ENKRIPSI *SHORT MESSAGE SERVICE* (SMS) DENGAN MENGGUNAKAN METODE RSA PADA *SMARTPHONE*

SKRIPSI

Oleh:

RUDY SETIAWAN
NIM. 07650060

JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
2013

ENKRIPSI *SHORT MESSAGE SERVICE* (SMS) DENGAN MENGGUNAKAN METODE RSA PADA *SMARTPHONE*

SKRIPSI

Diajukan Kepada:
Fakultas Sains dan Teknologi
Universitas Islam Negeri
Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S. Kom)

Oleh:
RUDY SETIAWAN
NIM. 07650060

JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
2013

ENKRIPSI *SHORT MESSAGE SERVICE* (SMS) DENGAN MENGGUNAKAN METODE RSA PADA *SMARTPHONE*

SKRIPSI

Oleh:

RUDY SETIAWAN
NIM. 07650060

Telah Diperiksa dan Disetujui untuk Diuji
Tanggal : 1 Juli 2013

Dosen Pembimbing I	Dosen Pembimbing II
Zainal Abidin, M.Kom	**A'la Syauqi, M.Kom**
NIP. 19760613 200501 1 004	**NIP. 19771201 200801 1 007**

Mengetahui
Ketua Jurusan Teknik Informatika

Ririen Kusumawati, M.Kom
NIP. 19720309 200501 2 002

ENKRIPSI *SHORT MESSAGE SERVICE* (SMS) DENGAN MENGGUNAKAN METODE RSA PADA *SMARTPHONE*

SKRIPSI

Oleh:
RUDY SETIAWAN
NIM. 07650060

Telah Dipertahankan di Depan Dewan Penguji Skripsi dan
Dinyatakan Diterima Sebagai Salah Satu Persyaratan untuk
Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal : 19 Juli 2013

Susunan Dewan Penguji			Tanda Tangan
1. Penguji Utama	:	Fachrul Kurniawan, M.MT NIP. 19771020 200901 1 001	( )
2. Ketua	:	Fresy Nugroho, M.T NIP. 19710722 201101 1 001	( )
3. Sekretaris	:	Zainal Abidin, M.Kom NIP. 19760613 200501 1 004	( )
4. Anggota	:	A'la Syauqi, M.Kom NIP. 19771201 200801 1 007	( )

Mengetahui dan Mengesahkan
Ketua Jurusan Teknik Informatika

Ririen Kusumawati, M.Kom
NIP. 19720309 200501 2 002

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Rudy Setiawan

NIM : 07650060

Fakultas/Jurusan : Sains dan Teknologi / Teknik Informatika

Judul Penelitian : **Enkripsi *Short Message Service* (SMS) Dengan Menggunakan Metode RSA pada *Smartphone***

Menyatakan dengan sebenarnya bahwa hasil penelitian saya ini tidak terdapat unsur-unsur penjiplakan karya penelitian atau karya ilmiah yang pernah dilakukan atau dibuat orang lain, kecuali yang secara tertulis dikutip dalam naskah ini atau disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata hasil penelitian ini terbukti terdapat unsur-unsur jiplakan maka saya bersedia untuk mempertanggung jawabkan, serta diproses sesuai peraturan yang berlaku.

Malang, 19 Juli 2013
Yang Membuat Pernyataan,

Rudy Setiawan
NIM. 07650060

MOTTO

Life doesn't have to be perfect.
It just has to be lived

~Dexter Morgan~

PERSEMBAHAN

Buat Ibu Bapak dan Adekq.

Terimakasih atas doa dan semangat yang tanpa henti tertuju padaq.

Setiap saat setiap waktu setiap detik selalu mengingatkan untuk segera menyelesaikan skripsi ini.. dan akhirnya.. alhamdulillah.....

~done~

KATA PENGANTAR

Assalamu'alaikum Wr.Wb.

Syukur Alhamdulillah penulis haturkan kehadirat Allah SWT yang telah melimpahkan rahmat, taufik serta hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi ini sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Ucapan terima kasih penulis sampaikan kepada:

1. Prof. Dr. H. Mudjia Rahardjo MSc, selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. Drh. Bayyinatul Muchtaromah, M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Ibu Ririen Kusumawati, M.Kom selaku Ketua Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Bapak Zainal Abidin, M.Kom selaku pembimbing dalam skripsi ini yang telah memberikan bimbingan dan pengarahan dalam proses penyelesaian skripsi ini.
5. Bapak A'la Syauqi, M.Kom selaku pembimbing integrasi sains dan Islam yang telah memberikan bimbingan dan pengarahan integrasi dalam skripsi ini.
6. Seluruh Dosen Universitas Islam Negeri Maulana Malik Ibrahim Malang, khususnya dosen Teknik Informatika beserta seluruh staf yang telah memberikan ilmu dan membantu dalam penyelesaian skripsi ini.

7. Bapak, Ibu, adik, dan seluruh keluarga besar di Malang dan Pasuruan yang selalu memberikan do'a dan motivasi dalam penyelesaian skripsi ini.
8. Seluruh teman-teman Jurusan Teknik Informatika khususnya angkatan 2007.
9. Sahabat-sahabat penulis yang telah memotivasi dan membantu dalam proses penyelesaian skripsi ini.
10. Dan kepada seluruh pihak yang membantu penulisan skripsi ini, yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat kekurangan. Penulis berharap semoga skripsi ini dapat memberikan manfaat kepada pembaca dan khususnya bermanfaat bagi penulis secara pribadi.

Wassalamu'alaikum Wr. Wb.

Malang, 19 Juli 2013
Penulis,

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

ABSTRAK

Setiawan, Rudy. 2013. **Enkripsi *Short Message Service* (SMS) Dengan Menggunakan Metode RSA pada *Smartphone***. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (1) Zainal Abidin, M.Kom. (2) A'la Syauqi, M.Kom.

Short Message Service (SMS) merupakan salah satu fasilitas yang ada pada telepon seluler yang berfungsi sebagai media komunikasi antar manusia. Banyak orang menggunakan layanan SMS untuk mengirimkan informasi penting seperti data hasil transaksi atau data pribadi. Dalam mekanisme pengiriman SMS diketahui masih ada cela yang memungkinkan bocornya informasi yang dikirim melalui SMS seperti yang terjadi pada kasus Nourse pada tahun 2002. Maka diperlukan suatu tindakan untuk menghindari kebocoran informasi itu terjadi lagi.
Salah satu cara untuk mengantisipasi masalah tersebut adalah dengan mengenkripsi pesan yang akan dikirim melalui layanan SMS. Prinsip dasarnya adalah menyembunyikan informasi sedemikian rupa agar orang yang berhak saja yang dapat mengetahui isi dari informasi yang tersembunyi tersebut. Algoritma yang digunakan untuk mengenkripsi adalah algoritma RSA. RSA adalah algoritma kriptografi asimetris yang sering digunakan karena tingkat keamanan yang sulit dibobol. Karena dalam proses pembuatan kunci publik dan kunci pribadi terdapat beberapa faktor yang jadi pertimbangan, yaitu ukuran kunci, penentuan dua nilai untuk pemfaktoran, sehingga memiliki tingkat kesulitan yang tinggi untuk memecahkan kunci tersebut.
Aplikasi enkripsi SMS ini berjalan pada *smartphone* yang berbasis android. Pengirim dan penerima pesan harus sama-sama menggunakan aplikasi ini. Dalam ujicoba yang telah dilakukan, aplikasi ini bisa menjalankan semua fiturnya berdasarkan fungsinya masing-masing. Namun terdapat beberapa keterbatasan yang dimiliki oleh aplikasi ini, diantaranya yaitu aplikasi enkripsi SMS ini memiliki kunci publik yang cukup pendek untuk menghemat biaya pengiriman SMS. Kemudian dalam mendekripsikan pesan, aplikasi ini tidak bisa mendekripsikan lebih dari 50 karakter. Dan tidak bisa menampilkan fiturnya secara rapi pada *device* yang memiliki dimensi lebih dari 4.0.

Kata Kunci: SMS, Kriptografi, *chipertext,* RSA, Android.

ABSTRACT

Setiawan, Rudy. 2013. **Encryption Short Message Service (SMS) Using RSA Algorithm on Smartphone**. Thesis. Informatics Engineering Faculty of Science and Technology the State of Islamic University Maulana Malik Ibrahim Malang. Supervisor: (1) Zainal Abidin, M.Kom. (2) A'la Syauqi, M.Kom.

Short Message Service (SMS) is one of the facilities that exist in the mobile phone serves as a medium of communication between people. Many people use the SMS service to send important information such as data from transactions or personal data. In the SMS delivery mechanism known to still exist defect that allows leaking of information sent via SMS as happened in the case of Nourse in 2002. It would require an act to avoid information leaks that happen again.

One way to anticipate such problems is to encrypt messages to be sent via the SMS service. The basic principle is to hide information in such a way that only people who are entitled to know the contents of the hidden information. Algorithm is used to encrypt the RSA algorithm. RSA is an asymmetric cryptographic algorithm that is often used because of the level of security that is difficult burglarized. Because in the process of making public key and private key are several factors be considered, namely the size of the key, the determination of the two values for factoring, so it has a high degree of difficulty to solve the key.

This SMS encryption application runs on Android based smartphones. Sender and the recipient must both use this application. In the trials that have been done, this application can run all its functions based on respectively. But there are some limitations that are owned by this application, including the application of SMS encryption has a public key that is short enough to save the cost of sending SMS. Later in decrypting the message, the application can not decrypt more than 50 characters. And can not be neatly in its display device has a dimension of 4.0.

Keywords: SMS, cryptography, ciphertext, RSA, Android.

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Komunikasi merupakan cara manusia untuk berinteraksi satu sama lain. Dengan melakukan komunikasi antar sesama, manusia dapat mengekspresikan dirinya dan mengembangkan kepribadiannya. Melalui komunikasi, seseorang dapat menyampaikan apa yang ada dipikirannya maupun perasaan yang ada di hatinya kepada orang lain baik secara langsung maupun tidak langsung. Pada zaman dahulu untuk melakukan komunikasi antar sesama manusia harus dilakukan dengan cara bertemu langsung dengan yang bersangkutan. Namun, seiring dengan perkembangan teknologi, seseorang tidak harus bertemu langsung untuk menyampaikan pesannya. Dengan menggunakan telepon, seseorang bisa dengan mudah berbicara dengan orang lain atau saling bertukar pesan walaupun tidak pada satu tempat.

Short Message Service (SMS) merupakan salah satu fasilitas yang ada pada telepon seluler yang juga berfungsi sebagai media komunikasi, dimana seseorang bisa mengirimkan pesan langsung ke sesama pengguna telepon seluler hanya dengan menulis dan mengirimkannya ke nomor seluler yang dituju. Banyak orang menggunakan layanan SMS untuk mengirimkan informasi penting seperti data hasil transaksi atau data pribadi yang bersifat rahasia. Dalam dunia bisnis SMS sering digunakan untuk mengaktifkan layanan peringatan dan pemberitahuan, seperti pengiriman informasi dari perusahaan kartu kredit untuk traksaksi yang beresiko tinggi, pemberitahuan kepada administrator sistem pada suatu perusahaan jika terjadi

peristiwa penting dalam sistem IT, atau pengiriman password ke nasabah bank untuk mengkonfirmasi transaksi *on-line* dengan resiko tinggi. (The Government of the Hong Kong Special Administrative Region, 2008)

Namun kebanyakan orang tidak menyadari bahwa bertukar informasi melaui SMS tidak menjamin kerahasian dari isi informasi tersebut, karena layanan SMS diketahui masih memiliki beberapa kelemahan. Secara umum SMS yang dikirim akan melewati *Short Message Service Center* (SMSC), tempat dimana SMS disimpan sebelum dikirim ke tujuan. Sehingga apabila terjadi penyerangan pada SMSC, maka pesan yang terkirim dapat terbaca dan dimanfaatkan oleh pihak yang tidak berkepentingan. (Hulisani, Johnny dan Judith, 2004)

Sebuah contoh kasus pada tanggal 19 November 2002 Philip Nourse, seorang mahasiswa di Inggris dijatuhi hukuman lima bulan penjara karena telah mendapatkan salinan SMS mantan pacarnya secara ilegal. Kemudian ia memposting informasi yang sangat pribadi dari mantan pacarnya di situs jejaring sosial "Friends Reunited". Nourse mendapatkan salinan SMS tersebut dari karyawan operator seluler di mm02, salah satu operator jaringan seluler di Inggris. Dalam kasus Nourse memang tidak terlalu besar, namun itu menunjukkan betapa mudahnya data SMS diperoleh untuk digunakan dalam hal kejahatan. (Nick Jones, 2002)

Untuk mencegah terjadinya kebocoran informasi dari SMS yang dikirim, peneliti membuat aplikasi untuk enkripsi SMS. Apabila terjadi serangan pada SMSC seperti contoh kasus Nourse, SMS yang ada pada *server* SMSC adalah SMS yang sudah terenkripsi. Karena dalam Al-Quran surat An Nisaa' ayat 58 dijelaskan bahwa:

إِنَّ ٱللَّهَ يَأۡمُرُكُمۡ أَن تُؤَدُّواْ ٱلۡأَمَٰنَٰتِ إِلَىٰٓ أَهۡلِهَا وَإِذَا حَكَمۡتُم بَيۡنَ ٱلنَّاسِ أَن تَحۡكُمُواْ بِٱلۡعَدۡلِۚ إِنَّ ٱللَّهَ نِعِمَّا يَعِظُكُم بِهِۦٓۗ إِنَّ ٱللَّهَ كَانَ سَمِيعَۢا بَصِيرًا .

"Sesungguhnya Allah menyuruh kamu menyampaikan amanat kepada yang berhak menerimanya, dan (menyuruh kamu) apabila menetapkan hukum di antara manusia supaya kamu menetapkan dengan adil. Sesungguhnya Allah memberi pengajaran yang sebaik-baiknya kepadamu. Sesungguhnya Allah adalah Maha mendengar lagi Maha melihat."(QS. An Nisaa':58)

Pada ayat diatas menjelaskan bahwa Allah SWT menyuruh umat manusia agar menyampaikan amanat yang di berikan kepadanya kepada yang berhak menerimanya. Hal ini berhubungan dengan tujuan dari tukar menukar pesan melalui media SMS. Dimana orang yang menggunakan layanan tersebut memiliki hak untuk menerima pesan sesuai dengan apa yang telah dikirimkan kepadanya. Apabila seseorang tidak menerima pesan sesuai dengan apa yang dikirimkan, maka hal tersebut akan menjauhkan dari sifat amanat yang diberikan oleh Allah SWT.

Menurut tafsir Jalalain maksud dari ayat diatas adalah (Sesungguhnya Allah menyuruh kamu untuk menyampaikan amanat) artinya kewajiban-kewajiban yang dipercayakan dari seseorang (kepada yang berhak menerimanya) ayat ini turun ketika Ali r.a. hendak mengambil kunci Kakbah dari Usman bin Thalhah Al-Hajabi penjaganya secara paksa yakni ketika Nabi SAW. datang ke Mekah pada tahun pembebasan. Usman ketika itu tidak mau memberikannya lalu katanya, "Seandainya saya tahu bahwa ia Rasulullah tentulah saya tidak akan menghalanginya." Maka Rasulullah saw. pun menyuruh mengembalikan kunci itu padanya seraya bersabda, "Terimalah ini untuk selama-lamanya tiada putus-putusnya!" Usman merasa heran

atas hal itu lalu dibacakannya ayat tersebut sehingga Usman pun masuk Islamlah. Ketika akan meninggal kunci itu diserahkan kepada saudaranya Syaibah lalu tinggal pada anaknya. Ayat ini walaupun datang dengan sebab khusus tetapi umumnya berlaku disebabkan persamaan di antaranya (dan apabila kamu mengadili di antara manusia) maka Allah menitahkanmu (agar menetapkan hukum dengan adil. Sesungguhnya Allah amat baik sekali) pada ni`immaa diidgamkan mim kepada ma, yakni nakirah maushufah artinya ni`ma syaian atau sesuatu yang amat baik (nasihat yang diberikan-Nya kepadamu) yakni menyampaikan amanat dan menjatuhkan putusan secara adil. (Sesungguhnya Allah Maha Mendengar) akan semua perkataan (lagi Maha Melihat) segala perbuatan. (Jalaluddin Asy-Syuyuthi dan Jalaluddin Muhammad Ibn Ahmad Al-Mahalliy, 2010)

Pada penelitian ini algoritma yang akan digunakan untuk mengekripsi adalah algoritma kriptografi asimetris RSA (Ron Rivest, Adi Shamir, dan Leonard Adleman). RSA digunakan karena dari segi teknis perhitungan, sistem RSA mempunyai cara enkripsi yang cukup mudah, namun apabila data sudah terenkripsi akan sulit untuk dibobol jika hanya mempunyai kunci publiknya saja. Karena dalam proses pembuatan kunci publik dan kunci pribadi terdapat beberapa faktor yang jadi pertimbangan, yaitu ukuran kunci, penentuan dua nilai untuk pemfaktoran, sehingga sulit untuk dibobol. Melalui penelitian ini diharapkan RSA dapat diimplementasikan pada enkripsi dan dekripsi pada aplikasi SMS.

