IMPLEMENTASI ALGORITMA RSA UNTUK ENKRIPSI DAN DEKRIPSI SMS (*SHORT MESSAGE SERVICE*) PADA PONSEL BERBASIS ANDROID

(*IMPLEMENTATION OF RSA ALGORITHM FOR ENCRYPTION AND DECRYPTION OF SMS (SHORT MESSAGE SERVICE) BASED ON ANDROID PHONE*)

Erick Ruliyanto Sardju[1], Ir.Rita Magdalena,M.T[2], Ratri Dwi Atmaja,S.T.,M.T[3]
[1,2]Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom
[2] Prodi D3 Teknik Telekomunikasi, Fakultas Ilmu Terapan, Universitas Telkom
ruliyantoerick@gmail.com, **ritamagdalena@telkomuniversity.com**, **ratridwiatmaja@yahoo.co.id**

Abstrak - Perkembangan teknologi telepon seluler saat ini sudah sangat pesat dan maju. Dengan berkembangnya teknologi ini, akhir-akhir ini sedang berkembang teknologi ponsel pintar (*smartphone*) yang membuat ponsel tidak hanya digunakan untuk menelpon atau mengirim pesan saja, kini ponsel sudah dilengkapi dengan teknologi *touch ID*, namun masih ada beberapa kekurangan dari aplikasi tersebut. Namun keamanan dari suatu ponsel dalam komunikasi belum terjamin, salah satunya komunikasi dalam bentuk pesan singkat.

Keamanan pesan pada SMS sendiri masih belum terjamin, maka penulis membuat sebuah aplikasi enkripsi dan dekripsi dengan menggunakan metode algoritma RSA untuk mengamankan informasi pada pesan singkat. Sehingga user dapat berkomunikasi pada pesan singkat tanpa perlu khawatir dari para attackers yang ingin mengetahui isi pesan yang dikirim oleh user. Aplikasi ini menggunakan eclipse yang berbasis bahasa pemograman java yang dapat dikembangkan pada smartphone yang berbasis android. Hasil keluaran dari sistem ini yaitu pada pengiriman sms yang telah terenkripsi akan terkirim apabila ≤ 160 karakter, dan sms tidak akan terkirim apabila ≥ 160 karakter, pada proses enkripsi dan dekripsi membutuhkan waktu rata-rata 0,18 detik, pada pengujian *avalanche effect* dengan menggunakan inputan plaintext yang berbeda tiap percobaan akan menghasilkan chipertext yang berbeda dengan presentase rata-rata sebesar 10,35 %, sedangkan pada pengujian *brute force* membutuhkan waktu selama 1,652 x tahun untuk mencoba semua kemungkinan kunci yang ada.

Kata kunci: Android, Algoritma RSA, Aplikasi, Pengiriman SMS

Abstract - **The development of mobile phone technology nowadays is very fast and advanced. By the development of it, now mobile phones not only could send messages or make a call, but also equipped with ID touch technology, although there are still some shortcomings of the application. The security of a mobile phone in communication is not guaranteed, as in the form of a short message service.**

SMS content on its own security is still not assured, so that the authors make an encryption and decryption application using the RSA algorithm for securing information in a short message service in purpose to make mobile phone user can communicate in short messages service without worrying of the attackers who want to know the content of the message sent by the user. This application uses eclipse based java language programming which can be developed on android based smartphone.

The output of this system is the SMS's delivery process which has been encrypted will be sent if ≤ 160 characters, and text will not be sent if ≥ 160 characters, the encryption and decryption process takes an average time of 0.18 seconds, the avalanche effect testing using different plaintext input for each experiment will produce a different ciphertext with an average percentage of 10.35%, while the brute force testing takes over 1,652× years to try all possible keys there.

Keywords : Android, RSA Algorithm, Application, SMS Delivery Process

I. PENDAHULUAN

Beberapa tahun terakhir ini terjadi perkembangan yang pesat pada teknologi, salah satunya adalah telepon seluler (ponsel). Mulai dari ponsel yang hanya bisa digunakan untuk bicara dan sms hingga "ponsel cerdas" (*smart phone)* yang memiliki berbagai fungsi seperti *multimedia*, *multiplayer games*, transfer data, video *streaming* dan lain-lain. Salah satu fasilitas yang disediakan handphone adalah untuk melakukan pengiriman data berupa pesan singkat melalui *Short Message Service* (SMS). Namun dengan fasilitas SMS yang ada, timbul pertanyaan mengenai keamanan informasi jika seseorang ingin mengirimkan suatu informasi rahasia melalui

fasilitas SMS. Implementasi Algoritma RSA untuk Enkripsi dan Dekripsi SMS (Short Message Service) pada ponsel berbasis Android dapat digunakan untuk mengirim dan menerima pesan teks sekaligus memiliki fasilitas untuk mengamankan atau menyembunyikan informasi dari pesan yang dikirimkan.

