul Choiril Huda**

**06.11.1129**


**Kepada :**


**JURUSAN TEKNIK INFORMATIKA**
**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER**
**AMIKOM**
**YOGYAKARTA**


**2010**

**Naskah Publikasi**

# Perancangan Aplikasi Enkripsi Dan Dekripsi SMS Dengan Algoritma Simetri AES Pada Telepon Seluler


Disusun oleh:

**Miftahul Choiril Huda**

**06.11.1129**


Dosen Pembimbing,

**Emha Taufiq Luthfi, S.T, M.Kom**

**NIK. 190302125**

Tanggal 29 Mei 2010

**Ketua Jurusan**

**Teknik Informatika**

**Ir. Abas Ali Pangera, M.Kom.**

**NIK. 190302008**

**Designing An Application Of Encryption And Decryption SMS By Symmetry Algorithm AES On Cellular Phone**

**Perancangan Aplikasi Enkripsi Dan Dekripsi SMS Dengan Algoritma Simetri AES Pada Telepone Seluler**

Miftahul Choiril Huda

Teknik Informatika

STMIK Amikom Yogyakarta

*Communication is a mekanism to convey information from one party to another party. In line as growing of technology, a need of communication media increased. Currently, a variety of communications technologies and increasingly pamper people. Starting from telephone, facsimile, e-mail, and many others such as SMS technology.*

*SMS technology to this day remains a popular medium of communication by the people, besides easy to use it is also cheaper cost. But on the other side of the SMS technology also has disadvantages. SMS technology does not guarantee the security and confidentiality of messages sent. Some risks are also a threat to security including SMS spoofing, SMS snooping , and SMS interception. From some of the threats to the risk of such an SMS message, it is necessary to build an application that is able to secure and keep confidential SMS messages, so that in the event of a threat and that message is opened, the contents of the message remains secret. One solution to secure and keep the message is to encrypt SMS messages before sending.*

*Making encryption and decryption SMS application is expected to be the solution of the problems above. In addition to securing and keeping the message, the application built to be easy to use (userfriendly) by people who using and take advantage of this application.*

**Keywords:** SMS, encryption, decryption, AES, BouncyCastle.

## 1 Latar Belakang Masalah

Sejalan dengan perkembangan teknologi informasi, saat ini orang dapat saling berkomunikasi dengan cepat dan mudah, sehingga orang akan dapat mengetahui apa saja yang terjadi di belahan bumi ini dengan bertukar informasi. Proses komunikasi sendiri senantiasa terjadi setiap waktu yang tidak terbatas oleh jarak. Hingga pada ahirnya berbagai perangkat alat komunikasi dikembangkan untuk menunjang tercapainya proses tersebut.

SMS merupakan layanan pengiriman pesan singkat dalam jaringan komunikasi. Metode komunikasi SMS ini banyak digunakan orang karena selain biaya yang murah, penggunaannya pun mudah dan cepat. Namun, di samping kemudahan-kemudahan yang diberikan, layanan SMS ini juga memiliki kelemahan yaitu keamanan data. Secara umum SMS tidak menjamin kerahasiaan dan keutuhan pesan yang dikirimkan oleh pengguna. Oleh karena pesan-pesan yang dikirimkan pengguna terkadang merupakan pesan yang rahasia dan pribadi, sehingga kerahasiaan pesan menjadi sangat penting untuk dijaga dari orang-orang yang tidak berhak menerimanya.

## 2 Kriptografi

Kriptografi (cryptography) berasal dari bahasa Yunani yaitu cryptos artinya rahasia (secret) dan graphein artinya tulisan (writing). Jadi kriptografi berarti tulisan rahasia (secret writing). Secara istilah kriptografi didefinisikan sebagai ilmu sekaligus seni untuk menjaga kerahasiaan pesan (data atau informasi) yang mempunyai arti atau nilai, dengan cara menyamarkannya (mengacak) menjadi bentuk yang tidak dapat dimengerti menggunakan suatu algoritma tertentu.

### 2.1 AES (Advanced Encryption Standard)

*Advance Encryption Standar* dipublikasikan oleh Nist (National Institute of Standard and Technology) pada tahun 2001. AES merupakan blok kode simetris untuk menggantikan Des (Data Encryption Standar). AES diperoleh dari hasil kompetisi yang diadakan NIST pada tahun 1997. Pada tahap pertama, 15 peserta dari 21 peserta lolos ke tahap berikutnya berdasarkan penilaian tingkat keamanan, harga, algoritma, dan karakteristik implementasi. Sepuluh dari 15 peserta tersebut gugur pada tahap berikutnya karena dianggap kurang aman dan kurang efektif. Pada Agustus 1999 dipilih lima kandidat untuk seleksi akhir, yaitu Mars (IBM, Amerika Serikat), RSA (RSA corp., Amerika Serikat), Rijndael (Belgia), Serpent (Israel, Norwegia, dan Inggris), dan Twofish (Counterpane, Amerika Serikat). Akhirnya, pada tanggal 2

Oktober 2000 terpilihlah algoritma Rijndael, yang dibuat oleh Dr. Vincent Rijment dan Dr. Joan Daemen, sebagai pemenang.