1.2 Rumusan Masalah

Berdasarkan uraian di atas maka rumusan masalahnya adalah bagaimana mengimplementasikan teknologi enkripsi dan dekripsi sms pada telepon seluler yang berbasis android dengan menggunakan algoritma RSA?

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah untuk mengimplementasikan algoritma RSA dalam mengenkripsi SMS pada *Smartphone*.

1.4 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah dapat memberikan keamanan data atau informasi yang dikirim melalui *SMS* yang sudah terenkripsi.

1.5 Batasan Masalah

Batasan masalah dalam penelitian ini adalah:

1. Aplikasi berjalan pada *Smartphone* dengan *Operating System Android*.
2. Pengirim dan penerima pesan harus sama-sama menggunakan aplikasi ini.

1.6 Metode Penelitian

Untuk mencapai tujuan yang telah dirumuskan sebelumnya, maka metodologi penelitian yang digunakan dalam penulisan skripsi ini adalah:

1. Pembangkitan kunci publik dan kunci pribadi.

2. Menyimpan kunci publik dan kunci pribadi yang sudah dihasilkan dalam format (.txt).
3. Kunci publik yang sudah tersimpan kemudian dikirim kepada pengguna lain yang ingin bertukar pesan.
4. Untuk mengirimkan pesan yang di enkripsi harus mengetahui kunci publik dari penerima.
5. Mengekripsi pesan yang akan dikirim dengan mengambil kunci publik yang sudah tersimpan dalam kontak telepon.
6. Pesan yang sudah di enkripsi kemudian dikirim ke nomor tujuan.

1.7 Sistematika Penulisan

Untuk memberikan gambaran dan kerangka yang jelas mengenai pokok bahasan dalam setiap bab dalam penelitian ini maka diperlukan sistematika pembahasan. Berikut gambaran sistematika pembahasan pada masing-masing bab:

Bab I Pendahuluan, SMS merupakan fasilitas yang ada pada telepon seluler yang banyak digunakan oleh umat manusia sampai saat ini. Namun SMS diketahui mempunyai kelemahan-kelemahan yang memungkinkan bocornya informasi yang dikirim. Karena SMS yang dikirim tidak langsung diterima oleh telepon seluler penerima, tapi SMS tersebut harus melewati SMSC, tempat SMS di simpan sementara sebelum dikirim ke nomer tujuan. Oleh karena itu untuk mencegah kebocoran itu perlu di adakannya pengamanan atas SMS yang akan dikirim dengan cara mengenkripsi SMS itu.

Bab II Tinjauan Pustaka, RSA merupakan algoritma kriptografi asimetris yang ini menawarkan keamanan yang lebih tinggi dan lebih kompleks daripada metode konvensional. Keamanan dari sistem kriptografi RSA adalah didasari oleh dua problem matematika yaitu problem dalam faktorisasi bilangan berjumlah banyak dan problem RSA, yaitu mencari modulo akar e dari sebuah bilangan komposit *n* yang faktor-faktornya tidak diketahui.

Bab III Analisis dan Perancangan Aplikasi, Bab ini menjelaskan tentang pembuatan analisis dan perancangan program aplikasi enkripsi SMS dengan menggunakan metode RSA pada telepon seluler berbasis Android.

Bab IV Hasil dan Pembahasan, Bab ini menjelaskan tentang implementasi dari sistem yang telah dibuat kedalam bentuk sebuah program aplikasi secara keseluruhan.

Bab V Penutup, Bab ini merupakan penutup, yang di dalamnya berisi kesimpulan dan rangkuman dari pembahasan penelitian ini, serta berisi saran yang diharapkan dapat bermanfaat untuk pengembangan pembuatan program aplikasi selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Kriptografi

Dalam ilmu kriptografi terdapat beberapa metode yang cukup penting dalam pengamanan data, untuk menjaga kerahasiaan suatu data salah satunya adalah enkripsi (*encryption*). Enkripsi adalah suatu proses yang dilakukan untuk mengubah pesan asli menjadi *chipertext.* Sedangkan suatu proses yang dilakukan untuk mengubah pesan tersembunyi menjadi pesan biasa (yang mudah dibaca) disebut dekripsi. Pesan biasa atau pesan asli biasa disebut dengan *plaintext* sedangkan pesan yang telah diubah atau disandikan supaya tidak mudah dibaca disetut dengan *chipertext.* (Eko Aribowo, 2008)

Prinsip dasarnya adalah menyembunyikan informasi sedemikian rupa agar orang yang berhak saja yang dapat mengetahui isi dari informasi yang tersembunyi tersebut. Teknik ini sudah ada sejak jaman dahulu, bahkan sejak jaman sebelum Masehi pada masa perang yang digunakan untuk mengirim pesan rahasia antar sesama kawan agar apabila pesan terbaca oleh musuh ditengah jalan, isi dari pesan tersebut tidak dapat terbaca. Seiring dengan kemajuan teknik yang digunakan untuk mengenkripsi maka didalamnya terkandung unsur matematis yang membuat isi dari informasi itu semakin sulit untuk dibongkar.

Ada empat tujuan mendasar dari ilmu kriptografi ini yang juga merupakan aspek keamanan informasi yaitu: (Ivan Wibowo, Budi Susanto dan Junius Karel T, 2009)

1. *Confidelity* (kerahasiaan) yaitu layanan agar isi pesan yang dikirimkan tetap rahasia dan tidak diketahui oleh pihak lain (kecuali pihak pengirim, pihak

penerima / pihak-pihak memiliki ijin). Umumnya hal ini dilakukan dengan cara membuat suatu algoritma matematis yang mampu mengubah data hingga menjadi sulit untuk dibaca dan dipahami.

2. *Data integrity* (keutuhan data) yaitu layanan yang mampu mengenali/ mendeteksi adanya manipulasi (penghapusan, pengubahan atau penambahan) data yang tidak sah (oleh pihak lain).
3. *Authentication* (keotentikan) yaitu layanan yang berhubungan dengan identifikasi. Baik otentikasi pihak-pihak yang terlibat dalam pengiriman data maupun otentikasi keaslian data/informasi.
4. *Non-repudiation* (anti-penyangkalan) yaitu layanan yang dapat mencegah suatu pihak untuk menyangkal aksi yang dilakukan sebelumnya (menyangkal bahwa pesan tersebut berasal dirinya).

2.2 Algoritma Ron Rivest, Adi Shamir dan Leonard Adleman (RSA)

RSA merupakan algoritma kriptografi asimetris. Ditemukan pertama kali pada tahun 1977 oleh Ron Rivest, Adi Shamir, dan Leonard Adleman, peneliti dari *Massachussets Institute of Technology* (MIT). Sebagai algoritma kunci publik, RSA mempunyai dua kunci, yaitu kunci publik dan kunci pribadi. Kunci publik boleh diketahui oleh siapa saja, dan digunakan untuk proses enkripsi. Sedangkan kunci pribadi hanya pihak-pihak tertentu saja yang boleh mengetahuinya, dan digunakan untuk proses dekripsi. (Riyanto dan Ardhi, 2008).

RSA merupakan algoritma pertama yang cocok untuk digital signature seperti halnya ekripsi, dan salah satu yang paling maju dalam bidang kriptografi kunci publik. RSA masih digunakan secara luas dalam protokol *electronic*

commerce, dan dipercaya dalam mengamankan dengan menggunakan kunci yang cukup panjang.

Metode enkripsi kunci publik seperti RSA ini menawarkan keamanan yang lebih tinggi dan lebih kompleks daripada metode konvensional. Keamanan dari sistem kriptografi RSA adalah didasari oleh dua problem matematika:

1. Problem dalam faktorisasi bilangan berjumlah banyak.
2. Problem RSA, yaitu mencari modulo akar e dari sebuah bilangan komposit *n* yang faktor-faktornya tidak diketahui.

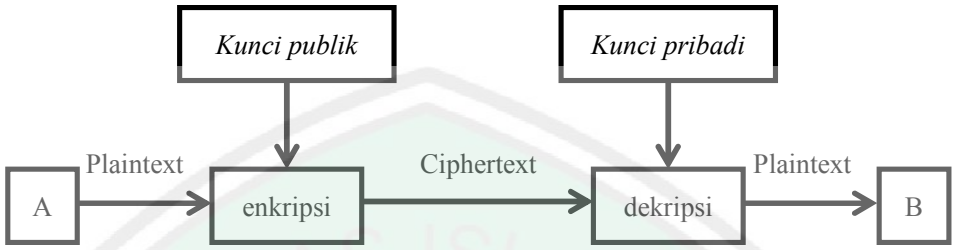
Syarat yang harus dipenuhi di dalam metode enkripsi kunci publik :

1. Mudah bagi user untuk membentuk sepasang kunci (kunci publik dan kunci pribadi).
2. Mudah bagi user lain untuk mengetahui kunci publik milik kita dan kemudian mengenkripsi pesan yang akan dikirimkan kepada kita dengan kunci tersebut.
3. Mudah bagi penerima pesan untuk mendekripsi pesan ciphertext yang telah diterima dengan kunci pribadi yang dimilikinya.
4. Tidak mudah bagi musuh untuk mengetahui kunci publik untuk menentukan kunci pribadi yang dimiliki oleh user lain. (Prasetyo Andy Wicaksono, 2009)

Dengan menggunakan teknik enkripsi data, sebuah pesan text dapat dienkode sedemikian sehingga sangat tidak beraturan dan sulit untuk dikembalikan ke pesan asal tanpa kunci rahasia. Pesan tersebut dapat berupa ASCII, file database atau data apapun yang akan kita kirimkan atau kita simpan melalui media yang tidak aman. Dalam kontek kriptografi, plaintext adalah data yang belum di enktripsi sedangkan ciphertext adalah data yang telah dienkripsi.

Jika sebuah pesan telah dienkripsi maka pesan tersebut dapat disimpan atau ditransmisikan dalam media yang tidak aman namun tetap terjaga kerahasiannya.

Kemudian, pesan tersebut dapat didekripsi kedalam bentuk aslinya. Ilustrasi proses tersebut dapat dilihat pada gambar dibawah.



Gambar 2.1 Ilustrasi proses enkripsi dan dekripsi.

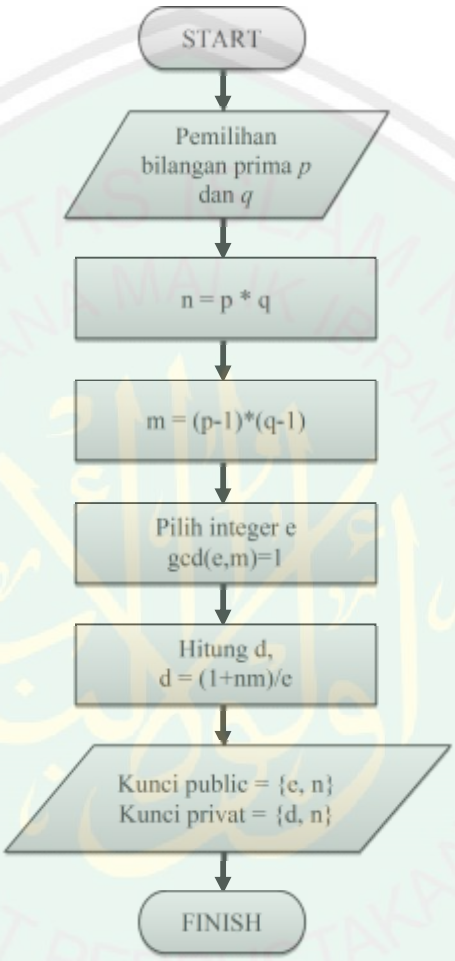
Pada saat pesan dienkripsi maka kunci enkripsi digunakan untuk proses tersebut. Hal ini sama dengan kunci yang biasa kita gunakan untuk mengunci gembok pintu. Untuk mendekripsikan pesan, maka kunci dekripsi yang cocok harus digunakan. Dalam hal ini, sangatlah penting untuk membatasi akses kepada kunci dekripsi, sebab semua orang yang dapat melihat kunci dekripsi berarti dapat pula mendekripsikan semua pesan yang telah dienkripsi.

Pada algoritma RSA terdapat 3 langkah utama yaitu *key generation* (pembangkitan kunci), enkripsi, dan dekripsi. (Moh. Yos Rizal, 2012)

1. Pembangkitan kunci atau *key generation* dari RSA adalah sebagai berikut:
    a. Hasilkan dua buah integer prima besar, p dan q Untuk memperoleh tingkat keamanan yang tinggi pilih p dan q yang berukuran besar, misalnya 1024 bit.
    b. Hitung n = p*q
    c. Hitung m = (p-1)*(q-1)
    d. Pilih d yang relative prima terhadap m, e relative prima terhadap m artinya faktor pembagi terbesar keduanya adalah 1, secara matematis disebut gcd(e,m) = 1. Untuk mencarinya dapat digunakan algoritma Euclid.

e. Cari d, sehingga e*d = 1 mod (m), atau d = (1+nm)/e Untuk bilangan besar, dapat digunakan algoritma extended Euclid.

f. Kunci publik : e, n Kunci private : d, n



Gambar 2.2 Flowchart pembangkitan kunci RSA

2. Proses enkripsi dapat dilakukan dengan :

$$C_i = P_i^e \bmod n$$

Dimana, C : Chipertext

P : Plaintext

e : kunci publik

n : modulus yang digunakan.

3. Proses dekripsi dapat dilakukan dengan:

$$P_i = C_i^d \bmod n$$

Dimana, C : Chipertext

P : Plaintext

d : kunci privat

n : modulus yang digunakan.

Kekuatan algoritma RSA terletak pada tingkat kesulitan dalam memfaktorkan bilangan menjadi faktor primanya, dalam hal ini memfaktorkan *n* menjadi *p* dan *q*. Karena sekali *n* berhasil difaktorkan, maka menghitung nilai *m* adalah perkara mudah. Selanjutnya, walau nilai *e* diumumkan, perhitungan kunci *d* tidaklah mudah pula karena nilai *m* yang tidak diketahui.

Keuntungan utama dari RSA yang merupakan kriptografi kunci publik adalah menambah keamanan dan kenyamanan. Kunci privat tidak pernah diperlukan untuk dikirim atau diberi tahu ke orang lain. Pada sebuah sistem kunci rahasia, secara terang-terangan kunci rahasia ini harus dikirim (bisa secara manual atau melalui sebuah saluran komunikasi), dan akan terjadi suatu kemingkinan dimana penyerang dapat mencari tahu kunci rahasia tersebut saat proses pengiriman.

Kekurangan dari pemakaian kriptografi kunci publik, dalam hal ini RSA, adalah dalam masalah kecepatan. Banyak metode enkripsi kunci rahasia yang populer yang memiliki kecepatan enkripsi-dekripsi yang lebih cepat

dibandingkan dengan metode enkripsi kunci publik yang ada sekarang. Namun kriptografi kunci publik dapat digunakan dengan kriptografi kunci rahasia untuk mendapatkan metode enkripsi yang terbaik di dunia.

Untuk enkripsi, solusi terbaik adalah dengan cara mengkombinasi sistem kunci publik dan sistem kunci rahasia untuk mendapatkan kedua keuntungan yang dimiliki oleh metode enkripsi ini, keuntungan keamanan dari segi sistem kunci publik, dan keuntungan kecepatan dari segi sistem kunci rahasia. Sistem kunci publik dapat digunakan untuk mengenkripsi sebuat kunci rahasia, yang bisa digunakan untuk mengenkripsi file atau pesan yang berukuran besar sekalipun.