II. TEORI PENUNJANG

A. Enkripsi

Enkripsi adalah proses mengamankan data atau informasi, dengan kata lain mengacak data atau informasi agar tidak dapt dibaca oleh pihak lain[7].

B. Dekripsi

Dekripsi adalah kebalikan dari enkripsi yaitu proses mengkonversi data yang sudah di enkripsi kembali menjadi data aslinya sehingga dapat dibaca atau dimengerti kembali[7].

C. Plaintext dan Ciphertext

Plaintext adalah teks yang diencode dalam format ASCII. Plain text tidak memiliki format dan informasi struktur seperti ukuran dan tipe font, warna, atau layout. Plain text biasanya digunakan antar-komputer yang tidak memiliki kesepakatan untuk saling bertukar informasi format dan layout teks. Sedangkan Ciphertext adalah bentuk setelah pesan dalam plaintext telah diubah bentuknya menjadi lebih aman dan tidak dapat dibaca. Proses mengubah plaintext menjadi ciphertext disebut encryption (enciphering), dan proses membalikkannya kembali disebut decryption (deciphering)[3].

D. Kriptografi

Secara etimologi (ilmu asal usul kata), kata kriptografi berasal dari gabungan dua kata dalam bahasa Yunani yaitu “kriptos” dan “graphia”. Kata kriptos digunakan untuk mendeskripsikan sesuatu yang disembunyikan, rahasia atau misterius. Sedangkan kata graphia berarti tulisan. Kriptografi didefinisikan sebagai ilmu dan pelajaran untuk tulisan rahasia dengan pertimbangan bahwa komunikasi dan data dapat dikodekan untuk mencegah dari mata-mata atau orang lain yang ingin mengetahui isinya, dengan menggunakan kode-kode dan aturan-aturan tertentu dan metode lainnya sehingga hanya orang yang berhak yang dapat mengetahui isi pesan sebenarnya[3].

E. Short Message Service (SMS)

Short Message Sevice (SMS) adalah salah satu fasilitas dari teknologi GSM yang memungkinkan mengirim dan menerima pesan pesan singkat berupa text dari Mobile Station (MS). Layanan SMS juga memungkinkan pengiriman pesan dalam bentuk alphanumeric, layanan SMS ini banyak diaplikasikan pada sistem komunikasi tanpa kabel (wireless)[9].

F. Algoritma RSA

RSA adalah salah satu algoritma kriptografi asimetris yang menggunakan sepasang kunci yaitu, kunci publik dan kunci pribadi.Panjang kunci dapat diatur, dimana semakin panjang bit pembentukan kunci maka semakin sukar untuk dipecahkan karena sulitnya memfaktorkan dua bilangan yang sangat besar[10].

Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima.Pemfaktoran dilakukan untuk memperoleh kunci privat. Selama pemfaktoran bilangan besar menjadi faktor-faktor prima belum ditemukan algoritma yang efisien, maka selama itu pula keamanan algoritma RSA tetap terjamin[9].

Besaran-besaran yang digunakan pada Algoritma RSA[9]:

1. p dan q bilangan prima (rahasia)
2. $n = p \cdot q$ (tidak rahasia)
3. $\phi(n) = (p - 1)(q-1)$ (rahasia)
4. e (kunci enkripsi) (tidak rahasia)
5. d(kunci dekripsi) (rahasia)
6. m (plainteks) (rahasia)
7. c (cipherteks) (tidak rahasia)

G. Android

Gambar 2.1 *Logo Android*
Sumber:smartnotes.us

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh

bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia[14].

H. Android SDK (*Software Development Kit*)

Android SDK adalah *tools* API (Aplication Programming Interface) yang diperlukan untuk memulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Saat ini disediakan Android SDK sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java[8].

I. Eclipse

Eclipse adalah sebuah IDE (Integrated Development Environment) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (platform-independent). Eclipse dibuat menggunakan bahasa Java sehingga bersifat cross-platform. Eclipse mendukung banyak plugin tambahan yang berguna untuk mengembangkan ranah kebutuhan software development. Akan tetapi selain untuk java Eclipse juga mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya[4].