## 2.2 Algoritma Advanced Encryption Standard (AES)

AES merupakan algoritma block cipher dengan menggunakan sistem permutasi dan substitusi (P-Box dan S-Box). Berdasarkan panjang kunci yang digunakan AES dikelompokkan menjadi 3 bagian, Angka-angka di belakang kata AES menggambarkan panjang kunci yang digunakan pada tipa-tiap AES. Selain itu, hal yang membedakan dari masing-masing AES ini adalah banyaknya round yang dipakai. AES-128 menggunakan 10 round, AES-192 sebanyak 12 round, dan AES-256 sebanyak 14 round. Berikut adalah tabel jumlah putaran yang digunakan algoritma ini.

**Tabel 2.1 jumlah putaran operasi pada AES**

| Tipe | Panjang Kunci | Panjang Blok Input | Jumlah putaran |
|---|---|---|---|
| AES-128 | 4 words (128 bit) | 4 words (128 bit) | 10 |
| AES-192 | 6 words (192 bit) | 4 words (128 bit) | 12 |
| AES-256 | 8 words (256 bit) | 4 words 1(28 bit) | 14 |

Garis besar algoritma Rijndael yang beroperasi blok 128-bit dengan kunci 128-bit adalah sebagai berikut:

1. AddRoundKey:

   melakukan XOR antara state awal (plainteks) dengan cipher key. Tahap ini disebut juga initial round.

2. Putaran sebanyak Nr – 1 kali.

Proses yang dilakukan pada setiap putaran adalah:

1. ByteSub:

   substitusi byte dengan menggunakan tabel substitusi (S-box).

2. ShiftRow:

   pergeseran baris-baris array state secara wrapping.

3. MixColumn:

   mengacak data di masing-masing kolom array state.

4. AddRoundKey:

   melakukan XOR antara state sekarang dengan round key.

3. Final round:
   proses untuk putaran terakhir:
   a. ByteSub.
   b. ShiftRow.
   c. AddRoundKey.

### 2.3 Istilah-istilah

dalam ilmu kriptografi terdapat beberapa istilah penting. Berikut adalah istilah-istilah yang sering digunakan dalam kriptografi:

1. **Plaintext** : pesan asli sebelum dienkripsikan ( data asli).
2. **Ciphertext :** pesan ter-enkrip (tersandi) yang merupakan hasil enkripsi.
3. **Enkripsi** : proses pengubahan *plaintext* menjadi *ciphertext*.
4. **Dekripsi** : kebalikan dari enkripsi yakni mengubah *ciphertext* menjadi *plaintext*, sehingga berupa data awal/asli.
5. **Kriptografer** : orang yang menggunakan algoritma kriptografi untuk merahasiakan pesan dan mendeskripsikannya kembali.
6. **Kriptanalisis** (*cryptanalysis*) : ilmu dan seni untuk memecahkan chipertext, berupa memperoleh plaintext dari chipertext tanpa mengetahui kuncinya. Pelakunya disebut **kriptanalis.**
7. **Kriptologi** (*cryptology*) : studi mengenai kriptografi dan kriptanalisis.

### 2.4 Short Message Service (SMS)

*Short Message Service* atau biasa dikenal dengan pesan teks singkat adalah suatu mekanisme pengiriman pesan teks singkat dari dan ke telepon seluler. Setiap pesan tidak dapat berisi lebih dari 160 karakter (Jika karakter non-latin digunakan, misal huruf cina atau arab, maka pesan hanya dapat berisi 70 karakter). Karakter tersebut dapat berupa teks atau binary Non Text.

### 2.5 Java 2 Micro Edition

Pada awal perilisannya java disebut dengan JDK (Java Development Kit). Dalam JDK, semua kebutuhan untuk pengembangan program dan eksekusi program masih tergabung jadi satu. Sekarang Sun microsystem telah merubah namanya menjadi JSDK

(Java Software Development Kit) dimana kebutuhan pengembangan program dipisahkan dengan kebutuhan eksekusi program.

J2ME merupakan bagian dari J2SE yang dirancang untuk dapat menjalankan program Java pada perangkat-perangkat kecil semacam handphone dan PDA, yang memiliki karakteristik yang berbeda dengan sebuah komputer biasa, misalnya kecilnya jumlah memori pada handphone dan PDA.

J2ME terdiri atas komponen-komponen sebagai berikut :

1. Java Virtual Machine (JVM)
   Komponen ini untuk menjalankan program program Java pada emulator atau handled devices.
2. Java API (Application Programming Interface)

Komponen ini merupakan kumpulan librari untuk menjalankan dan mengembangkan program Java pada handled devices.

## 3 ANALISIS DAN PERANCANGAN SYSTEM

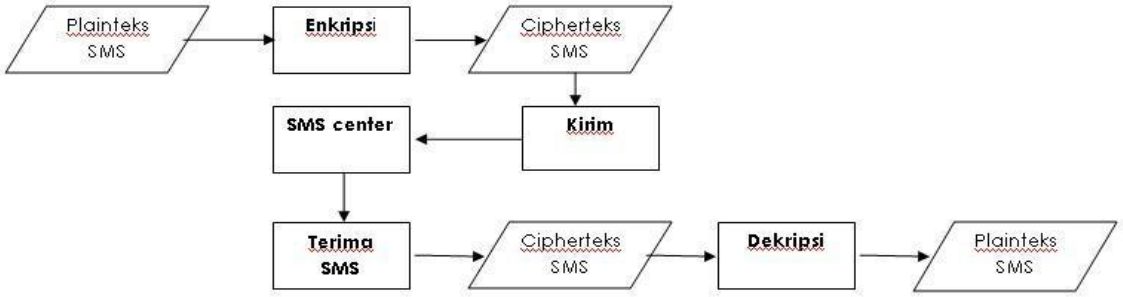
Pengembangan aplikasi enkripsi dekripsi SMS dilakukan secara bertahap dengan model proses perangkat lunak menggunakan metode *waterfall*. Tahap pertama adalah aliran kerja kebutuhan, selanjutnya diikuti dengan aliran kerja analisa, aliran kerja desain, dan terahir adalah uji coba yang akan dimasukkan ke dalam bab tersendiri.