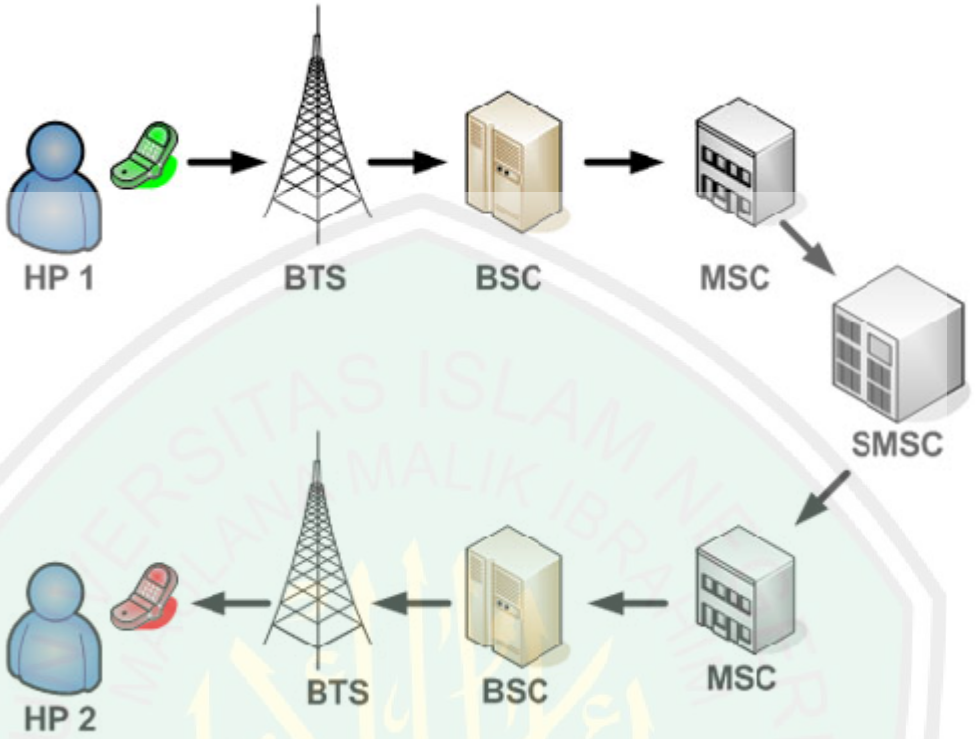
2.3 Teknologi SMS

Layanan *Short Message Service* (SMS) sangat populer dan sering dipakai oleh pengguna *handphone*. SMS menyediakan pengiriman pesan text secara cepat, mudah dan murah. Kini SMS tidak terbatas untuk komunikasi antar manusia pengguna saja, namun juga bisa dibuat otomatis dikirim/diterima oleh peralatan (komputer, mikrokontroler, dsb) untuk mencapai suatu tujuan tertentu. Namun untuk melakukannya, kita harus memahami dulu cara kerja SMS itu sendiri. SMS adalah protokol layanan pertukaran pesan text singkat (sebanyak 160 karakter per pesan) antar telepon seluler. SMS ini pada awalnya adalah bagian dari standar teknologi seluler GSM, yang kemudian juga tersedia di teknologi CDMA, telepon rumah PSTN, dan lainnya.

Pada jaringan mobile telekomunikasi, trasnportasi data pada layanan SMS dapat dilakukan pada jaringan GSM dan GPRS. SMS berbentuk bilangan biner

yang memuat informasi penting untuk menghasilkan message header untuk trasnsportasi data dan messsage body sebagai *payload*. Skema dasar pengalamatan SMS adalah nomor mobile phone yang disebut *Mobile Station International Subscriber Directory Number* (MSISDN).

Ketika pengguna mengirim SMS, maka pesan dikirim ke *Mobile Switcching Center* (MSC) melalui jaringan seluler yang tersedia yang meliputi tower *Base Transceiver Station* (BTS) yang sedang meng-handle komunikasi pengguna, lalu ke *Base Station Controller* (BSC), kemudian sampai ke MSC. MSC kemudian mem-forward lagi SMS ke SMSC untuk disimpan. SMSC kemudian mengecek (lewat HLR – Home Location Register) untuk mengetahui apakah handphone tujuan sedang aktif dan dimanakah handphone tujuan tersebut. Jika handphone sedang tidak aktif maka pesan tetap disimpan di SMSC itu sendiri, menunggu MSC memberitahukan bahwa handphone sudah aktif kembali untuk kemudian SMS dikirim dengan batas maksimum waktu tunggu yaitu validity period dari pesan SMS itu sendiri. Jika handphone tujuan aktif maka pesan disampaikan MSC lewat jaringan yang sedang meng-handle penerima (BSC dan BTS).



Gambar 2.3 Alur pengiriman SMS.
(Sumber: http://ciptamedia-sms-broadcast.blogspot.com/2010/06/mekanisme-pengiriman-sms-dari-hp.html)

2.4 Android

Android adalah sebuah *Operating System* (OS) yang dikembangkan oleh Google untuk *mobile device* atau yang lebih kita kenal sebagai *smartphone*. OS ini bersifat *open source*.

Salah satu kelebihan dari Android adalah ketersediaan aplikasi dari berbagai macam kategori: sosial, hiburan, permainan, dsb. Para developer bisa mengembangkan aplikasi sesuai dengan minat mereka masing-masing menggunakan Software Development Kit (SDK) yang telah didistribusikan oleh Google. Karena Android adalah OS dengan bentuk open source, OS ini dapat

terus dikembangkan dan memiliki evolusi yang sangat cepat sesuai dengan pertambahan jumlah aplikasi. (Nazaruddin Safaat H, 2012)

2.4.1 Sejarah Android

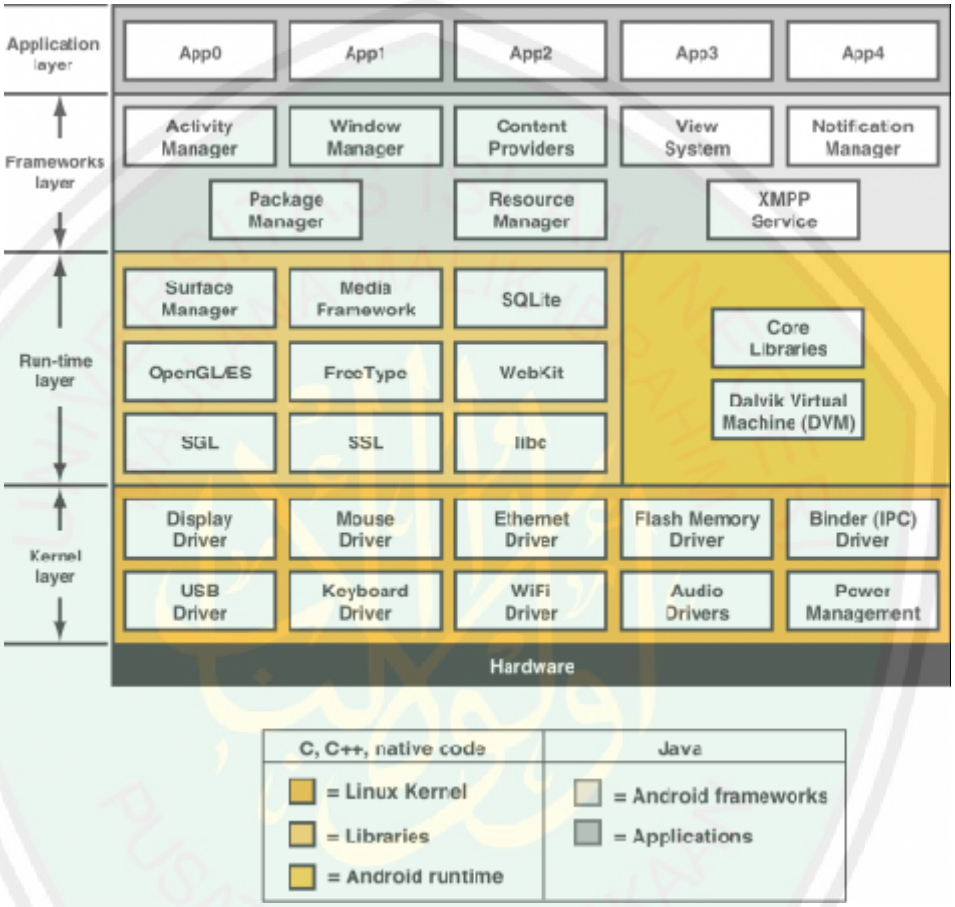
Pada bulan Juli tahun 2005, Google mengakuisisi banyak perusahaan *start-up*, termasuk sebuah perusahaan kecil perancang software untuk mobile phones bernama Android Inc. Salah satu pendiri Android Inc adalah Andy Rubin, yang sekarang menjabat sebagai *Director of Mobile Platforms* di Google.

Akuisisi Android Inc. diikuti oleh rumor tentang rencana Google membuat *handset*-nya sendiri untuk membantu pengembangan fungsi *mobile search*-nya. Kabarnya juga, *handset* ini akan menghadirkan *location-based services* serta mengimplementasikan semua ide dari Google Labs, termasuk dua aplikasi terfavorit, *Maps* dan *Mail*.

Setelah penantian cukup panjang, akhirnya perusahaan yang berbasis di California ini mengumumkan pada 5 November 2007 bahwa mereka sedang merancang sebuah open-source OS baru bernama Android untuk bersaing dengan Symbian, Microsoft, dan lain-lain. (Husen Syariati, 2012)

2.4.2 Arsitektur Android

Diagram berikut menunjukkan komponen-komponen utama dari sistem operasi Android.



Gambar 2.4 Arsitektur Android
(Sumber: https://community.freescale.com/community/the-embedded-beat/blog/2010/05/24/android-makes-the-move-to-power-architecture-technology)

1. Linux Kernel

Android dibangun di atas kernel Linux 2.6. Namun secara keseluruhan android bukanlah linux, karena dalam android tidak terdapat paket standar yang dimiliki oleh linux lainnya. Linux merupakan sistem operasi terbuka yang handal dalam manajemen memori dan proses. Oleh karenanya pada android hanya

terdapat beberapa servis yang diperlukan seperti keamanan, manajemen memori, manajemen proses, jaringan dan driver. Kernel linux menyediakan driver layar, kamera, keypad, WiFi, Flash Memory, audio, dan *Interprocess Communication* (IPC) untuk mengatur aplikasi dan lubang keamanan.

2. Libraries

Android menggunakan beberapa paket pustaka yang terdapat pada C/C++ dengan standar Berkeley Software Distribution (BSD) hanya setengah dari yang aslinya untuk tertanam pada kernel Linux. Beberapa pustaka diantaranya:

a. Media Library untuk memutar dan merekam berbagai macam format audio dan video.
b. Surface Manager untuk mengatur hak akses layer dari berbagai aplikasi.
c. Graphic Library termasuk didalamnya SGL dan OpenGL, untuk tampilan 2D dan 3D.
d. SQLite untuk mengatur relasi database yang digunakan pada aplikasi.
e. SSl dan WebKit untuk browser dan keamanan internet.

Pustaka-pustaka tersebut bukanlah aplikasi yang berjalan sendiri, namun hanya dapat digunakan oleh program yang berada di level atasnya. Sejak versi Android 1.5, pengembang dapat membuat dan menggunakan pustaka sendiri menggunakan Native Development Toolkit (NDK).

3. Android Runtime

Pada android tertanam paket pustaka inti yang menyediakan sebagian besar fungsi android. Inilah yang membedakan Android dibandingkan dengan sistem operasi lain yang juga mengimplementasikan Linux. Android Runtime merupakan mesin virtual yang membuat aplikasi android menjadi lebih tangguh dengan paket

pustaka yang telah ada. Dalam Android Runtime terdapat 2 bagian utama, diantaranya:

a. Android dikembangkan melalui bahasa pemrograman Java, tapi Android Runtime bukanlah mesin virtual Java. Pustaka inti android menyediakan hampir semua fungsi yang terdapat pada pustaka Java serta beberapa pustaka khusus android.
b. Mesin Virtual Dalvik, Dalvik merupakan sebuah mesin virtual yang dikembangkan oleh Dan Bornstein yang terinspirasi dari nama sebuah perkampungan yang berada di Iceland. Dalvik hanyalah interpreter mesin virtual yang mengeksekusi file dalam format Dalvik Executable (*.dex). Dengan format ini Dalvik akan mengoptimalkan efisiensi penyimpanan dan pengalamatan memori pada file yang dieksekusi. Dalvik berjalan di atas kernel Linux 2.6, dengan fungsi dasar seperti threading dan manajemen memori yang terbatas.(Nicolas Gramlich, Andbook, anddev.org)

4. *Application Framework*

Kerangka aplikasi menyediakan kelas-kelas yang dapat digunakan untuk mengembangkan aplikasi android. Selain itu, juga menyediakan abstraksi generik untuk mengakses perangkat, serta mengatur tampilan user interface dan sumber daya aplikasi. Bagian terpenting dalam kerangka aplikasi android adalah sebagai berikut:

a. *Activity Manager*, berfungsi untuk mengontrol siklus hidup aplikasi dan menjaga keadaan ”*Backstack*“ untuk navigasi penggunaan.
b. *Content Providers*, berfungsi untuk merangkum data yang memungkinkan digunakan oleh aplikasi lainnya, seperti daftar nama.

c. *Resuource Manager*, untuk mengatur sumber daya yang ada dalam program. Serta menyediakan akses sumber daya diluar kode program, seperti karakter, grafik, dan file layout.
d. *Location Manager*, berfungsi untuk memberikan informasi detail mengenai lokasi perangkat android berada.
e. *Notification Manager*, mencakup berbagai macam peringatan seperti, pesan masuk, janji, dan lain sebagainya yang akan ditampilkan pada status bar.

5. *Application Layer*

Puncak dari diagram arsitektur android adalah lapisan aplikasi dan *widget*. Lapisan aplikasi merupakan lapisan yang paling tampak pada pengguna ketika menjalankan program. Pengguna hanya akan melihat program ketika digunakan tanpa mengetahui proses yang terjadi dibalik lapisan aplikasi. Lapisan ini berjalan dalam Android runtime dengan menggunakan kelas dan servis yang tersedia pada framework aplikasi.

Lapisan aplikasi android sangat berbeda dibandingkan dengan sistem operasi lainnya. Pada android semua aplikasi, baik aplikasi inti (*native*) maupun aplikasi pihak ketiga berjalan diatas lapisan aplikasi dengan menggunakan pustaka *Application Programming Interface* (API) yang sama.

2.4.3 Komponen Aplikasi

Fitur penting android lainnya adalah bahwa satu aplikasi dapat menggunakan elemen dari aplikasi lain (untuk aplikasi yang memungkinkan). Sebagai contoh, sebuah aplikasi memerlukan fitur scroller dan aplikasi lain telah mengembangkan fitur scroller yang baik dan memungkinkan aplikasi lain menggunakannya. Maka pengembang tidak perlu lagi mengembangkan hal serupa untuk aplikasinya, cukup menggunakan scroller yang telah ada.

Agar fitur tersebut dapat bekerja, sistem harus dapat menjalankan aplikasi ketika setiap bagian aplikasi itu dibutuhkan, dan pemanggilan objek java untuk bagian itu. Oleh karenanya android berbeda dari sistem-sistem lain, Android tidak memiliki satu tampilan utama program seperti fungsi main() pada aplikasi lain.

2.4.4 Tipe Aplikasi Android

Terdapat tiga kategori aplikasi pada android: (Reto Meier, Profesional Android Application Development, Wiley Publishing, Canada, 2009)

1. *Foreground Activity*

Aplikasi yang hanya dapat dijalankan jika tampil pada layar dan tetap efektif walaupun tidak terlihat. Aplikasi dengan tipe ini pasti mempertimbangkan siklus hidup activity, sehingga perpindahan antar activity dapat berlangsung dengan lancar.

2. *Background Service*

Aplikasi yang memiliki interaksi terbatas dengan user, selain dari pengaturan konfigurasi, semua dari prosesnya tidak tidak tampak pada layar. Contohnya aplikasi penyaringan panggilan atau sms auto respon.

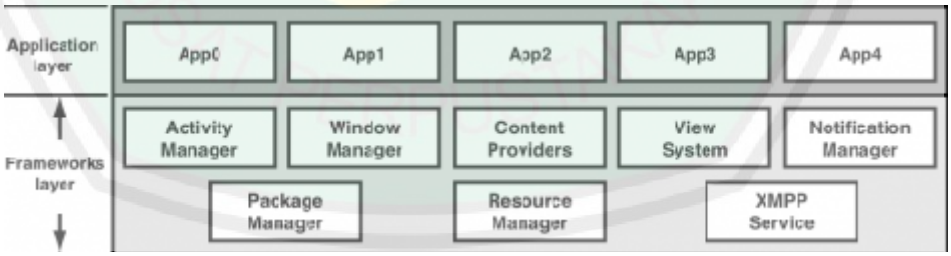
3. *Intermittent Activity*

Aplikasi yang masih membutuhkan beberapa masukkan dari pengguna, namun sebagian sangat efektif jika dijalankan di background dan jika diperlukan akan memberi tahu pengguna tentang kondisi tertentu. Contohnya pemutar musik. Untuk aplikasi yang kompleks akan sulit untuk menentukan kategori aplikasi tersebut apalagi aplikasi memiliki ciri-ciri dari semua kategori. Oleh karenanya perlu pertimbangan bagaimana aplikasi tersebut digunakan dan menentukan kategori aplikasi yang sesuai.