J. ADT (Android Development Tools)

Android Development Tools (ADT) adalah *plugin* untuk Eclipse yang didesain untuk pengembangan aplikasi Android. ADT memungkinkan Eclipse untuk digunakan dalam membuat aplikasi Android baru, membuat *User Interface*, menambahkan komponen berdasarkan *framework* API Android, *debug* aplikasi, dan pemaketan aplikasi Android[5].

K. Android Mobile

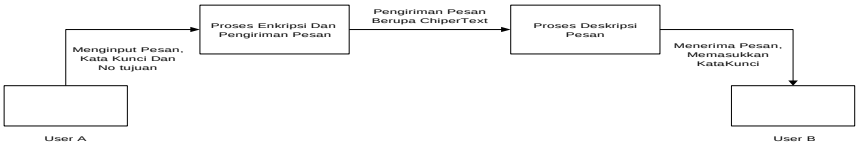
Gambar 2.2 samsung gt-s5360
Sumber:www.samsung.com

Pada tugas akhir ini penulis menggunakan samsung gt-s5360 sebagai android mobile yang telah di instal aplikasi. Samsung gt-s5360 menggunakan *operating system* Android 2.3.6 *Gingerbread* . Dengan layar capacitive touchscreen, dengan memori internal 180 MB dan RAM 290 MB dilengkapi slot micro card sampai dengan 32 GB[12].

III. PERANCANGAN SISTEM

A. Blok Diagram Sistem

Blok diagram adalah bagaimana Menggambarkan secara umum proses system ini bekerja yang dimana merepresentasikan sebuah interaksi antara User A menuju User B.

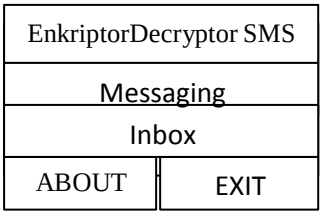


Gambar 3.1 Blok Diagram

B. Rancangan Aplikasi

1. Tampilan Awal Interface Program dan Menu

Pada awal aplikasi dijalankan menu yang ditampilkan terdiri dari 2 tombol *button* yang mewakili *Enkriptor* dan *Dekriptor* beserta menu tentang aplikasi.

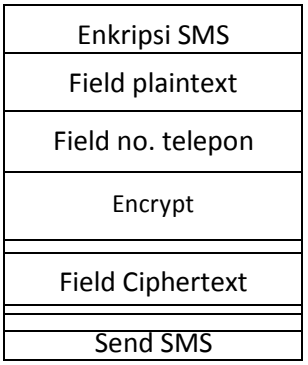


Gambar 3.2 *Interface Awal Program dan Menu*

2. Tampilan Interface Form Encryption

Tampilan form Enkryption terdiri atas 3 field text dan 2 button :

1. 3 Field text mewakili plaintext, ciphertext dan nomor tujuan
2. 2 button mewakili tombol enkripsi dan tombol pengiriman pesan

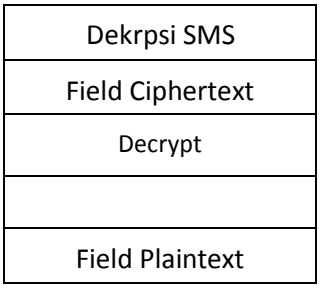


Gambar 3.3 *Interface Form Enkripsi*

3. Tampilan Interface Form Decryption

Tampilan form Dekryption terdiri atas 2 field text dan 1 button :

1. 2 Field text terdiri dari field ciphertext dan field plaintext
2. Button terdiri dari tombol Dekrypt



Gambar 3.4 *Interface Form Dekripsi*

4. Tampilan Interface Tentang Aplikasi

Tampilan interface tentang aplikasi berupa pop-up window yang akan muncul ketika menu About Aplikasi di tekan.

Gambar 3.5 *Interface TentangAplikasi*

C. Avalanche Effect

Avalanche Effect merupakan rasio antara jumlah bit-bit ciphertext yang berubah akibat perubahan plaintext ataupun kunci terhadap jumlah bit total. Perubahan bit-bit ciphertext yang kecil pada plaintext atau kunci akan menyebabkan perubahan yang signifikan terhadap ciphertext yang dihasilkan, dengan kata lain perubahan yang hanya satu bit pada plaintext ataupun kunci akan menghasilkan banyak perubahan pada bit ciphertext. Jika perubahan bit adalah setengah dari jumlah bit ciphertext maka akan sulit bagi kriptanalis untuk melakukan kriptalinisis[13].