### 3.1 Aliran Kerja Kebutuhan

Dalam aliran kerja kebutuhan dibagi menjadi dua tahapan yang berbeda. Tahapan pertama adalah menentukan area aplikasi yang akan dipakai jika aplikasi akan diimplementasikan, dan yang kedua adalah menentukan kebutuhan awal sebuah perangkat lunak.

#### 3.1.1 Area Aplikasi

Masih banyak masyarakat umum yang tidak tau cara untuk menjadikan informasi yang disampaikan melalui SMS agar tetap aman. Mereka menyadari betul bahwa keamanan jaringan pada pengiriman SMS masih lemah. Maka .perlu sebuah aplikasi yang mampu mengamankan pesan SMS terebut dengan memiliki fungsi enkripsi dan dekripsi. Berikut adalah gambar area aplikasi yang akan dikembangkan




**Gambar 3.1 skema area aplikasi**

### 3.1.2 Kebutuhan Awal

Dari penjelasan informasi yang terdapat pada Area aplikasi maka akan didapat informasi-informasi lain, salah satunya adanya kondisi yang akan terjadi dari seseorang yang akan mengirim maupun yang menerima pesan SMS yaitu:

1. Pada masing-masing ponsel pengirim dan penerima pesan SMS harus sudah terinstall aplikasi ini.

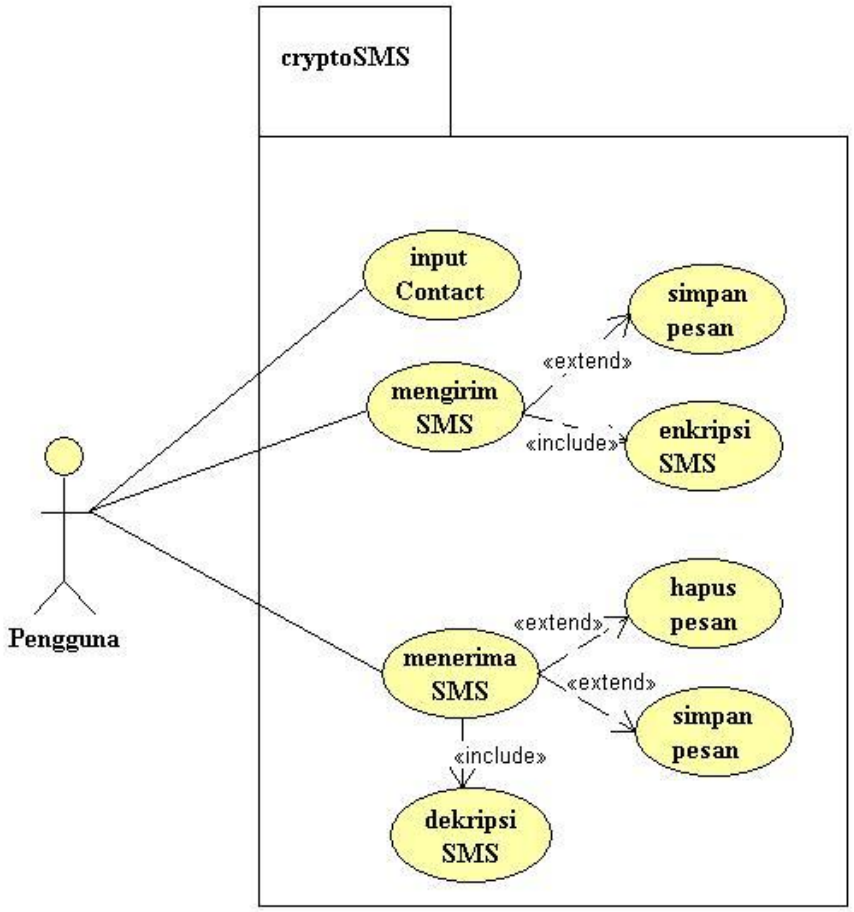
2. Pengirim dan penerima pesan SMS harus mempunyai kunci simetri yang sama, sesuai dengan kesepakatan pengirim dan penerima sebelum dapat melakukan pengiriman dan penerimaan pesan SMS terencripsi.

## 3.2 Aliran Kerja Analisa

Pada tahap aliran kerja analisa yang harus dilakukan adalah mengekstraksi kelas entity. Dalam melakukan hal ini dibagi menjadi 4 langkah, yaitu fungsional modeling, entity class modeling, interaction modeling dan dynamic modeling.