2.4.5 Siklus Hidup Aplikasi Android

Siklus hidup aplikasi android dikelola oleh sistem, berdasarkan kebutuhan pengguna, sumberdaya yang tersedia, dan sebagainya. Misalnya Pengguna ingin menjalankan browser web, pada akhirnya sistem yang akan menentukan menjalankan aplikasi. Sistem sangat berperan dalam menentukan apakah aplikasi dijalankan, dihentikan sementara, atau dihentikan sama sekali. Jika pengguna ketika itu sedang menjalankan sebuah Activity, maka sistem akan memberikan perioritas utama untuk aplikasi yang tersebut. Sebaliknya, jika suatu Activity tidak terlihat dan sistem membutuhkan sumber daya yang lebih, maka Activity yang prioritas rendah akan ditutup. (Sayed . Y. Hashimi dan Satya Komatineni, 2009)

Android menjalankan setiap aplikasi dalam proses secara terpisah, yang masing-masing memliki mesin virtual pengolah sendiri, dengan ini melindungi penggunaan memori pada aplikasi. Selain itu juga android dapat mengontrol aplikasi mana yang layak menjadi prioritas utama. Karenanya android sangat sensitive dengan siklus hidup aplikasi dan komponen-komponennya. Perlu adanya penanganan terhadap setiap kondisi agar aplikasi menjadi stabil.



Gamabar 2.5 *Aplication Layer* dan *Framework Layer*
(Sumber: https://community.freescale.com/community/the-embedded-beat/blog/2010/05/24/android-makes-the-move-to-power-architecture-technology)

Aplikasi yang nantinya akan dibuat terletak pada layer Aplikasi. Namun secara proses tidak lepas dari Layer Framework, karena semua proses yang

dibutuhkan ada pada layer ini, seperti *Notofication Manager* yang berfungsi untuk memberi peringatan ketika ada pesan masuk. Kemudian *Content Provider* yang berfungsi untuk merangkum data yang memungkinkan digunakan oleh aplikasi lainnya, seperti daftar nama.

Kemudian tipe aplikasi yang akan dibuat termasuk dalam tipe *Intermittent Activity,* dimana aplikasi ini masih membutuhkan beberapa masukkan dari pengguna, namun sebagian sangat efektif jika dijalankan di background dan jika diperlukan akan memberi tahu pengguna tentang kondisi tertentu.

BAB III

PERANCANGAN SISTEM

3.1 Perancangan Sistem

3.1.1 Deskripsi Umum Sistem

Enkripsi merupakan salah satu cara untuk mengamankan suatu pesan atau informasi dengan cara mengacak pesan tersebut sehingga tidak bisa terbaca oleh pihak lain. Pada proses enkripsi harus disertakan kunci agar pesan yang dienkripsi bisa didekripsikan kembali dengan menggunakan kunci tersebut. Di dalam penelitian ini pesan yang akan dienkripsi adalah pesan yang dikirim melalui fasilitas *Short Message Service* (SMS) yang terdapat pada telepon seluler menggunakan ilmu kriptografi asimetris *Rivest Shamit Adleman* (RSA).

Kriptografi asimetris memiliki dua kunci yaitu kunci publik untuk mengenkripsi pesan dan kunci privat untuk mendekripsi pesan. Untuk memperoleh sepasang kunci tersebut, sistem melakukan perhitungan/*genetare* yang kemudian tesimpan sebagai pasangan kunci dari sistem itu sendiri. Kemudian kunci publik akan dikirim melalui SMS kepada *user* lain yang akan bertukar pesan.

Tahapan proses dalam mengenkripsi pesan adalah *user* menginputkan pesan teks yang kemudian akan di enkripsi oleh sistem. Dimana, untuk mengenkripsi pesan tersebut *user* harus mempunyai kunci publik dari *user* lain yang akan bertukar pesan. Begitu pula sebaliknya.

Kemudian untuk membaca pesan masuk yang sudah terenkripsi, sistem akan mendekripsikan secara otomatis menggunakan kunci privat yang sudah di *generate*. Proses *generate key* atau pembangkitan kunci adalah sebagai berikut.

Contoh perhitungan pembangkitan kunci RSA.

Misalkan bilangan prima $p = 47$ dan $q = 71$

Maka,

$n = p \square q = 3337$

$m = (p - 1)(q - 1) = 3220$

mencari d, gcd(e,3337)

misal dipilih = 79

hitung d → e*d = 1 mod (m)

79 * d = 1 mod 3220

79 * d mod 3220 = 1

→ 1019

Kunci publik : e, n Kunci private : d, n

Private key : (1019, 3337)

Public key : (79, 3337)

Setelah didapatkan kunci publik dan kunci pribadi, maka proses selanjutnya yaitu melakukan enkripsi terhadap pesan yang akan dikirim. Proses enkripsi pesan adalah sebagai berikut.

Misalkan plainteks yang akan dienkripsikan adalah

X = HARI INI

Dalam sistem desimal (pengkodean ASCII)adalah

H A RI (SPASI) I N I = 72 65 82 73 32 73 78 73

Pecah X menjadi blok yang lebih kecil,misalnya X dipecah menjadi enam blok yang berukuran 3 digit:

- $x1 = 726$, $x4 = 273$,
- $x2 = 582$, $x5 = 787$,
- $x3 = 733$, $x6 = 003$ (ditambah 0)

Proses pemecahan melihat dalam interval

[0,n-1] → interval [0, 3336]

Blok-blok plainteks dienkripsikan sebagaiberikut:

726^79 mod 3337 = 215 = $y1$

582^79 mod 3337 = 776 = $y2$

733^79 mod 3337 = 1743 = $y3$

273^79 mod 3337 = 933 = $y4$

787^79 mod 3337 = 1731 = $y5$

003^79 mod 3337 = 158 = $y6$

Jadi, cipherteks yang dihasilkan adalah

Y = 215 776 1743 933 1731 158.

- 72 65 82 73 32 73 78 73 → ASLI
- 215 776 1743 933 1731 158→ hasil enkripsi

Dekripsi dilakukan dengan menggunakan kunci rahasia

Blok-blok cipherteks didekripsikan sebagaiberikut:

215^1019 mod 3337 = 726 = $x1$ 1731^1019 mod 3337 = 787 =$x5$

776^1019 mod 3337 = 582 = $x2$ 158^1019 mod 3337 = 3 = $x6$

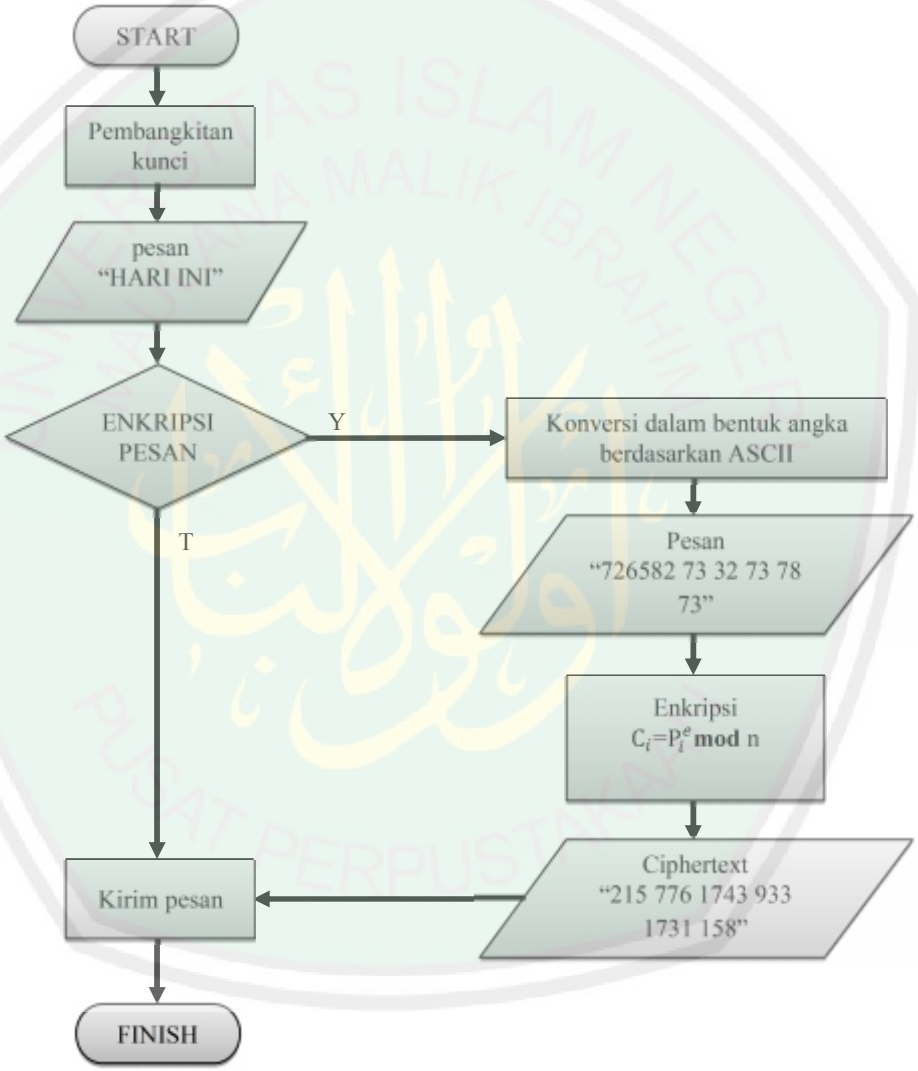
1743^1019 mod 3337 = 733 = $x3$

933^1019 mod 3337 = 273 = $x4$

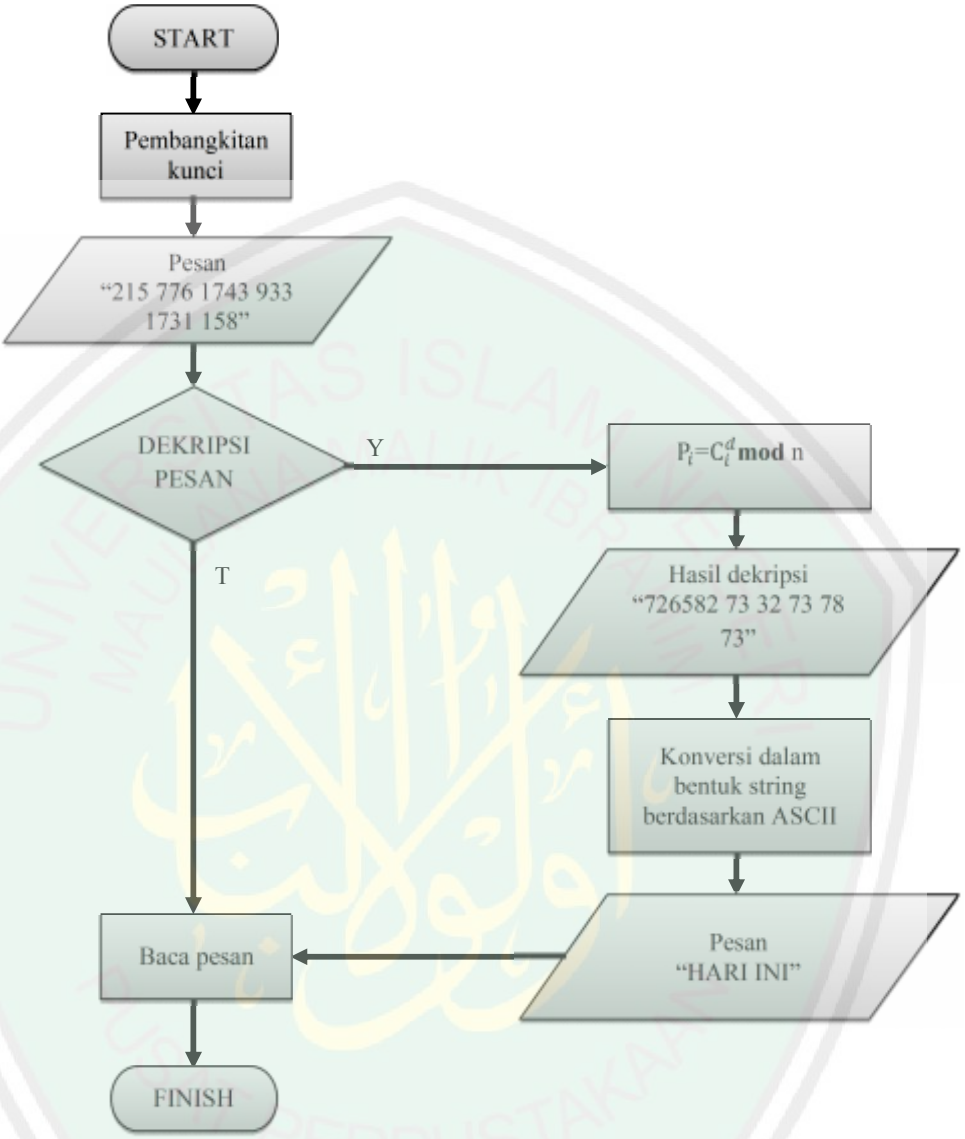
Blok plainteks yang lain dikembalikan dengan cara yang serupa. Akhirnya kita memperoleh kembali plainteks semula

P = 7265827332737873

yang dalam karakter ASCII adalah P = HARI INI.