Pada sistem ini akan dilakukan pengujian avalanche effect (perubahan plaintext) pada proses enkripsi untuk mengetahui perubahan bit-bit pada plaintext sehingga dapat diperoleh hasil persentase yang dapat menentukan baik atau tidaknya algoritma RSA yang digunakan. Berikut cara menghitung avalanche effect:

$$Avalanche\ Effect\ (AE) = \frac{\qquad\qquad\qquad\qquad}{\qquad\qquad\qquad\qquad}$$

D. Brute Force

Pada sistem ini brute force digunakan untuk mengukur berapa lama waktu yang dibutuhkan untuk mencoba-coba semua kemungkinan kunci yang ada. Sehingga kita dapat mengetahui ketahanan algoritma RSA terhadap para attackers. Cara menghitung brute force:

$$Brute\ Force = \frac{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}$$

IV. PENGUJIAN DAN ANALISIS

Pada bab ini akan membahas mengenai pengujian dan analisa dari aplikasi yang telah dibuat. Pengujian ini berupa pengujian *hardware* yang bertujuan untuk mengetahui apakah *hardware* yang dibuat sudah bekerja dengan baik dan sesuai dengan harapan.

A. Pengujian *Hardware*

Pada bagian ini dibahas mengenai pengujian *hardware* meliputi pengujian jumlah maksimal karakter yang dapat dikirim pada saat mengirim sms.

1. Tujuan Pengujian

Pengujian ini bertujuan untuk mengetahui apakah aplikasi yang digunakan dapat mengririm sms sesuai dengan standar sekali sms yaitu 160 karakter.

2. Cara Pengujian

Cara pengujiannya yaitu dengan cara mencoba mengirim sms yang telah dienkripsi pada aplikasi *Encryptor* dan *Decryptor* yang telah ada.

Gambar 4.1 Tampilan sms pada aplikasi *Encryptor* dan *Decryptor*

3. Hasil Pengujian

Dari pengujian tersebut, kita mengetahui apakah sms yang telah terenkripsi dapat terkirim sesuai dengan standar maksimal sekali sms yaitu 160 karakter.Sebagai contoh kita dapat melihat hasil pengujian dibawah ini.

Tabel 4.1 Pengiriman sms pada aplikasi *Encryptor* dan *Decryptor*

PERCOBAAN KE-	HASIL PENGUJIAN	
	JUMLAH KARAKTER	STATUS
1	20	SMS Terkirim
2	40	SMS Terkirim
3	60	SMS Terkirim
4	80	SMS Terkirim
5	100	SMS Terkirim
6	120	SMS Terkirim
7	140	SMS Terkirim
8	160	SMS Terkirim
9	180	SMS Tidak Terkirim
10	200	SMS Tidak Terkirim

Dari percobaan diatas aplikasi bekerja dengan baik sehingga dapat kita lihat bahwa sms yang telah dienkripsi akan terkirim apabila $\leq$ 160 karakter, dan sms tidak akan terkirim apabila $\geq$ 160 karakter karena sudah melebihi kapasitas batas karakter sekali sms yaitu sebanyak 160 karakter.

B. Kecepatan Pada Proses Enkripsi dan Dekripsi SMS

1. Tujuan Pengujian

Pengujian ini bertujuan untuk mengetahui berapa lama waktu yang dibutuhkan untuk melakukan enkripsi dan dekripsi SMS.

2. Cara Pengujian

Pengujian dilakukan dengan menginput pesan yang akan dikirim setelah itu melakukan enkripsi dengan menekan tombol *Encrypt* pada aplikasi. Setelah itu membuka tombol inbox dan lalu dekripsi pesan yang masuk dengan cara menekan tombol *Decrypt* pada aplikasi.