### 3.2.1 Fungsional Modeling

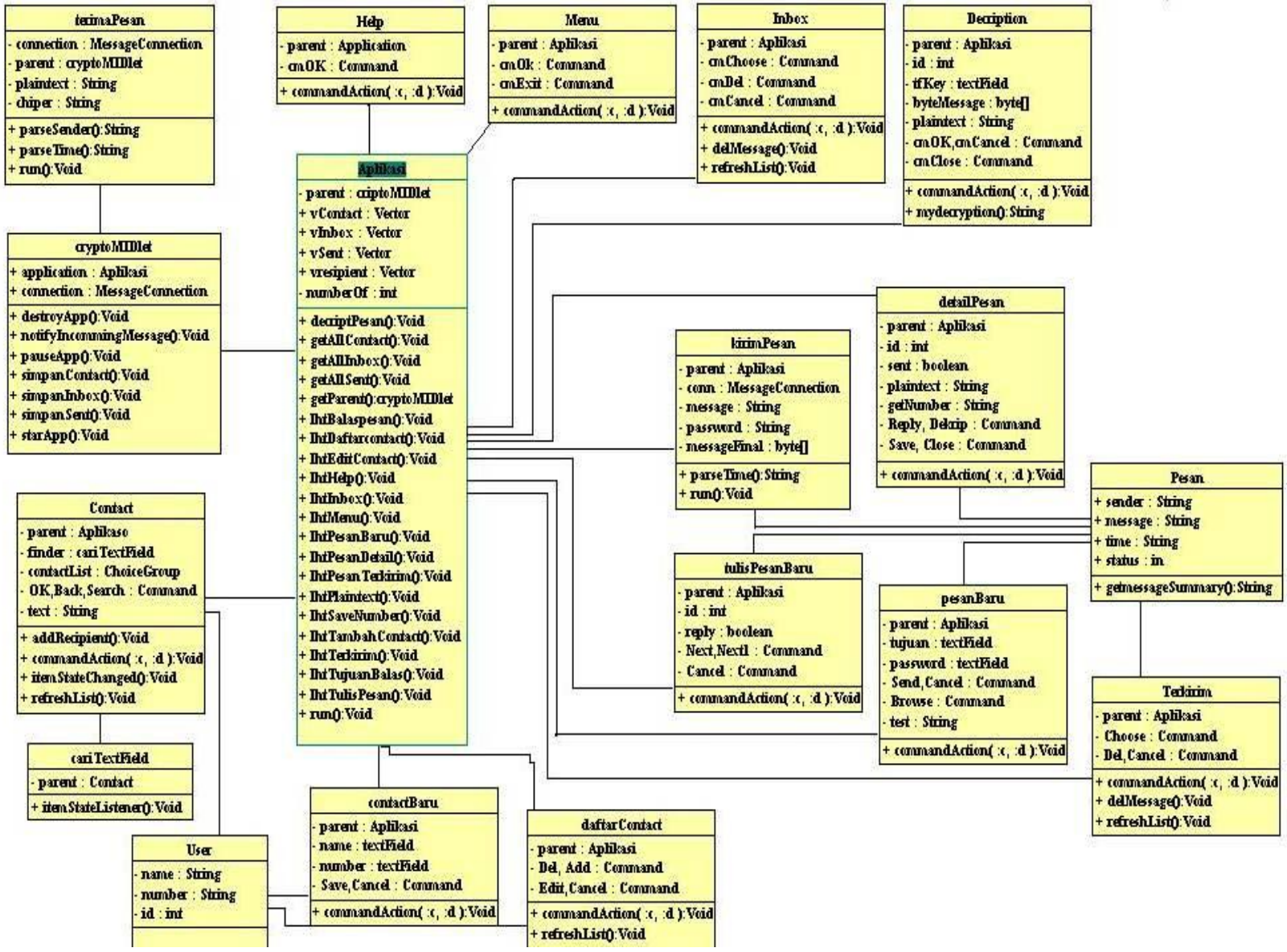
Di dalam fungsional modeling yang dilakukan adalah mendeskripsikan use case. Use case digunakan untuk melihat interaksi antara sistem dengan pengguna atau disebut aktor.




**Gambar 3.2 Use Case Diagram**

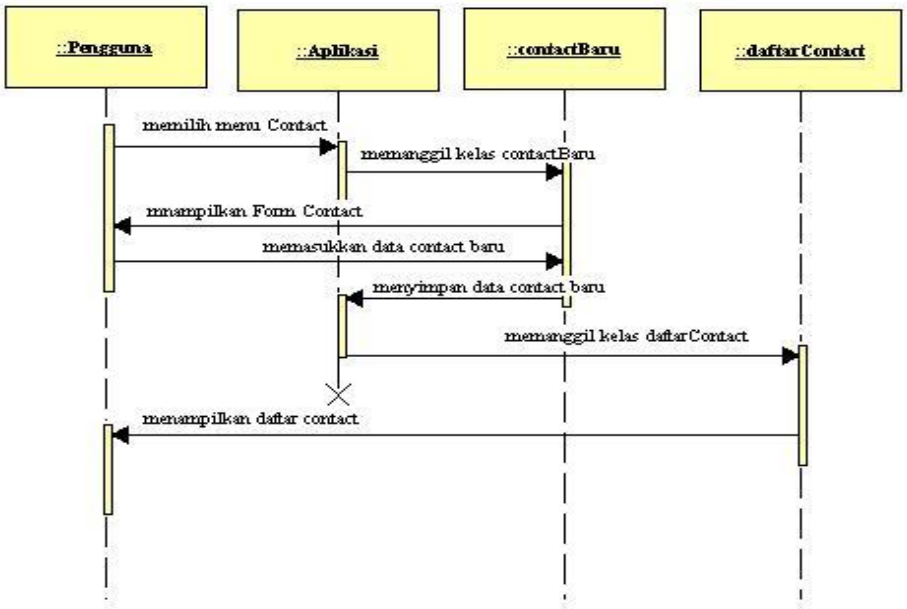
### 3.2.2 Entity Class Modeling

Pada tahap ini menentukan kelas entitas beserta atributnya, kemudian putuskan hubungan dan interaksi antara kelas entitas dan ditampilkan dalam bentuk diagram kelas seperti gambar 3.3