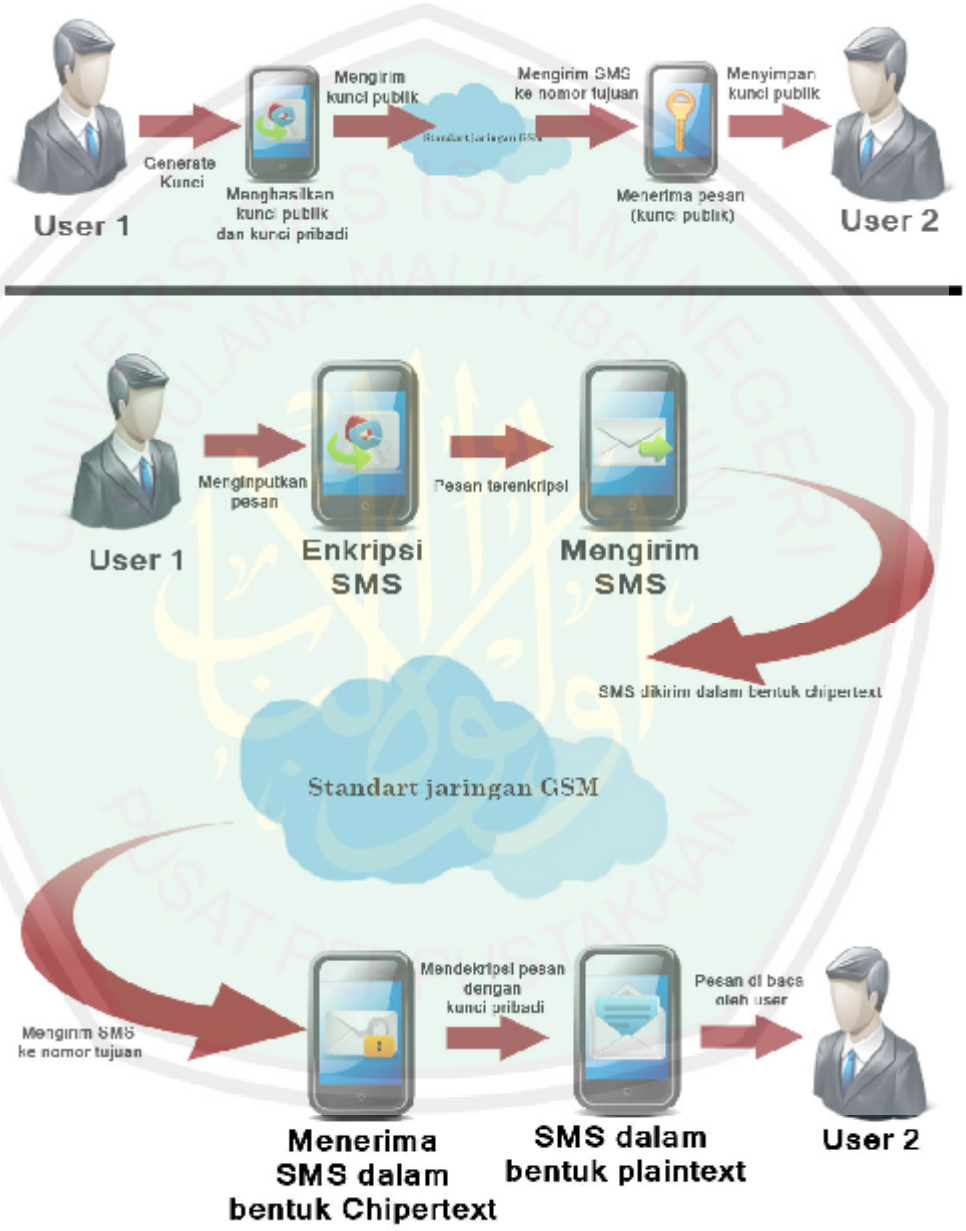


Gambar 3.1 Flowchart enkripsi pesan

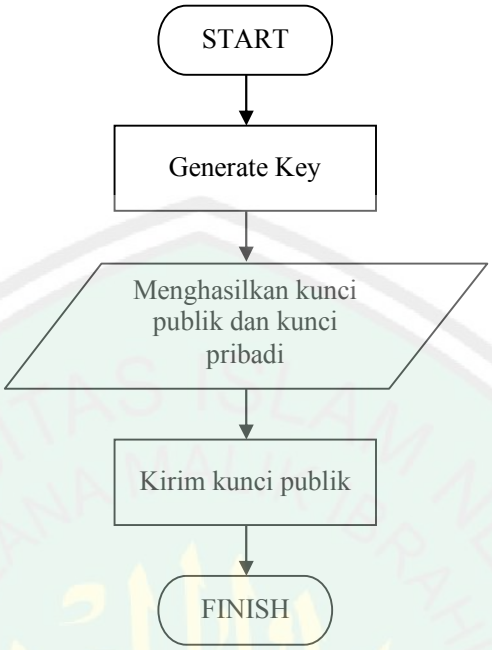


Gambar 3.2 Flowchart dekripsi pesan

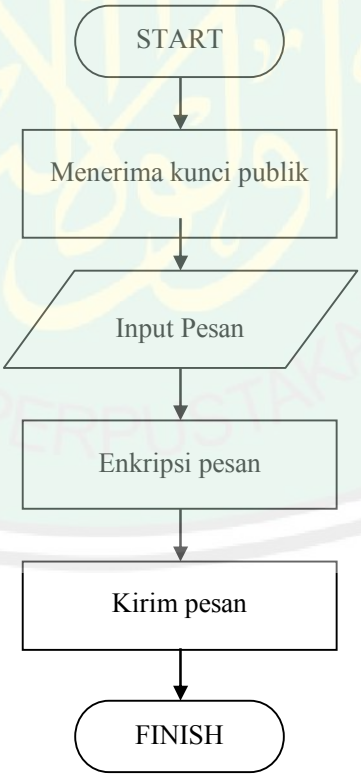
Gambaran umum serta alur sistem pada aplikasi enkripsi SMS dengan metode RSA pada *smartphone* ini dapat dilihat pada **Gambar 3.3**, proses pembangkitan pasangan kunci pada **Gambar 3.4,** proses pengiriman SMS pada **Gambar 3.5** dan proses dekripsi SMS pada **Gambar 3.6.**



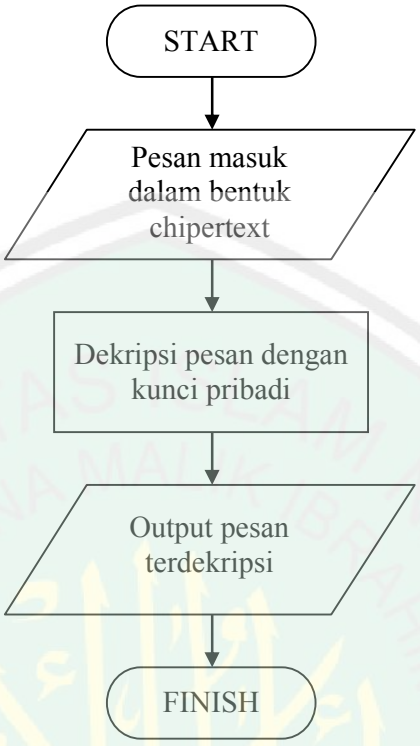
Gambar 3.3 Gambaran umum aplikasi enkripsi SMS



Gambar 3.4 Proses pembangkitan pasangan kunci



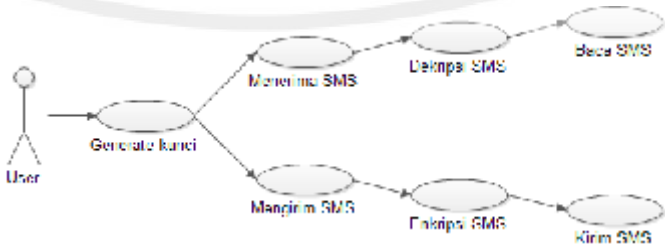
Gambar 3.5 Proses pengiriman SMS



Gambar 3.6 Proses dekripsi SMS

3.1.2 Use Case Diagram

Aktor pada aplikasi enkripsi SMS ini adalah *user* yang berperan sebagai penerima dan pengirim SMS. Aktor berhubungan langsung dengan sistem dengan memasukkan pesan yang akan dikirim ke dalam aplikasi, kemudian sistem akan memproses pesan untuk dienkripsi dan dikirim. *Use case diagram* dari aplikasi ini dapat dilihat pada **Gambar 3.7**.



Gambar 3.7 *Use Case Diagram*

3.2 Analisis Kebutuhan

3.2.1 Kebutuhan *Hardware*

Dalam proses pembuatan aplikasi enkripsi SMS, dibutuhkan perangkat keras yang digunakan untuk menunjang pembuatan sistem. Perangkat keras yang dibutuhkan antara lain:

1. Laptop (*Notebook*) / Komputer

Laptop/Komputer digunakan untuk membangun sistem aplikasi enkripsi SMS menggunakan metode RSA. Dalam penelitian ini, spesifikasi minimum yang akan di gunakan adalah:

a. *OS* : Windows 7
b. *Processsor* : Dual Core
c. *Memory* : DDR2 2 GB
d. *Graphics* : 512 MB

2. *Smartphone*

Smartphone digunakan untuk mengaplikasikan program yang telah dibuat untuk mengetahui apakah aplikasi tersebut berjalan lancar sesuai tujuan penelitian. Aplikasi yang telah berjalan di laptop dan emulator belum tentu dapat berjalan dengan baik pada *Smartphone* sesungguhnya. Dalam penelitian ini *Smartphone* yang digunakan sebagai uji coba adalah *Smartphone* yang berbasis sistem operasi *Android*, dengan minimum spesifikasi sebagai berikut:

a. *OS* : Android OS v2.1
b. *CPU* : 600 MHz
c. *Memory* : 168 MB RAM

3.2.2 Kebutuhan *Software*

Adapun untuk kebutuhan *software* tahap penelitian sampai tahap implementasi dari aplikasi enkripsi SMS dengan menggunakan metode RSA pada *Smartphone,* menggunakan beberapa software sebagai berikut:

1. Windows 7

Windows 7 merupakan sistem operasi yang menjembatani antara komputer dan *user.* Windows 7 digunakan karena *compatible* dengan *software* lain yang dibutuhkan dalam pembuatan aplikasi enkripsi SMS.

2. *Java Development Kit* (JDK)

JDK adalah salah satu produk yang dikeluarkan oleh *Sun Microsystem* yang berguna untuk mengembangkan progam yang ditulis dalam bahasa pemrograman Java.

3. Android *Software Development Kit* (SDK)

Android SDK adalah *tools* API (*Application Programming Interface*) yang diperlukan untuk mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. (Nazruddin, 2012:5)

4. *Android Development Tools* (ADT)

ADT adalah *plugin* untuk *Integrated Development Environment* (IDE) Eclipse yang berisi kumpulan peralatan yang diperlukan untuk mengembangkan aplikasi Android. Dengan menggunakan ADT untuk Eclipse akan memudahkan pembuatan aplikasi *project* Android , membuat GUI aplikasi, dan menambahkan komponen-komponen yang dibutuhkan. ADT juga dapat melakukan pembuatan *package* android (.apk) yang digunakan untuk distribusi aplikasi Android yang dirancang. (Nazruddin, 2012:6)

5. Eclipse

Eclipse adalah suatu IDE yang digunakan untuk membuat dan mengembangkan perangkat lunak dan bisa dijalankan di semua *platform*.

6. Edraw Max

Edraw Max merupakan perangkat lunak yang digunakan untuk membuat membuat berbagai struktur diagram. Dalam penelitian ini Edraw Max digunakan untuk membuat *Use Case Diagram.*

7. *Power Designer*

Power *Designer* adalah program yang digunakan untuk membantu membuat perancangan sistem.

8. *Microsoft Office Word 2007*

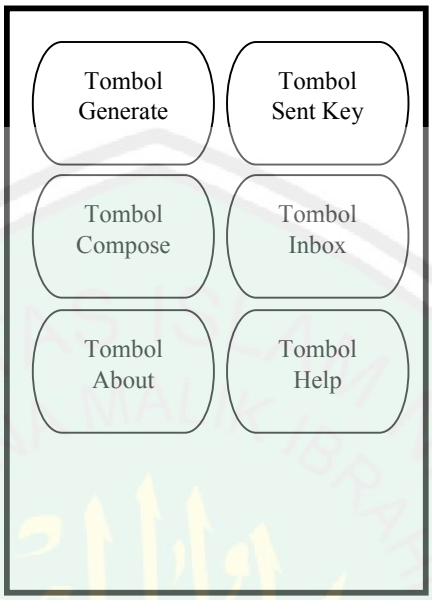
Microsoft *Office Word 2007* digunakan untuk membuat dan menyusun laporan dari hasil penelitian yang telah dilakukan.

3.3 Perancangan Antar Muka

Perancangan antar muka dalam aplikasi ini dibagi menjadi 6 menu yaitu: *Generate Keys*, *Sent My Public Key*, *Compose, Inbox, Help dan About* seperti yang terlihat pada **Gambar 3.8.**

3.3.1 Menu *Generate Keys*

Menu *Generate Keys* merupakan menu untuk menghasilkan pasangan kunci yang akan digunakan untuk mengenkripsi dan mendekripsikan pesan. Desain antar muka menu *Generate Keys* dapat dilihat pada **Gambar 3.9.**



Gambar 3.8 Rancangan Antar Muka Menu Utama

Di dalam menu *Generate Keys* terdapat dua tombol, yaitu tombol Generate dan tombol Save. Dimana tombol *generate* berfungsi untuk membuat kunci publik dan kunci privat yang kemudian di tampilkan pada *field Public Key* dan *Private Key*. Kemudian tombol *save* berfungsi untuk menyimpan kunci yang telah di *generate.*

3.3.2 *Sent My Public Key*

Menu *Sent My Public Key* merupakan menu yang berisi kunci publik yang akan dikirim ke pengguna lain. Rancangan antar muka menu *Sent My Public Key* dapat dilihat pada **Gambar 3.10**.

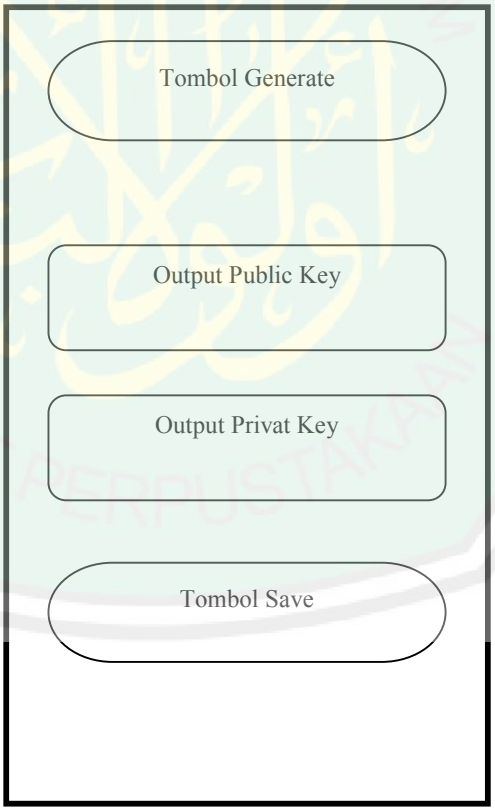
Dalam menu *Sent My Public Key* terdapat *textfield* input nomor telepon yang berfungsi untuk memasukkan nomor telepon yang dituju. Kemudian *textview Public Key* berfungsi untuk menampilkan kunci publik yang akan dikirim. Dan tombol kirim berfungsi untuk mengirim kunci publik tersebut.

3.3.3 *Compose*

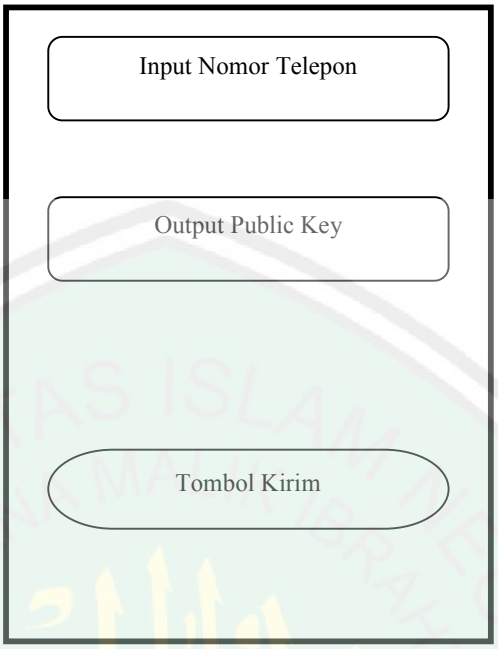
Menu *Compose* merupakan menu yang berisi tentang bagaimana aplikasi akan mengirim pesan kepada pengguna lain.

Rancangan antar muka menu *Compose* dapat dilihat pada **Gambar 3.11**.

Dalam menu *Compose* ini terdapat dua tombol, yaitu tombol enkripsi dan tombol kirim. Dimana tombol enkripsi berfungsi untuk mengenkripsi pesan yang di inputkan *user* pada *field* input pesan berdasarkan kunci publik yang muncul pada *textview Public Key*. Kemudian tombol kirim berfungsi untuk mengirim pesan yang telah diinputkan user kepada nomor yang telah di inputkan pada field input nomor telepon.



Gambar 3.9 Rancangan Antar Muka Menu *Generate Keys*



Gambar 3.10 Rancangan Antar Muka Menu *Sent My Public Key*

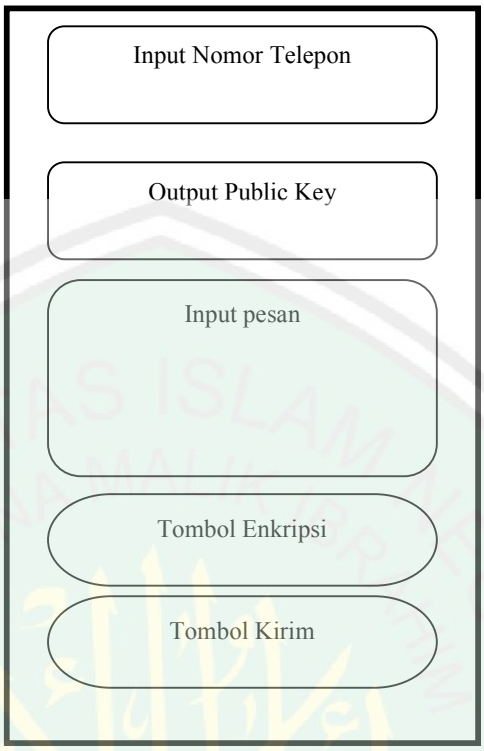
3.3.4 *About*

Menu *About* merupakan menu yang berisi tentang bagaimana aplikasi *Enkripsi SMS* ini dibuat. Rancangan antar muka menu *about* dapat dilihat pada **Gambar 3.12**.

3.3.5 *Help*

Menu *help* merupakan menu yang berisi tentang bagaimana cara menggunakan aplikasi Enkripsi *SMS*.