Gambar 4.2 Menginput pesan dan enkripsi pesan

Gambar 4.3 Membuka inbox dan dekripsi pesan

3. Hasil Pengujian

Tabel 4.2 Proses Enkripsi SMS

PERCOBAAN KE-	HASIL PENGUJIAN	
	JUMLAH KARAKTER	WAKTU
1	20	0,18 detik
2	40	0,18 detik
3	60	0,18 detik
4	80	0,18 detik
5	100	0,18 detik
6	120	0,18 detik
7	140	0,18 detik
8	160	0,18 detik

Dari pengujian tersebut, dengan jumlah karakter yang berbeda proses enkripsi tetap bias dilakukan dengan waktu rata-rata 0,18 detik. Sehingga dapat dikatakan bahwa aplikasi bekerja dengan baik. Dengan waktu tersebut maka user tidak membutuhkan waktu yang lama pada proses enkripsi pesan

Tabel 4.3 Proses Dekripsi SMS

PERCOBAAN KE-	HASIL PENGUJIAN	
	JUMLAH KARAKTER	WAKTU
1	20	0,18 detik
2	40	0,18 detik
3	60	0,18 detik
4	80	0,18 detik
5	100	0,18 detik
6	120	0,18 detik
7	140	0,18 detik
8	160	0,18 detik

Dari pengujian tersebut, dengan jumlah karakter yang berbeda proses dekripsi tetap bisa dilakukan dengan waktu rata-rata 0,18 detik. Sehingga dapat dikatakan bahwa aplikasi bekerja dengan baik. Dengan waktu tersebut maka user tidak membutuhkan waktu yang lama pada proses dekripsi pesan.

C. Avalanche Effect

$$\textit{Avalanche Effect (AE)} = \frac{\qquad\qquad}{\qquad\qquad}$$

Tabel 4.4 Avalanche effect

Percobaan ke-	Plaintext	Kunci publik (e)	Ciphertext	Presentase (%)
1	Buku Buka	7	32 39 68 39 32 39 68 59	12 %
2	Bus Bis	7	32 39 80 32 118 80	11,11 %
3	Kawan Kawat	7	68 59 37 59 33 68 59 37 59 129	6,06 %
4	Peta Peti	7	18 62 129 59 18 62 129 118	15,38 %
5	Makro Mikro	7	21 59 68 49 45 21 118 68 49 45	12,90 %
6	Jakarta Jakarto	7	50 59 68 59 49 129 59 50 59 68 59 49 129 45	6,67 %
7	Bandung Banding	7	32 59 33 100 39 33 38 32 59 33 100 118 33 38	6,81 %
8	Harapan Karapan	7	91 59 49 59 18 59 33 68 59 49 59 18 59 33	11,90 %

Dari pengujian tersebut, dengan inputan plaintext yang berbeda 1 karakter tiap percobaan tetapi menggunakan kunci yang sama akan menghasilkan ciphertext yang berbeda. Kemudian percobaan tersebut menghasilkan rata-rata presentase *avalanche effect* yaitu sebesar 10,35 %.

D. Brute Force

Pengujian ini untuk mengetahui kekuatan algoritma terhadap brute force, diperlukan waktu untuk mencoba semua kemungkinan kunci yang ada. Pada aplikasi ini menggunakan batas maksimal kunci public dan private yaitu 255-bit dan mempunyai waktu rata-rata enkripsi yaitu 0,18 detik. Maka hasil perhitungannya sebagai berikut :

Brute Force = ———— tahun

Dari hasil perhitungan diatas, apabila seorang *attackers* ingin mencoba semua kemungkinan kunci yang ada pada aplikasi ini maka akan membutuhkan waktu selama 1,652 x tahun.

E. Keamanan Aplikasi (RSA)

Sistem ini menggunakan sepasang kunci yaitu, kunci publik dan kunci privat. Panjang kuncinya telah ditentukan, dimana semakin panjang bit pembentukan kunci maka semakin sukar untuk dipecahkan karena sulitnya memfaktorkan dua bilangan yang sangat besar (bilangan p dan q). Keamanannya sendiri terletak pada sulitnya memfaktorkan bilangan yang besar tersebut menjadi faktor-faktor prima. Pemfaktoran dilakukan untuk memperoleh kunci privat. Selama pemfaktoran bilangan besar menjadi faktor-faktor prima tersebut belum ditemukan, maka selama itu pula keamanan sistem ini tetap terjamin. Namun, dengan menggunakan bilangan yang besar tersebut dapat berpengaruh pada proses dekripsinya, yaitu prosesnya akan menjadi lambat. Ini merupakan salah satu kelemahan dari algoritma RSA.

Pada sistem ini sendiri, nilai bilangan p dan q yang digunakan masing-masing berjumlah 2 digit, namun dengan nilai tersebut tingkat keamanan pada sistem ini sudah cukup baik dan tidak mudah untuk diserang oleh *attackers.* Selain itu untuk proses dekripsinya tidak membutuhkan waktu yang lama.