terimaPesan
- connection : MessageConnection
- parent : cryptoMIDlet
- plaintext : String
- chiper : String
+ parseSender():String
+ parseTime():String
+ run():Void
Help
- parent : Application
- cmOK : Command
+ commandAction( :c, :d ):Void
Menu
- parent : Aplikasi
- cmOk : Command
- cmExit : Command
+ commandAction( :c, :d ):Void
Inbox
- parent : Aplikasi
- cmChoose : Command
- cmDel : Command
- cmCancel : Command
+ commandAction( :c, :d ):Void
+ delMessage():Void
+ refreshList():Void
Decription
- parent : Aplikasi
- id : int
- tfKey : textField
- byteMessage : byte[]
- plaintext : String
- cmOK,cmCancel : Command
- cmClose : Command
+ commandAction( :c, :d ):Void
+ mydecryption():String
cryptoMIDlet
+ application : Aplikasi
+ connection : MessageConnection
+ destroyApp():Void
+ notifyIncommingMessage():Void
+ pauseApp():Void
+ simpanContact():Void
+ simpanInbox():Void
+ simpanSent():Void
+ starApp():Void
Aplikasi
- parent : criptoMIDlet
+ vContact : Vector
+ vInbox : Vector
+ vSent : Vector
+ vrecipient : Vector
- numberOf : int
+ decriptPesan():Void
+ getAllContact():Void
+ getAllInbox():Void
+ getAllSent():Void
+ getParent():cryptoMIDlet
+ IhtBalaspesan():Void
+ IhtDaftarcontact():Void
+ IhtEditContact():Void
+ IhtHelp():Void
+ IhtInbox():Void
+ IhtMenu():Void
+ IhtPesanBaru():Void
+ IhtPesanDetail():Void
+ IhtPesanTerkirim():Void
+ IhtPlaintext():Void
+ IhtSaveNumber():Void
+ IhtTambahContact():Void
+ IhtTerkirim():Void
+ IhtTujuanBalas():Void
+ IhtTulisPesan():Void
+ run():Void
kirimPesan
- parent : Aplikasi
- conn : MessageConnection
- message : String
- password : String
- messageFinal : byte[]
+ parseTime():String
+ run():Void
detailPesan
- parent : Aplikasi
- id : int
- sent : boolean
- plaintext : String
- getNumber : String
- Reply, Dekrip : Command
- Save, Close : Command
+ commandAction( :c, :d ):Void
Pesan
+ sender : String
+ message : String
+ time : String
+ status : in
+ getmessageSummary():String
Contact
- parent : Aplikaso
- finder : cariTextField
- contactList : ChoiceGroup
- OK,Back,Search : Command
- text : String
+ addRecipient():Void
+ commandAction( :c, :d ):Void
+ itemStateChanged():Void
+ refreshList():Void
tulisPesanBaru
- parent : Aplikasi
- id : int
- reply : boolean
- Next,Nextl : Command
- Cancel : Command
+ commandAction( :c, :d ):Void
pesanBaru
- parent : Aplikasi
- tujuan : textField
- password : textFidld
- Send,Cancel : Command
- Browse : Command
- test : String
+ commandAction( :c, :d ):Void
Terkirim
- parent : Aplikasi
- Choose : Command
- Del,Cancel : Command
+ commandAction( :c, :d ):Void
+ delMessage():Void
+ refreshList():Void
cariTextField
- parent : Contact
+ itemStateListener():Void
contactBaru
- parent : Aplikasi
- name : textField
- number : textField
- Save,Cancel : Command
+ commandAction( :c, :d ):Void
daftarContact
- parent : Aplikasi
- Del, Add : Command
- Edit,Cancel : Command
+ commandAction( :c, :d ):Void
+ refreshList():Void
User
- name : String
- number : String
- id : int