Rancangan antar muka menu *help* dapat dilihat pada **Gambar 3.13**.



Gambar 3.11 Rancangan Antar Muka Menu *Compose*

Gambar 3.12 Rancangan Antar Muka Menu *About*

Gambar 3.13 Rancangan Antar Muka Menu *Help*

3.4 Perancangan Uji Coba Sistem

Tujuan dari perancangan uji coba ini adalah untuk mengetahui hasil dari implementasi sistem maupun antar muka yang telah di desain sebelumnya. Pengujian dilakukan dengan cara mencoba semua fungsi dari aplikasi enkripsi SMS mulai dari pembangkitan kunci, enkripsi SMS sampai dekripsi SMS. Kemudian dilakukan ujicoba terhadap pesan yang akan dikirim. Hasil pengujian akan dilampirkan dalam bentuk tabel seperti yang terlihat pada Tabel 3.1.

Tabel 3.1 Hasil pengujian pesan yang dikirim.

No.	Pesan	Hasil Enkripsi	Jumlah Karakter	Keterangan

Pengujian selanjutnya yaitu menginstal aplikasi pada berbagai tipe *smartphone* yang berbasis Android dengan tujuan untuk mengetahui apakah

aplikasi enkripsi SMS ini bisa berjalan dan berfungsi dengan baik pada setiap *smartphone* yang dijadikan ujicoba. Hasil pengujian akan dilampirkan dalam bentuk tabel seperti yang terlihat pada Tabel 3.2.

Tabel 3.2 Hasil pengujian terhadap berbagai tipe *smartphone*

No.	Jenis *Smartphone*	Keterangan	Catatan

BAB IV

HASIL DAN PEMBAHASAN

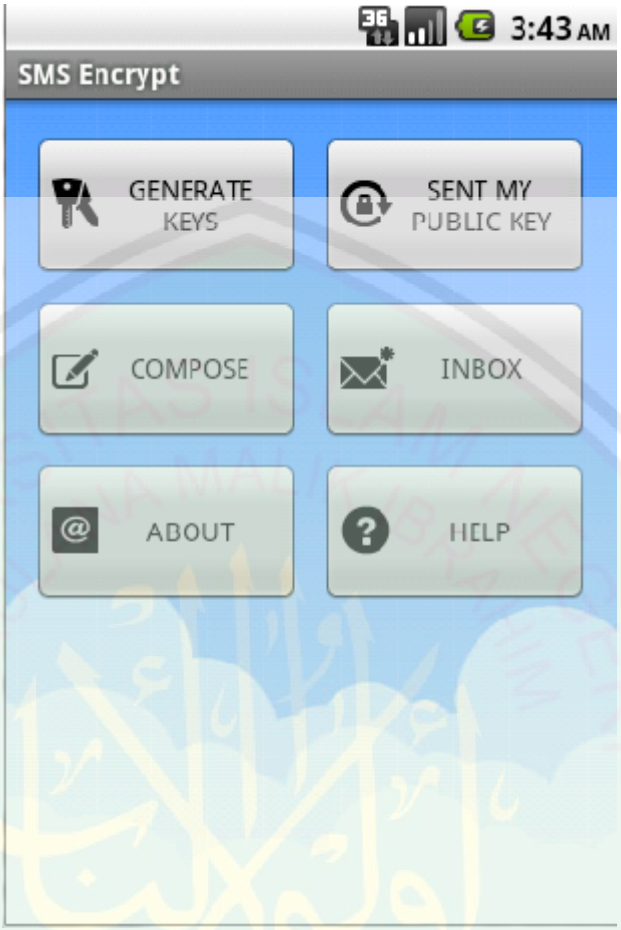
4.1 Implementasi

Implementasi adalah proses penerapan rancangan ke dalam bahasa pemrograman yang dapat dimengerti oleh komputer. Implementasi sistem membuat dan menerapkan sistem secara utuh baik dari sisi perangkat keras maupun perangkat lunaknya. Pada bagian implementasi ini akan dibahas hal-hal yang berkaitan dengan implementasi sistem enkripsi *Short Message Service* (SMS) dengan menggunakan metode RSA pada *Smartphone* sesuai dengan perancangan sistem pada Bab 3.

4.1.1 Implementasi Sistem dan Antar Muka

Menu utama di dalam aplikasi enkripsi SMS dengan menggunakan metode RSA pada *Smartphone* merupakan tampilan menu yang muncul pertama kali ketika aplikasi ini dijalankan. Menu utama dari aplikasi enkripsi SMS dengan menggunakan metode RSA ditunjukkan oleh **Gambar 4.1**.

Menu yang dapat dipanggil dari menu utama aplikasi adalah menu *Generate Keys* (**Gambar 4.10**), menu *Sent My Public Key* (**Gambar 4.11**), menu *Compose* (**Gambar 4.12**), menu *Inbox* (**Gambar 4.13**), menu *Help* (**Gambar 4.14**) dan menu *About* (**Gambar 4.15**).



Gambar 4.1 Menu utama aplikasi enkripsi SMS.

1. Menu *Generate Keys* merupakan menu yang menampilkan pembuatan kunci publik dan kunci pribadi. Didalam menu *Generate Keys* terdapat tombol *Generate* yang berfungsi untuk menghasilkan kunci publik dan kunci pribadi kemudian di tampilkan pada *textview Public Key* dan *Private Key.* Tombol *Save* dalam menu *Generate Keys* berfungsi untuk menyimpan pasangan kunci tersebut. Dan tombol *Back* berfungsi untuk kembali ke menu utama.

```
Gen.setOnClickListener(new View.OnClickListener() {
                    public void onClick(View v) {
                           KeyGeneration kg = new KeyGeneration();
                           BigInteger modulus = kg.getMod();
                           BigInteger publickey = kg.getPub();
                           BigInteger privatekey = kg.getPri();

                           Mod.setText(modulus.toString());
                           Pub.setText(publickey.toString());
                           Pri.setText(privatekey.toString());
                    }
             });
```

Gambar 4.2 *Source code* tombol *Generate Key*

```
Sve = (Button) findViewById(R.id.btnSave);
Sve.setOnClickListener(new OnClickListener() {
public void onClick(View v) {
       try {
       File myFile = new File("/sdcard/MyPublicKey.txt");
       myFile.createNewFile();
       FileOutputStream fOut = new FileOutputStream(myFile);
       OutputStreamWriter myOutWriter =
                           new OutputStreamWriter(fOut);
       myOutWriter.append(Pub.getText()+","+Mod.getText());
                           myOutWriter.close();
       fOut.close();
       Toast.makeText(getBaseContext(), "Public key saved",
              Toast.LENGTH_SHORT).show();
       } catch (Exception e) {
       Toast.makeText(getBaseContext(), e.getMessage(),
                                  Toast.LENGTH_SHORT).show();
       }
       try {
       File myFile = new File("/sdcard/MyPrivateKey.txt");
                           myFile.createNewFile();
       FileOutputStream fOut = new FileOutputStream(myFile);
       OutputStreamWriter myOutWriter = new
OutputStreamWriter(fOut);
       myOutWriter.append(Pri.getText()+","+Mod.getText());
       myOutWriter.close();
       fOut.close();
Toast.makeText(getBaseContext(), "Private key saved",
                           Toast.LENGTH_SHORT).show();
       } catch (Exception e) {
       Toast.makeText(getBaseContext(), e.getMessage(),
                           Toast.LENGTH_SHORT).show();
                    }
             }// onClick
             }); // btnWriteSDFile
```

Gambar 4.3 *Source code* tombol *Save*

```
Bck = (Button) findViewById(R.id.btnBack);
Bck.setOnClickListener(new View.OnClickListener() {
@Override
       public void onClick(View v) {
       // TODO Auto-generated method stub
       Intent i = new Intent(SetKey.this, MainActivity.class);
                 startActivity(i);
                 finish();
                       }
             });
```

Gambar 4.4 *Source code* tombol *Back*

2. Menu *Sent My Public Key* merupakan menu yang menampilkan kunci publik yang sudah di *generate*. Dalam menu ini terdapat *text field Enter Phone Number* yang berfungsi untuk memasukkan nomor telepon yang akan dituju dan juga terdapat tombol berupa icon contact untuk mencari nomor pada buku kontak. Kemudian *text field My Public Key* berfungsi untuk menampilkan kunci publik. Tombol *Sent My Key* berfungsi untuk mengirim kunci publik yang tersedia.

```
btnSendSMS.setOnClickListener(new View.OnClickListener()
        {
             public void onClick(View v)
            {
                String phoneNo = txtPhoneNo.getText().toString();
                String message = txtMessage.getText().toString();
                if (phoneNo.length()>0 && message.length()>0)
                    sendSMS(phoneNo, message);
                else
                    Toast.makeText(getBaseContext(),
                        "Please enter both phone number and
message.",
                        Toast.LENGTH_SHORT).show();
            }
        });
```

Gambar 4.5 *Source code* tombol *Sent My Key*

```
btnContact.setOnClickListener(new View.OnClickListener() {
@Override
       public void onClick(View v) {
       // TODO Auto-generated method stub
       Intent intent = new Intent(Intent.ACTION_PICK,
ContactsContract.Contacts.CONTENT_URI);
    startActivityForResult(intent, RQS_PICK_CONTACT);
                    }
             });
```

Gambar 4.6 *Source code* tombol *icon contact*

3. Menu *Compose* merupakan menu yang menampilkan fasilitas untuk pengiriman pesan. Di dalam menu *Compose* terdapat *text field Enter Phone Number* yang berfungsi untuk menginputkan nomor telepon yang dituju dan juga terdapat tombol berupa icon contact untuk mencari nomor pada buku kontak. Kemudian *text view Public Key* berfungsi untuk menampilkan kunci publik dari pengguna yang akan dikirimi pesan. *Text field Message* berfungsi untuk menginputkan pesa yang akan dikirim. Dalam menu *Compose* juga terdapat dua tombol yaitu tombol *Encrypt* yang berfungsi untuk mengenkripsi pesan yang ada di *text field Message* dan tombol *Sent SMS* berfungsi untuk mengirimkan pesan yang sudah terenkripsi.

```
btnContact.setOnClickListener(new View.OnClickListener() {
@Override
       public void onClick(View v) {
       // TODO Auto-generated method stub

       Intent intent = new Intent(Intent.ACTION_PICK,
ContactsContract.Contacts.CONTENT_URI);
    startActivityForResult(intent, RQS_PICK_CONTACT);
                    }
             });
```

Gambar 4.7 *Source code* tombol *icon contact*

```
btnEncrypt.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
// TODO Auto-generated method stub
       String message = txtMessage.getText().toString();
       KeyGeneration keyg = new KeyGeneration();
       try {
              File myFile = new File("/sdcard/buddies key.txt");
              FileInputStream fIn = new FileInputStream(myFile);
              BufferedReader myReader = new BufferedReader(
                                        new InputStreamReader(fIn));
              String aDataRow = "";
              while ((aDataRow = myReader.readLine()) != null) {
                     String[] strArr = aDataRow.split(",");
                     String pub = strArr[0];
                     String mod = strArr[1];
                     BigInteger e = new BigInteger(pub);
                     BigInteger n = new BigInteger(mod);
              BigInteger msg = new BigInteger(message.getBytes());
              BigInteger c = msg.modPow( e, n );
              String chipper = c.toString();
              txtMessage.setText(chipper);
              }
              myReader.close();
       } catch (Exception e) {}
}
});
```

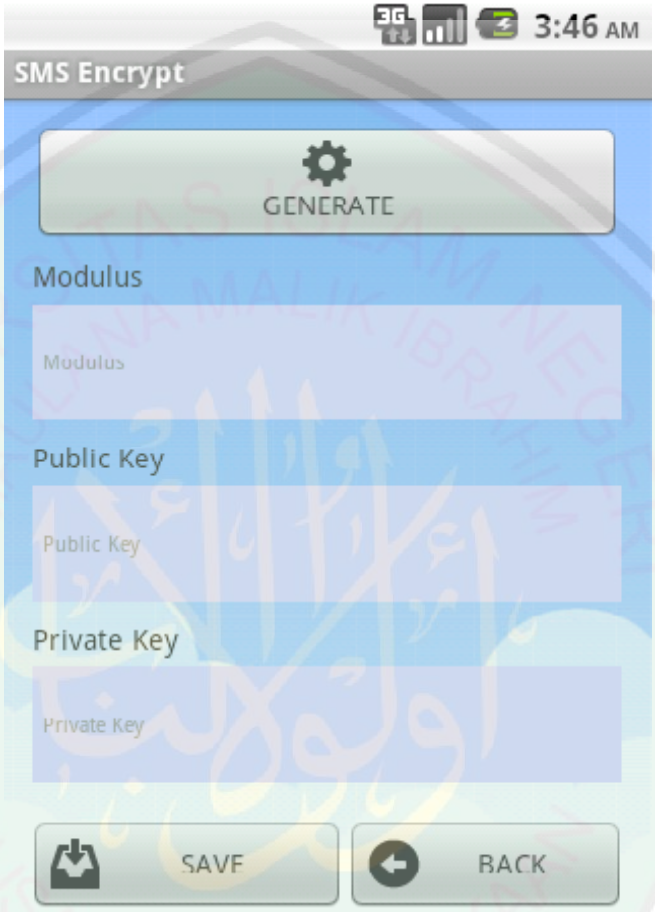
Gambar 4.8 *Source code* tombol *Encrypt*

```
btnSendSMS.setOnClickListener(new View.OnClickListener() {
public void onClick(View v) {
      String phoneNo = txtPhoneNo.getText().toString();
      String message = txtMessage.getText().toString();
      if (phoneNo.length()>0 && message.length()>0)
           sendSMS(phoneNo, message);
      else
           Toast.makeText(getBaseContext(),
                 "Please enter both phone number and message.",
                 Toast.LENGTH_SHORT).show();
            }
        });
```

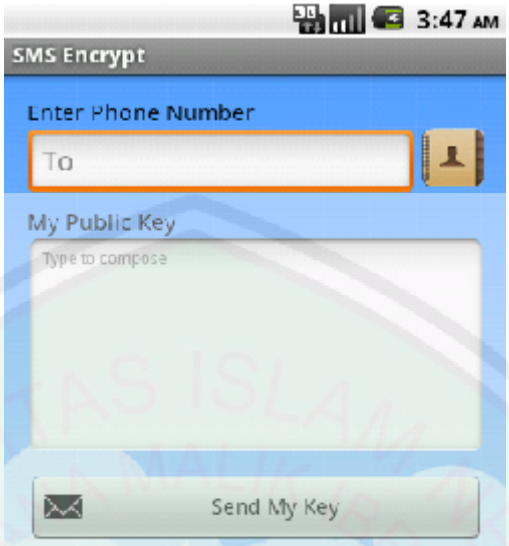
Gambar 4.9 *Source code* tombol *Send SMS*

4. Menu *Inbox merupakan* menu yang menampilkan semua pesan yang diterima oleh aplikasi.
5. Menu *Help* merupakan menu yang menampilkan cara menggunakan aplikasi enkripsi SMS.

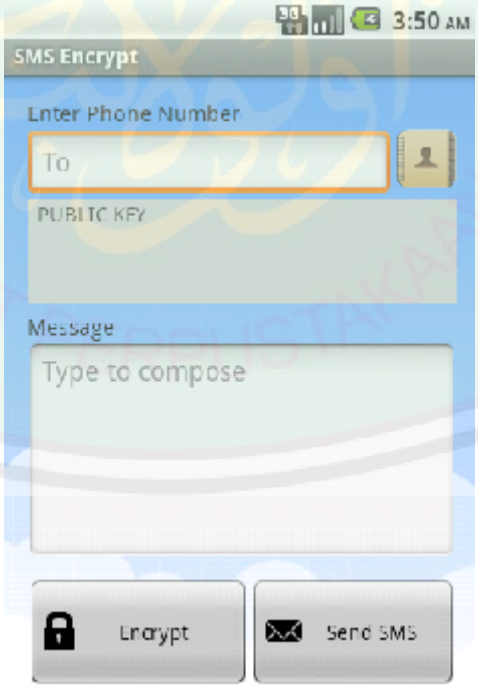
6. Menu *about* merupakan menu yang menampilan tentang bagaimana aplikasi enkripsi SMS dengan menggunakan metode RSA ini dibuat.