V. PENUTUP

Adapun beberapa kesimpulan dari hasil pengujian dan analisis yang telah dilakukan pada aplikasi ini adalah sebagai berikut:

1. Hasil pengujian pengiriman sms pada aplikasi *Encryptor* dan *Decryptor*, sms yang telah terenkripsi akan terkirim apabila ≤ 160 karakter, dan sms tidak akan terkirim apabila ≥ 160 karakter karena sudah melebihi kapasitas batas karakter sekali sms yaitu sebanyak 160 karakter.
2. Hasil pengujian kecepatan pada proses enkripsi dengan menggunakan jumlah karakter yang berbeda diperoleh rata-rata waktu yaitu sebesar 0,18 detik. Sehingga dapat disimpulkan bahwa jumlah karakter tidak mempengaruhi waktu enkripsi. Dengan waktu tersebut maka user tidak membutuhkan waktu yang lama pada proses enkripsi pesan.
3. Hasil pengujian kecepatan pada proses dekripsi dengan menggunakan jumlah karakter yang berbeda diperoleh rata-rata waktu yaitu sebesar 0,18 detik. Sehingga dapat disimpulkan bahwa jumlah karakter

tidak mempengaruhi waktu dekripsi. Dengan waktu tersebut maka user tidak membutuhkan waktu yang lama pada proses dekripsi pesan.

4. Pada pengujian *avalanche effect* dapat disimpulkan bahwa dengan inputan plaintext yang berbeda 1 karakter tiap percobaan tetapi menggunakan kunci yang sama akan menghasilkan ciphertext yang berbeda. Presentase rata-rata yang dihasilkan pada pengujian ini yaitu sebesar 10,35 %.
5. Pada pengujian *brute force*, waktu yang dibutuhkan apabila seorang *attackers* ingin mencoba semua kemungkinan kunci yang ada yaitu selama 1,652 × tahun.

REFERENSI

[1] Ananda. "Materi Bilangan Prima Teori Bilangan Matematika Diskrit". https://www.academia.edu/. Diakses pada tanggal 10 April 2015

[2] Andikafisma. "Algoritma dan Pemrograman". https://andikafisma.wordpress.com/. Diakses pada tanggal 10 April 2015

[3] Cahyono, Yuna. "Pengertian". 13 November 2013. http://yunacahyono.blogspot.com/. Diakses pada tanggal 15 April 2015.

[4] Fahmizal. "Eclipse Meet AVR Plugin". 16 Februari 2013. https://fahmizaleeits.wordpress.com/. Diakses pada tanggal 19 April 2015

[5] Haidibarasa. "Pengertian Android Development Tools". 6 juli 2013. https://haidibarasa.wordpress.com /. Diakses pada tanggal 19 April 2015.

[6] Karya Tulis Ilmiah. "Pengertian Implementasi" Wahidatul Muharomia. 3 agustus 2014. http://karyatulisilmiah.com /. Diakses pada tanggal 20 april 2015

[7] Lestari, Estik. "Enkripsi dan Dekripsi". 31 Desember 2012. http://estiklestari.blogspot.com/. Diakses pada tanggal 15 April 2015

[8] Moeneagain, kis. "Android SDK (Software Development kit)". 6 januari 2013 http://kismeoneagains.blogspot.com/. Diakses pada tanggal 19 April 2015

[9] Mulyana, Rahmat Andi. 2014. "Pembangunan Aplikasi Keamanan SMS Menggunakan Algoritma RSA Berbasis Mobile pada Platform Android". Skripsi. Fakultas Teknik Dan Ilmu Komputer Universitas Komputer Indonesia.

[10] Naxeha. "OS Android". 12 Januari 2012. https://naxeha.wordpress.com/. Diakses pada tanggal 20 April 2015

[11] Romansa, Gestihayu. "Makalah RSA". https://www.academia.edu/. Diakses pada tanggal 20 April 2015

[12] Samsung. "Galaxy Y seri". http://www.samsung.com/. Diakses pada tanggal 22 April 2015.

[13] Suryatna, Dodi. "Kriptografi Klasik". https://dodisuryatna.wordpress.com/. Diakses pada tanggal 23 Juni 2015

[14] Wahidatul Muharomia. 2013. *Sistem Operasi Android* .