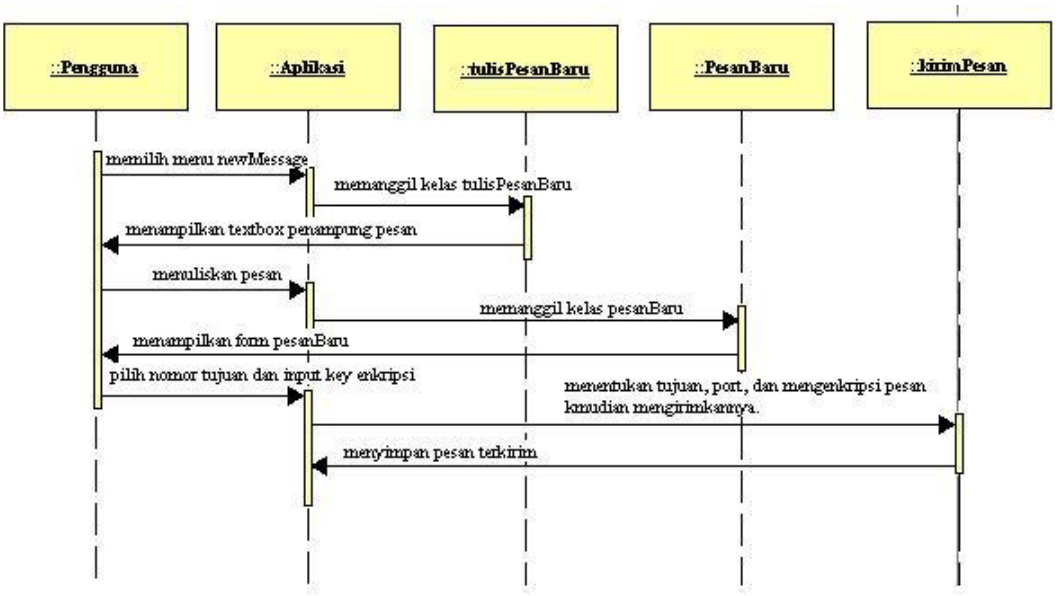
### 3.2.3 Interaction Modeling

Pada tahab ini menentukan interaksi antara beberapa objek menggunakan sequence diagram. Sequence diagram merepresentasikan interaksi antar objek di dalam dan di sekitar sistem, termasuk pengguna dan antarmuka pengguna. Sequence diagram terdiri atas dimensi vertikal yang merepresentasikan waktu, dan dimensi horizontal yang merepresentasikan objek-objek terkait. Aspek penting dari sequence diagram adalah keterurutan waktu, yang mengindikasikan bahwa interaksi direpresentasikan tahab demi tahab. Sequence diagram di sini dibangun berdasarkan diagran use case pada gambar 3.2.

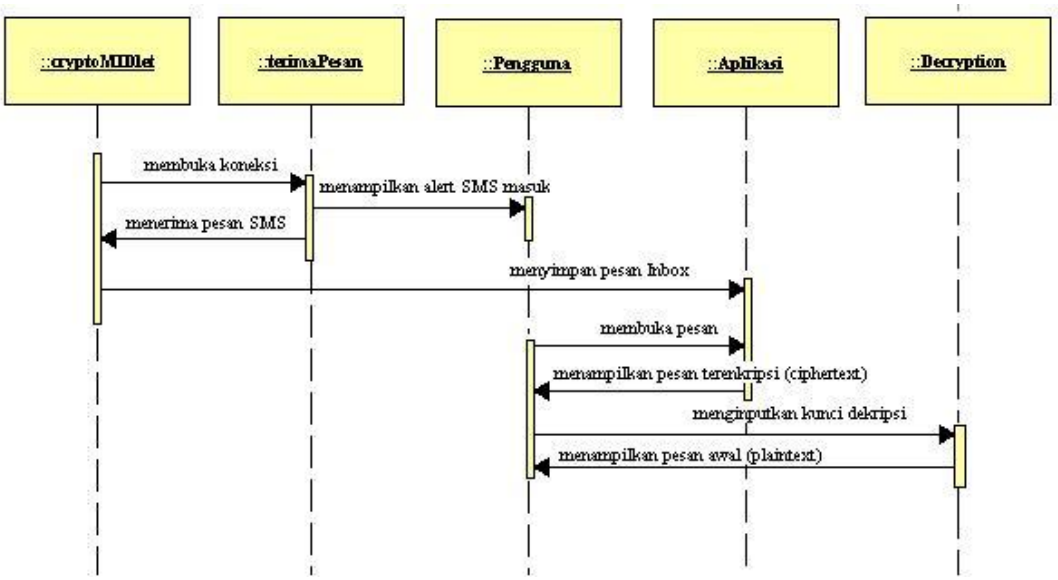



**Gambar 3.4 sequence diagram use case input Contact**

Untuk dua use case lainnya yaitu use case mengirim SMS dam menerima SMS diperlihatkan pada gambar 3.5 dan gambar 3.6.