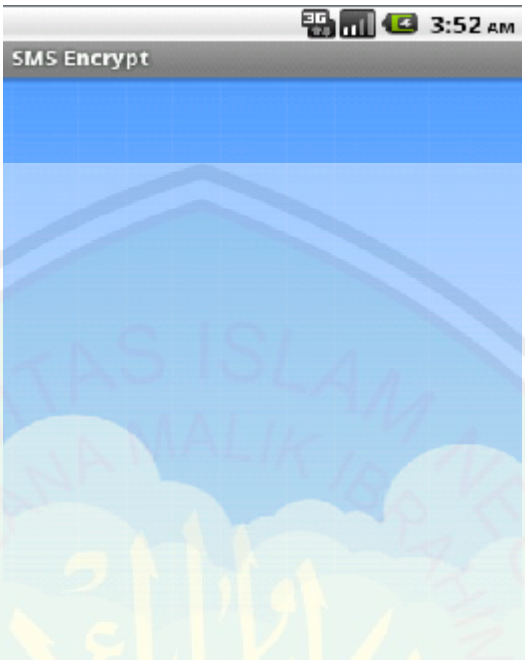
Gambar 4.10 Menu *Generate Keys*



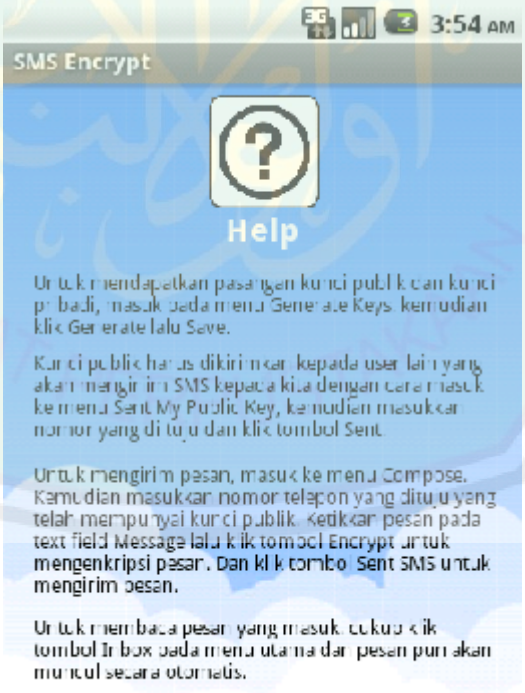
Gambar 4.11 Menu *Sent My Public Key*



Gambar 4.12 Menu *Compose*



Gambar 4.13 Menu *Inbox*



Gambar 4.14 Menu *Help*

Gambar 4.15 Menu *About*

4.1.2 Implementasi Algoritma RSA

Implementasi algoritma RSA merupakan implementasi dari perancangan sistem yang telah dibuat pada Bab 3.

4.1.2.1 Implementasi Proses Pembangkitan Kunci

Proses pembangkitan kunci pada algoritma RSA ini menggunakan panjang kunci 210 bit. Dan untuk mempermudah *user* dalam penggunaan aplikasi ini, bilangan prima yang di pakai diambil secara acak. Secara teori, setiap *user* harus (secara privat) memilih dua buah bilangan bulat acak p dan q untuk membuat kunci enkripsi dan kunci dekripsinya sendiri. Nomor-nomor ini haruslah sangat besar sehingga tidak mudah untuk memfaktorkan n = p.q

Direkomendasikan p dan q memiliki panjang 100 digit sehingga n akan memiliki panjang 200 digit. Untuk Menemukan bilangan prima sepanjang 100 digit, bangkitkanlah suatu angka 100 digit sampai satu angka prima ditemukan. Berdasarkan teorema bilangan prima, sekitar $(\ln 10^{100})/2 = 115$ angka dites sebelum kita mendapatkan angka prima.

Namun dalam prakteknya, aplikasi ini hanya bisa mendapatkan bilangan prima sepanjang 64 digit. Hal ini dikarenakan pengiriman kunci publik dilakukan melalui SMS yang mana dalam 1 kali pengiriman SMS memliliki batasan karakter sebanyak 160 karakter. Dengan bilangan prima sepanjang 64 digit, maka akan menghasilkan *n* sepanjang 127 digit. Kemudian setelah melalui proses perhitungan ditemukan *e* sepanjang 32 digit. Sedangkan kunci publik yang dikirim terdiri dari (e, *n*) yang berarti kunci publik yang akan dikirim berjumlah 160 karakter.

Source code program proses pembangkitan kunci ini ada pada *class KeyGeneration.java* bisa dilihat pada **Gambar 4.16.**

```
package com.enkripsisms;

import java.math.BigInteger;
import java.security.SecureRandom;
import java.util.Random;

public class KeyGeneration {

      private BigInteger n, d, e, p, q, phi;
      private int bitlength = 210;

public KeyGeneration() {
            Random r = new Random();
            p = BigInteger.probablePrime(bitlength, r);
            q = BigInteger.probablePrime(bitlength, r);
            n = p.multiply(q);
phi= p.subtract(BigInteger.ONE).multiply
      (q.subtract(BigInteger.ONE));
            e = BigInteger.probablePrime(bitlength/2, r);
while (phi.gcd(e).compareTo(BigInteger.ONE) > 0 &&
       e.compareTo(phi) < 0 ) {
                  e.add(BigInteger.ONE);
      }
                  d = e.modInverse(phi);
}

public synchronized String encrypt(String message ,
      BigInteger E) {
return(new BigInteger(message.getBytes())).modPow
      (E, n).toString();
        }
public synchronized BigInteger encrypt(BigInteger message ,
BigInteger E , BigInteger N  ) {
          return message.modPow(E, N);}
        }

        /** Decrypt the given ciphertext message. */
        public synchronized String decrypt(String message ,
BigInteger D) {
          return                new                String((new
BigInteger(message)).modPow(D, n).toByteArray());
        }

        /** Decrypt the given ciphertext message. */
        public   synchronized   BigInteger   decrypt(BigInteger
message , BigInteger D , BigInteger N) {
          return message.modPow(D, N);
        }
```

```
public synchronized String decrypt(String message ,
      BigInteger D) {
return new String((new BigInteger(message)).modPow
      (D, n).toByteArray());
        }
public synchronized BigInteger decrypt(BigInteger message ,
BigInteger D , BigInteger N) {
          return message.modPow(D, N);
        }

        /** Return the modulus. */
        public synchronized BigInteger getMod() {
          return n;
        }

        /** Return the public key. */
        public synchronized BigInteger getPub() {
          return e;
        }

        public synchronized BigInteger getPri() {
            return d;
              }
}
```

Gambar 4.16 *Key Generation.java*

4.1.2.2 Implementasi Proses Enkripsi SMS

Proses enkripsi sms pada aplikasi ini terletak pada *class Sent.java*. *Class* ini berfungsi untuk mengirim SMS sekaligus mengenkripsi pesan yang akan dikirim. Pesan yang akan dienkripsi menunggu perintah dari tombol *Encrypt*. Setelah tombol disentuh maka pesan akan dienkripsi dengan menggunakan kunci publik dari nomor yang akan dituju. Apabila nomor tersebut tidak mempunyai kunci publik maka pesan tidak bisa dieksekusi atau di enkripsi. Untuk mempermudah proses enkripsi, dalam aplikasi ini kunci publik yang akan digunakan untuk proses enkripsi akan otomatis tersimpan dalam memori. Kemudian ketika sampai proses perhitungan enkripsi, kunci publik yang terdiri

dari (e, n) akan dipanggil kembali untuk diimplementasikan dalam rumus

$$C_i = P_i^e \bmod n.$$

Dimana, C : Chipertext

P : Plaintext

e : kunci publik

n : modulus yang digunakan.

Source code programnya dapat dilihat pada **Gambar 4.17** dan **Gambar 4.18.**

```
btnEncrypt.setOnClickListener(new View.OnClickListener() {
      @Override
public void onClick(View v) {
// TODO Auto-generated method stub
String message = txtMessage.getText().toString();
KeyGeneration keyg = new KeyGeneration();
      try {
      File myFile = new File("/sdcard/buddies key.txt");
      FileInputStream fIn = new FileInputStream(myFile);
      BufferedReader myReader = new BufferedReader(
      new InputStreamReader(fIn));
      String aDataRow = "";
while ((aDataRow = myReader.readLine()) != null) {
             String[] strArr = aDataRow.split(",");
             String pub = strArr[0];
              String mod = strArr[1];

              BigInteger e = new BigInteger(pub);
              BigInteger n = new BigInteger(mod);
             BigInteger msg = new BigInteger(message.getBytes());
            BigInteger c = msg.modPow( e, n );
            String chipper = c.toString();
            txtMessage.setText(chipper);

            }
            myReader.close();
      } catch (Exception e) {}
}
});
```

Gambar 4.17 *Source code* proses enkripsi SMS

```
btnSendSMS.setOnClickListener(new View.OnClickListener()
  {
      public void onClick(View v)
      {
      String phoneNo = txtPhoneNo.getText().toString();
      String message = txtMessage.getText().toString();
      if (phoneNo.length()>0 && message.length()>0)
          sendSMS(phoneNo, message);
      else
      Toast.makeText(getBaseContext(),
      "Please enter both phone number and message.",
      Toast.LENGTH_SHORT).show();

            }

        });
    }

private void sendSMS(String phoneNumber, String message)
    {
        String SENT = "SMS_SENT";
        String DELIVERED = "SMS_DELIVERED";
        PendingIntent sentPI = PendingIntent.getBroadcast
                        (this, 0, new Intent(SENT), 0);

        PendingIntent deliveredPI = PendingIntent.getBroadcast
                        (this, 0, new Intent(DELIVERED), 0);

        //---when the SMS has been sent---
registerReceiver(new BroadcastReceiver(){
       @Override
public void onReceive(Context arg0, Intent arg1) {
                switch (getResultCode())
                {
      case Activity.RESULT_OK:
      Toast.makeText(getBaseContext(), "SMS sent",
      Toast.LENGTH_SHORT).show();
      break;
      case SmsManager.RESULT_ERROR_GENERIC_FAILURE:
      Toast.makeText(getBaseContext(), "Generic failure",
      Toast.LENGTH_SHORT).show();
      break;
      case SmsManager.RESULT_ERROR_NO_SERVICE:
      Toast.makeText(getBaseContext(), "No service",
      Toast.LENGTH_SHORT).show();
      break;
      case SmsManager.RESULT_ERROR_NULL_PDU:
                        Toast.makeText(getBaseContext(),  "Null
PDU",
                                Toast.LENGTH_SHORT).show();
                        break;
                    case SmsManager.RESULT_ERROR_RADIO_OFF:
                        Toast.makeText(getBaseContext(),
"Radio off",
                                Toast.LENGTH_SHORT).show();
                        break;
```

```
            }
        }
    }, new IntentFilter(SENT));

//---when the SMS has been delivered---
        registerReceiver(new BroadcastReceiver(){
            @Override
            public void onReceive(Context arg0,Intent arg1) {
                switch (getResultCode())
                {
                    case Activity.RESULT_OK:
                        Toast.makeText(getBaseContext(), "SMS
delivered",
                                Toast.LENGTH_SHORT).show();
                        break;
                    case Activity.RESULT_CANCELED:
                        Toast.makeText(getBaseContext(), "SMS not
delivered",
                                Toast.LENGTH_SHORT).show();
                        break;
                }
            }
        }, new IntentFilter(DELIVERED));
        SmsManager sms = SmsManager.getDefault();
        sms.sendTextMessage(phoneNumber, null, message, sentPI,
deliveredPI);
    }
}
```

Gambar 4.18 *Source code* proses kirim SMS

4.1.2.3 Implementasi Proses Dekripsi SMS

Proses dekripsi sms pada aplikasi ini terletak pada *class Inbox.java*. *Class* ini berfungsi untuk membaca SMS yang masuk kemudian dilakukan proses dekripsi. Ketika ada pesan masuk, aplikasi ini secara otomatis membaca pesan tersebut yang kemudian dilakukan proses dekripsi. Apabila pesan yang masuk merupakan *chipertext* atau bukan pesan yang terenkripsi maka aplikasi akan menampilkannya sebagai simbol-simbol. Karena aplikasi ini tidak bisa memilah antara *chipertext* dan *plaintext,* jadi dalam inbox akan membaca semua pesan

yang masuk dan dihitung dengan kunci privat yang ada dengan rumus $P_i = C_i^d \bmod n$.

Dimana, C : Chipertext

P : Plaintext

d : kunci privat

n : modulus yang digunakan

. Namun apabila pesan yang masuk berupa *chipertext* maka dalam inbox akan ditampilkan kembali dalam bentuk *plaintext* setelah melalui proses seperti diatas. Source code programnya dapat dilihat pada **Gambar 4.19.**

```
try {
       File myFile = new File("/sdcard/MyPrivateKey.txt");
       FileInputStream fIn = new FileInputStream(myFile);
       BufferedReader myReader = new BufferedReader(
       new InputStreamReader(fIn));
       String aDataRow = "";
       while ((aDataRow = myReader.readLine()) != null) {
        String[] strArr = aDataRow.split(",");
             String pri = strArr[0];
             String mod = strArr[1];

        BigInteger d = new BigInteger(pri);
        BigInteger n = new BigInteger(mod);
       BigInteger chp = new BigInteger(msg);
       BigInteger dec = chp.modPow( d, n );
       String chipper = new String(dec.toByteArray());
       Inbox_msg[pos]=chipper;
                    }
       myReader.close();
} catch (Exception e) {}
```

Gambar 4.19 *Source code* proses dekripsi SMS

4.2 Uji Coba Sistem

Dalam uji coba ini, penulis ingin mengetahui hasil dari implementasi sistem mulai dari pembangkitan kunci, enkripsi SMS sampai dekripsi SMS. Dalam

pembangkitan kunci ini penulis menentukan beberapa ukuran kunci untuk menyesuaikan panjang kunci publik dengan batas panjang karakter dalam SMS yaitu 160 karakter.

Tabel 4.1 Ukuran kunci publik dan kunci pribadi.

Ukuran kunci	Kunci publik	Kunci privat
200	130	229
210	**159**	**253**
215	169	271
256	193	308

Setelah melakukan percobaan dalam menentukan panjang kunci yang akan digunakan, dari tabel diatas di ketahui panjang kunci publik yang mendekati batas maksimal karakter dalam SMS yaitu ukuran kunci 210. Maka panjang kunci yang di pilih untuk pembangkitan kunci RSA adalah 210 bit.

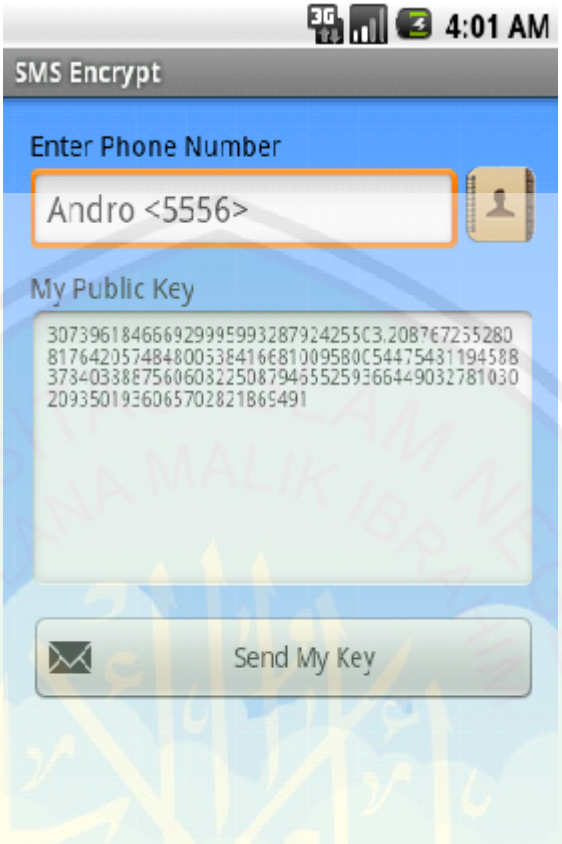
4.2.1 Proses Uji Coba Pembangkitan Kunci

Proses uji coba ini dilakukan untuk mengetahui kunci publik dan privat yang dihasilkan dari proses pembangkitan kunci dengan bilangan prima acak dan modulo yang telah di tentukan dalam sistem. Proses ini dapat di lihat pada **Gambar 4.20** dan **Gambar 4.21.**

Gambar 4.20 Proses pembangkitan kunci publik dan kunci privat

4.2.2 Proses Uji Coba Enkripsi SMS

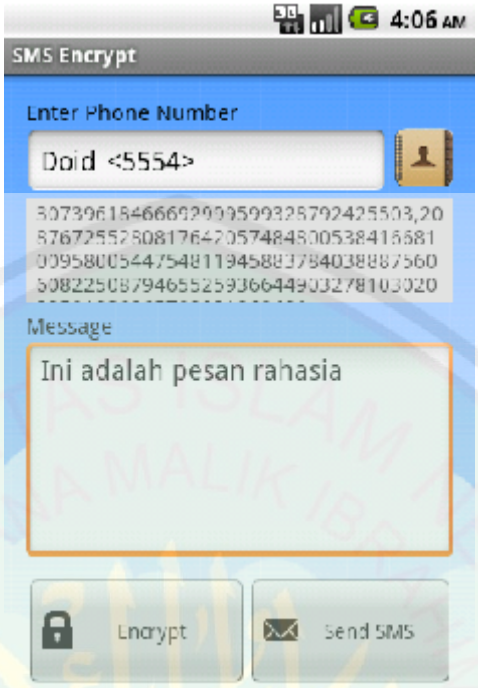
Proses uji coba ini dilakukan untuk mengetahui apakah pesan dapat dienkripsi atau tidak, serta apakah pesan yang sudah di enkripsi bisa di dekripsi atau tidak. Proses ini dapat di lihat pada **Gambar 4.22** dan **Gambar 4.23**.