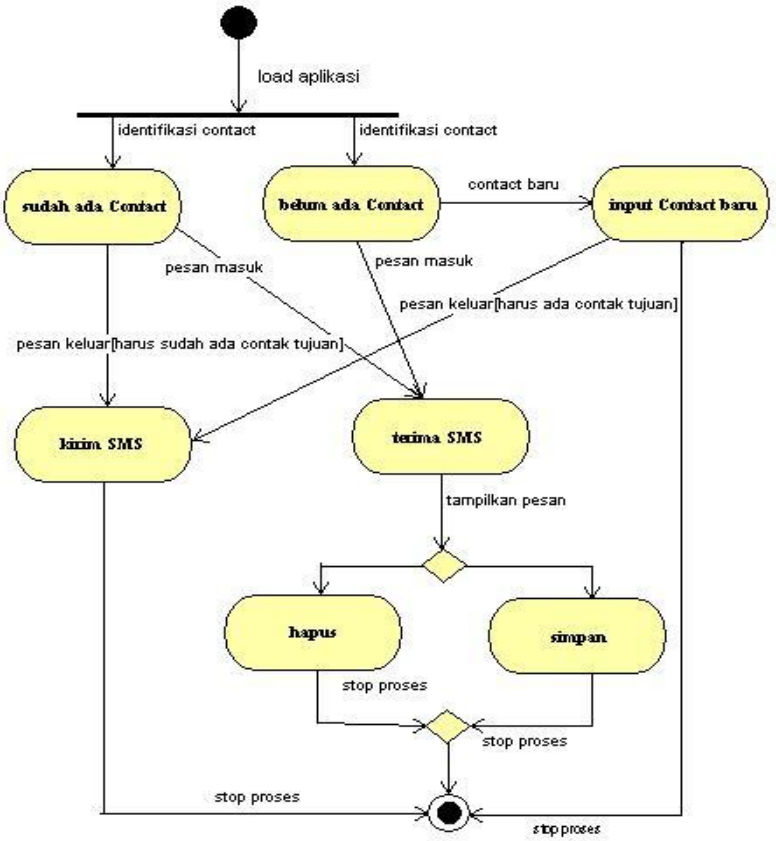

**Gambar 3.5 sequence diagram use case mengirim SMS**




**Gambar 3.6 sequence diagram use case menerima SMS**

### 3.2.4 Dinamic Modeling

Gambar 3.7 merupakan statechart diagram yang merefleksikan semua operasi yang menuju atau ada di sistem dan juga mengindikasikan semua kejadian yang menyebabkan transisi dari satu state ke state lainnya.




**Gambar 3.7 Statechart Diagram**

State atau keadaan dimulai ketika pengguna membuka aplikasi. Bagi pengguna aplikasi yang hendak mengirim pesan harus memastikan terlebih dahulu apakah nomor tujuan pengiriman pesan telah tersimpan di dalam record store. Jika belum, pengguna diharuskan menginputkan nama beserta nomor tujuan pada form new contact yang nantinya akan disimpan dan ditampilkan pada contact list. Namun jika nomor tujuan sudah ada, pengguna dapat langsung memakai fitur SMS. Fitur SMS yang berada pada aplikasi ini adalah mengirim dan menerima pesan SMS. Dalam pengiriman SMS, hal yang pertama kali harus dilakukan adalah menulis pesan yang akan dikirimkan, memilih nomor tujuan yang terdapat pada daftar contact, dan kemudian menuliskan kunci enkripsi, selanjutnya mengirim pesan tersebut. Sedangkan untuk menerima SMS, nomor pengirim tidak harus tersimpan dahulu, pengguna dapat membuka pesan SMS yang diterima cukup dengan menuliskan kunci dekripsi yang sesuai, kemudian dapat memilih apakah pesan SMS tersebut akan disimpan atau dihapus. Jika pengguna telah selesai menggunakan aplikasi ini, pengguna dapat langsung mematikan aplikasi ini.

# 4 IMPLEMENTASI DAN PEMBAHASAN

Setelah melakukan tahapan analisa dan perancangan sistem, maka tahapan selanjutnya adalah implementasi. Tahapan ini merupakan proses menerjemahkan rancangan yang telah didesain pada bahasa pemrograman, sehingga pada tahap ini sistem sudah siap untuk dioperasikan sesuai dengan fungsi dan tujuan dari pembuatan perangkat lunak tersebut.

## 4.1 Implementasi Antar Muka

Dalam aplikasi ini menggunakan media telepon seluler sehingga tampilan akan berbeda antara satu merk dengan yang lainnya. Disini penulis menggunakan emulator wtk 2.5.2 untuk menjalankan program.

### 4.1.1 Antar Muka Menu Utama

Menu utama merupakan tampilan list yang memuat beberapa elemen yaitu new message, inbox, sent, cobtact dan help. Penambahan beberapa ikon pada list ini dimaksudkan untuk menjadikan menu utama dapat tampil lebih menarik. Tampilan menu dapat dilihat pada gambar 4.1.

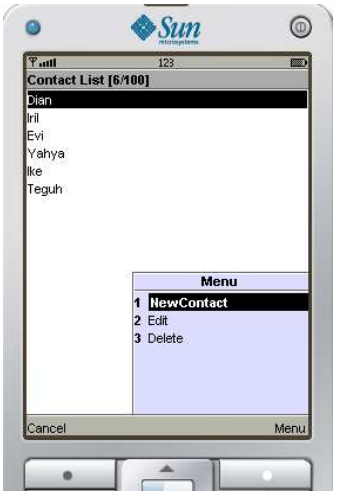

**Gambar 4.1 tampilan menu utama**

## 4.2 Kelas cryptoMidlet

Dalam pembuatan aplikasi MIDP harus ada sebuah file yang merepresentasikan kelas MIDlet. Pada aplikasi yang dibangun oleh punulis dinamakan cryptoMidlet.java. file ini merupakan file yang akan diproses ketika MIDlet dijalankan. Di kelas inilah fungsi-fungsi starApp(), pauseApp() dan destroyApp() dapat diimplementasikan.

### 4.3 Pengaturan Contact

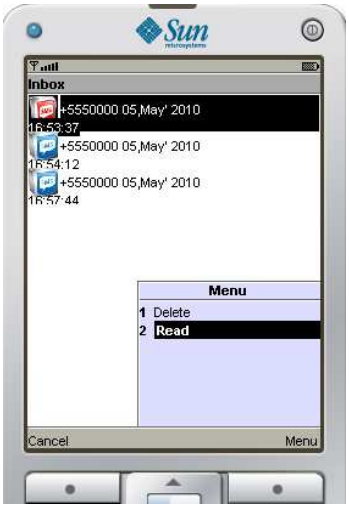
Kelas daftarContact berfungsi untuk menampilkan daftar contact . pertama ia akan menggunakan vector vContact dari kelas Aplikasi di mana vector tersebut dideklarasikan