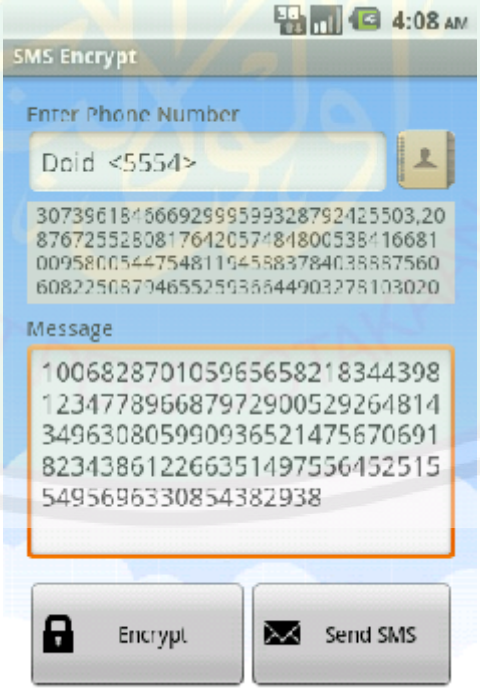
Gambar 4.21. Tampilan kirim kunci publik

4.2.3 Proses Uji Coba Dekripsi SMS

Proses uji coba ini dilakukan untuk mengetahui apakah pesan yang telah di enkripsi dapat didekripsi atau tidak. Proses ini dapat dilihat pada **Gambar 4.24** dan **Gambar 4.25.**



Gambar 4.22 Tampilan pesan sebelum di enkripsi



Gambar 4.23 Tampilan pesan setelah di enkripsi

Gambar 4.24 Pesan masuk pada aplikasi SMS biasa.

Gambar 4.25 Pesan masuk pada aplikasi enkripsi SMS.

4.2.4 Proses Uji Coba Enkripsi Pesan Dengan Karakter yang Berbeda-beda

Pada uji coba ini aplikasi akan di coba untuk mengirim pesan dengan karakter yang berbeda-beda. Berdasarkan uji coba yang telah dilakukan, maka hasil pengujian dapat dituliskan pada tabel berikut:

Tabel 4.2 Hasil percobaan pengiriman SMS dengan karakter yang berbeda-beda.

No	Pesan	Hasil Enkripsi	Keterangan
1	ini password web qt yg baru: Qwar db7war9	1346682662271816750583887521866827119199184342198345D52D8740856381072010060263509679928677D8338D964583441640D6491979809787001 22	41 karakter. Berhasil dikirim dan berhasil didekripsi.
2	besok aq ketemprmu ya. jgn pergi dulu sblum aq datang	765119691406000384450394408915735849326DD1425767839689461836983745985840460764721651267316681461292 7D144178756409D0290548897660	53 karakter. Berhasil dikirim tapi gagal didekripsi.
3	oh iyar coba pake akun yg baru ya. soalnya ngecek.	885764049695494388549575 7D8968190746426374996521820043230287863125441338614235523299841014356224820002308362243917506936279	50 karakter. Berhasil dikirim dan berhasil didekripsi dengan benar.
4	besok gak usah kesini. tadi barangnya udah diantar sama mesmu.	1577658982345310438738076608247751431986632964474494497D6866682182926649625574222730829670127B6B1294D0434D83312B1D1761674223542	61 karakter. Berhasil dikirim tapi gagal didekripsi.
5	besok pagi aja ya.	3329288225837299345280878485427144911163593314260382225139480607176563718578775710771087733787297638778873299816772833466722739	18 karakter. Berhasil dikirim dan berhasil di dekripsi dengan benar.
6	nanti malem jemput jangn lupa jmput diterminal ya. ni skrang prsisi masih di solo.	2004112681347666293665243207696686976031666676520910732224268672173600429216249006507322098586673649855870560933298609980741691	83 karakter. Berhasil dikirim tapi gagal didekripsi.

Dari hasil percobaan di atas, maka dapat dilihat bahwa pada aplikasi ini bisa mengenkripsi pesan yang akan di kirim. Pesan yang dikirim juga sampai ke tempat tujuan dengan utuh. Namun pada aplikasi ini tidak bisa mendekripsikan pesan yang memiliki karakter lebih dari 50 karakter.

4.2.5 Proses Uji Coba Terhadap Berbagai Tipe *Smartphone*

Pada uji coba ini aplikasi akan di instal ke beberapa macam merk *smartphone.* Selanjutnya akan di lihat apakah aplikasi dapat berjalan dengan baik atau tidak. Proses uji coba dilakukan terhadap *smartphone* yang berbasis android. Berdasarkan uji coba yang telah dilakukan, maka hasil pengujian dapat dituliskan pada tabel berikut:

Tabel 4.3 Hasil percobaan terhadap berbagai tipe *smartphone*

No.	Jenis Smartphone	Keterangan	Catatan
1	Sony Ericsson Xperia X8	Berjalan dengan baik	Semua fitur berjalan dengan baik seperti pada emulator
2	Sony Ericsson Xperia Mini	Berjalan dengan baik	Semua fitur berjalan dengan baik seperti pada emulator
3	Sony Ericsson Xperia Arc	Berjalan kurang baik	Tampilan awal pada aplikasi tidak bisa full layar. Namun semua fungsi berjalan dengan baik.
4	Sony Ericsson W8	Berjalan dengan baik	Semua fitur berjalan dengan baik seperti pada emulator
5	Sony Xperia Sola	Berjalan dengan baik	Semua fitur berjalan dengan baik seperti pada emulator
6	Sony Typo Dual	Berjalan dengan baik	Semua fitur berjalan dengan baik seperti pada emulator
7	Samsung Galaxy Ace	Berjalan dengan baik	Semua fitur berjalan dengan baik seperti pada

			emulator
8	Samsung Galaxy Mini	Berjalan dengan baik	Semua fitur berjalan dengan baik seperti pada emulator
9	Samsung Galaxy Young	Berjalan dengan baik	Semua fitur berjalan dengan baik seperti pada emulator
10	Samsung Galaxy Duos	Berjalan dengan baik	Semua fitur berjalan dengan baik seperti pada emulator
11	Smartfren Andromax	Berjalan dengan baik	Semua fitur berjalan dengan baik seperti pada emulator
12	Smartfren Andromax i	Berjalan dengan baik	Semua fitur berjalan dengan baik seperti pada emulator
13	Mito A300	Berjalan dengan baik	Semua fitur berjalan dengan baik seperti pada emulator
14	Axio Vigo	Berjalan kurang baik	Tampilan awal pada aplikasi tidak bisa full layar. Namun semua fungsi berjalan dengan baik.

Dari percobaan diatas, maka dapat dilihat bahwa aplikasi dapat berjalan dengan baik pada hampir semua *smartphone.* Namun pada *smartphone* Axioo vigo dan Xperia arc ada masalah dengan tampilan yang tidak bisa full layar, dimana kedua *smartphone* tersebut memiliki dimensi lebih dari 4.0 inch. Dengan demikian aplikasi ini dapat berjalan dan berfungsi dengan hampir pada semua *smartphone.*

4.3 Integrasi Enkripsi SMS dan Islam

Islam memandang bahwa agama adalah dasar dan pengatur kehidupan. Aqidah Islam menjadi basis dari segala ilmu pengetahuan. Aqidah Islam yang

terwujud dalam apa-apa yang ada dalam Al-Qur`an dan Al-Hadits menjadi qaidah fikriyah (landasan pemikiran), yaitu suatu asas yang di atasnya dibangun seluruh bangunan pemikiran dan ilmu pengetahuan manusia.

Islam memerintahkan manusia untuk membangun segala pemikirannya berdasarkan aqidah Islam, bukan lepas dari aqidah itu. Ini bisa kita pahami dari ayat yang pertama kali turun :

ٱقۡرَأۡ بِٱسۡمِ رَبِّكَ ٱلَّذِي خَلَقَ.

Artinya: "*Bacalah dengan (menyebut) nama Tuhanmu Yang menciptakan*".(QS. Al–Alaq: 1).

Ayat ini berarti manusia telah diperintahkan untuk membaca guna memperoleh berbagai pemikiran dan pemahaman. Tetapi segala pemikirannya itu tidak boleh lepas dari Aqidah Islam, karena iqra` haruslah dengan bismi rabbika, yaitu tetap berdasarkan iman kepada Allah, yang merupakan asas Aqidah Islam.

Peran Islam sendiri dalam perkembangan sains dan teknologi, adalah bahwa Syariah Islam harus dijadikan standar pemanfaatan sains dan teknologi. Ketentuan halal-haram (hukum-hukum syariah Islam) wajib dijadikan tolok ukur dalam pemanfaatan iptek, bagaimana pun juga bentuknya. Iptek yang boleh dimanfaatkan, adalah yang telah dihalalkan oleh syariah Islam. Sedangkan sains dan teknologi yang tidak boleh dimanfaatkan, adalah yang telah diharamkan syariah Islam. Jika peran ini dapat dimainkan oleh umat Islam dengan baik, insyaallah akan ada berbagai berkah dari Allah kepada umat Islam dan juga seluruh umat manusia.

Dalam hal ini aplikasi enkripsi SMS diharapkan dapat memberikan manfaat bagi penggunanya dan tidak disalahgunakan agar tidak menyimpang dari syariah Islam sesuai dengan firman Allah SWT:

إِنَّ فِي خَلْقِ ٱلسَّمَٰوَٰتِ وَٱلْأَرْضِ وَٱخْتِلَٰفِ ٱلَّيْلِ وَٱلنَّهَارِ لَءَايَٰتٍ لِّأُو۟لِى ٱلْأَلْبَٰبِ
ٱلَّذِينَ يَذْكُرُونَ ٱللَّهَ قِيَٰمًا وَقُعُودًا وَعَلَىٰ جُنُوبِهِمْ وَيَتَفَكَّرُونَ فِي خَلْقِ ٱلسَّمَٰوَٰتِ
وَٱلْأَرْضِ رَبَّنَا مَا خَلَقْتَ هَٰذَا بَٰطِلًا سُبْحَٰنَكَ فَقِنَا عَذَابَ ٱلنَّارِ

Artinya: "*Sesungguhnya dalam penciptaan langit dan bumi serta silih bergantinya malam dan siang, terdapat tanda-tanda (Kebesaran Allah) bagi kalangan ulul albab. Yaitu mereka yang hatinya selalu bersama Allah di waktu berdiri, duduk dan dalam keadaan berbaring dan memikirkan tentang penciptaan langit dan bumi (seraya berkata), Ya Tuhan kami,tidaklah Engkau menciptakan ini semua dengan sia-sia, Maha Suci Engkau, maka perliharalah kami dari azab neraka*." (QS Al Imron 190-191)

Dari ayat ini dapat kita lihat, bahwa melalui pengamatan, kajian dan pengembangan sains dan teknologi, Allah menghendaki manusia dapat lebih merasakan kebesaran, kehebatan dan keagunganNya. Betapa hebatnya alam ciptaan Allah, yang kebesaran dan keluasannyapun manusia belum sepenuhnya mengetahui, maka sudah tentu Maha hebat lagi Allah yang menciptakannya. Tidak terbayangkan oleh akal fikiran dan perasaan manusia Maha Hebatnya Allah. Kalaulah alam semesta yang nampak secara lahiriah saja sudah begitu luas, menurut kajian dengan menggunakan peralatan terkini yang canggih diameternya 20 milyar tahun cahaya, terasa betapa besar dan agungnya Allah yang menciptakannya. Ini alam lahiriah yang nampak dan dapat diukur secara lahiriah, belum lagi alam-alam yang berbagai jenis yang tidak dapat dikaji dan diobservasi dengan peralatan lahiriah buatan manusia, walau secanggih apapun.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah melakukan analisa, merancang dan mengimplementasikan sistem pada aplikasi enkripsi SMS pada *Smartphone,* penelitian ini telah menjawab bagaimana melakukan desain sistem, desain proses dan desain antarmuka agar aplikasi enkripsi SMS ini dapat diimplementasikan secara nyata sesuai dengan metodologi yang diuraikan diatas. Berdasarkan implementasi dan uji coba sistem yang telah dilakukan aplikasi bisa mengenkripsi pesan yang akan dikirim ke pengguna lain dengan benar. Dan juga aplikasi enkripsi SMS ini dapat mendekripsikan pesan yang terenkripsi sesuai dengan kunci publik dan kunci privat yang telah dihasilkan sebelumnya.

Namun masih ada banyak kekurangan pada aplikasi enkripsi SMS ini. Diantaranya yaitu aplikasi ini tidak bisa mendekripsikan pesan yang memiliki karakter lebih dari 50 karakter. Dan juga aplikasi ini tidak dapat menampilkan fitur-fiturnya dengan rapi apabila diinstal pada *smartphone* yang memiliki dimensi lebih dari 4.0 inch.

5.2 Saran

Dalam aplikasi enkripsi SMS ini perlu ditambahnya fitur lain agar lebih menarik dalam penggunaannya, seperti *pop-up inbox* atau fitur yang lebih interaktif. Aplikasi enkripsi SMS ini menggunakan panjang kunci 210 bit.

Sedangkan untuk memenuhi tingkat keamanan yang lebih tinggi diharapkan dalam penelitian selanjutnya aplikasi ini dapat dibuat dengan panjang kunci lebih dari 1024.

DAFTAR PUSTAKA

Android makes the move to power architectur technlogy. https://community.freescale.com/community/the-embedded-beat/blog/2010/05/24/android-makes-the-move-to-power-architecture-technology (diakses pada tanggal 10 maret 2013).

Andy Wicaksono, Prasetyo. 2009. *Enkripsi Menggunakan Algoritma RSA*. Bandung: Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.

Aribowo, Eko. *Aplikasi Pengamanan Dokumen Office Dengan Algoritma Kriptografi Kriptografi Kunci Asimeteris ElGamal*. Juli 2008.

Hulisani R, Johnny L, Judith B. 2004. *A Security Mechanism for Secure SMS Communication.* South Africa: University of Pretoria.

Ivan W, Budi S, Junius K. *Penerapan Algoritma Kriptografi Asimetris RSA Untuk Keamanan Data di Oracle.* April 2009.

Jalaluddin Asy-Syuyuthi, Jalaluddin Muhammad Ibn Ahmad Al-Mahalliy. 2010. *Tafsir Jalalain.* Tasikmalaya: Pesantren Persatuan Islam 91.

Jones, Nick. 2002. *Don't Use SMS for Confidential Communication*. Gartner Research. http://www.gartner.com/id=379178 (diakses pada tanggal 10 maret 2013).

Riyanto, M. Zaki., & Ardhi Ardian. 2008. *Kriptografi Kunci Publik: Sandi RSA*. Yogyakarta.

Rizal, Moh. Yose. 2012. *Perancangan Simulasi Man In The Middle Attack Pada Algoritma Kriptografi Rsa Dan Pencegahannya Dengan Interlock Protocol.* Yogyakarta: Sekolah Tinggi Manajemen Informatika Dan Komputer, Amikom.

Heri. 2010. *Mekanisme Pengiriman SMS dari HP.* http://ciptamedia-sms-broadcast.blogspot.com/2010/06/mekanisme-pengiriman-sms-dari-hp.html Diakses tanggal 12 Januari 2013.

Safaat H, Nazruddin. 2012. *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android.* Bandung: Informatika.

The Government of the Hong Kong Special Administrative Region. 2008. *Short Message Service Security*. Hongkong.