**Gambar 4.2 tampilan daftarContact**

### 4.4 Pengaturan Pesan

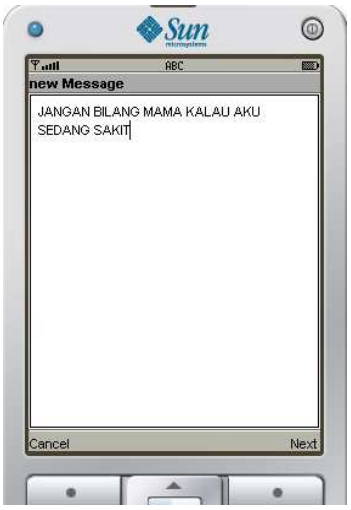
Pengaturan pesan diklasifikasikan menjadi dua bagian yaitu pesan masuk yang disimpan pada record store rdInbox dan pesan keluar yang disimpan dalam record store rdSent. Guna mempercepat dalam pengaksesan memori digunakan juga vector vInbox dan vSent..




**Gambar 4.3 tampilan Inbox**

Untuk pengaturan pesan keluar, sama seperti pengaturan pesan masuk vector vSent dibangun guna menampung isi dari record store rdSent. Selanjutnya dengan method refreshList() pesan keluar akan ditampilkan dalam bentuk list. Method untuk menampilkannya didefinisikan pada file terkirim.


**Gambar 4.4 tampilan Sent atau pesan keluar**




**Gambar 4.5 tampilan Textbox pesan baru**

Pada kelas ini ditambahkan dua buah command yaitu cmNext dan cmCancel. Penambahan command tersebut dimaksudkan ketika pengguna telah selesai menuliskan pesan, pengguna dapat meneruskan pengiriman pesan dengan memilih menu Next untuk memasuki tahapan selanjutnya yaitu menentukan tujuan dan menginputkan kunci enkripsi.


**Gambar 4.6 tampilan Form pesan baru**

### 4.5 Penerimaan SMS

Proses penerimaan SMS didefinisikan pada kelas terimaPesan. Kelas ini akan membuka koneksi untuk menerima pesan. Pada kelas ini terdapat beberapa method untuk keperluan menerima pesan. Pertama method yang digunakan untuk memparshing alamat dan mengembalikan nilai berupa nama pengirim.

### 4.6 Dekripsi SMS

Setelah pesan masuk disimpan pada vektor vInbox, pengguna dapat membaca isi pesan tentu saja dalam bentuk terenkripsi. Untuk mendekripsikan pesan pengguna dapat memilih command cmDekrip yang menampilkan textfiel kunci dekripsi. Agar pesan dapat kembali seperti pesan awal (plaintext) pengguna harus menginputkan kunci yang sama yang digunakan oleh pengirim.


**Gambar 4.7 tampilan Form *decryption key***


**Gambar 4.8 tampilan pesan hasil dekripsi**

## 5 PENUTUP

Bagian ini merupakan ahir dari penyusunan skripsi, yang menjelaskan kesimpulan-kesimpulan yang diperoleh selama pengerjaannya

### 5.1 Kesimpulan

Berikut adalah kesimpulan yang diperoleh selama pelaksanaan skripsi ini:

1. Sebuah aplikasi enkripsi SMS yang mengimplementasikan algorima kriptografi simetri telah berhasil dibangun.
2. Aplikasi enkripsi SMS telah berhasil meningkatkan keamanan pengiriman pesan SMS melalui telepon seluler.
3. Algoritma AES dapat diimplementasikan dengan baik untuk melakukan enkripsi SMS yang bekerja pada jaringan GSM dengan mengirimkan pesan dalam bentuk binary.

### 5.2 Saran

Untuk perbaikan dan pengembangan aplikasi enkripsi SMS lebih lanjut, disarankan perbaikan sebagai berikut:

1. Pengembangan aplikasi enkripsi SMS ini menggunakan satu jenis algoritma yaitu AES, guna meningkatkan keamanan yang lebih, aplikasi dapat ditambahkan beberapa algoritma kriptografi lainnya.
2. Aplikasi ini dapat mengengkripsikan pesan karakter standar ASCII, untuk pengembangan lebih lanjut diharapkan aplikasi dapat mengenkripsikan karakter Arabic dan Chines.

**Daftar Pustaka**


Ariyus,Dony. 2008. *Pengantar Ilmu Kriptografi Teori, Analisis, dan Implementasi,* Yogyakarta, Andi Offset.

Buchmann, J. A., 2000 *Introduction to Cryptography,* first editon Springer Verleg New York Inc, New York.

Dharwianti, S., Wahono, R.S., 2006 *Pengantar Unified Modeling Languange (UML)*, <http:// www.ilmukomputer.org/wp.content/upload/2006/08/yanti_uml.zip> diakses pada 30 Desember 2009

Glosarry Wireless Terms., <http://www.braddey.com/glosarry.html>. Diakses 8 Desember 2009.

Gupta, P., Short Message Sevice:What, How and Where., <http://www.wirelessdevnet.com/channels/sms/features/sms.html>. Diakses 9 Desember 2009.

JENI,*Pengenalan Bahasa Java*, <http://poss.ipb.ac.id/file/JENI-intro1-bab2-pengenalan_bahasa_JAVA.pdf>. Diakses pada 19 November 2009

Jhones, N., 2008, *Don't Use SMS for Confidential Communication*, <http://www.garter.com/displaydocument?doc_cd=111720>. Diakses pada 19 Desember 2009.

Kenudsen,J.,2003, *Wireless Java Developing With J2me*, Second Edition, Appress, Barkeley.