

Implementasi Aplikasi Kriptografi dengan Metode RSA pada Keamanan Teks Berbasis Android

Implementation of Cryptographic Applications with the RSA Method on Android-Based Text Security

Fani Fadillah Sinaga^{a,1}, Nurhayati^{b,2}

^aMahasiswa Program Studi Informatika Universitas Potensi Utama

^bDosen Program Studi Informatika Universitas Potensi Utama

Universitas Potensi Utama, K.L. Yos Sudarso KM 6,5 No. 3A Tj. Mulia - Medan

sinagafadillahfani@gmail.com, nurhayatimaulanaa@gmail.com

ABSTRAK

Keamanan data merupakan aspek penting dalam pengiriman data teks dan informasi melalui jaringan seluler. Menanggapi ancaman keamanan ini, berbagai jenis teknik keamanan telah dikembangkan untuk melindungi dan menjaga kerahasiaan data untuk mencegah orang yang tidak berhak mendapatkan informasi tersebut, terutama dengan menggunakan metode kriptografi. Salah satu metode kriptografi yang dapat digunakan adalah metode RSA. Proses enkripsi dan dekripsi algoritma RSA menggunakan kunci yang berbeda. Enkripsi menggunakan kunci publik sedangkan dekripsi membutuhkan kunci privat. Tingkat kesulitan menyerang cipher RSA tergantung pada kesulitan bilangan prima yang digunakan untuk cipher. RSA menggunakan dua bilangan prima untuk menurunkan kunci publik dan kunci privat yang akan digunakan dalam proses enkripsi dan dekripsi pesan. RSA digunakan dalam aplikasi ini untuk mengenkripsi pesan rahasia sebagai file untuk meningkatkan keamanan pesan rahasia. Pada penelitian ini dibuat sebuah aplikasi keamanan data teks dengan menggunakan metode RSA berbasis Android.

Kata Kunci : keamanan, kriptografi, RSA, android

ABSTRACT

Data security issues are an important aspect in sending text data and information via cellular networks. To deal with these security threats, various kinds of security techniques have been developed to protect and maintain the confidentiality of data to avoid people who are not entitled to obtain this information, namely using cryptographic methods. One of the cryptographic methods that can be used is the RSA method. The encryption and decryption process of the RSA algorithm uses a different key. The encryption process uses a public key while the decryption process requires a private key. The level of difficulty for attacking RSA encryption depends on the difficulty of finding the prime factor used for encryption. RSA uses two prime integers to get the public key and private key that will be used in the process of encrypting and decrypting messages. RSA is used in this application to encrypt secret messages in the form of files so that the security of the secret messages is stronger. In this study, a text data security application was produced using the Android-based RSA method.

Keywords: security, cryptographic, RSA, android

Info Artikel :

Disubmit: 27 Juni 2023

Direview: 14 Juli 2023

Diterima : 13 Agustus 2023

Copyright © 2023 - Journal UPU. All rights reserved.

1. PENDAHULUAN

Kita sekarang hidup di dunia di mana teknologi terus menciptakan produk-produk canggih yang memudahkan seluruh aktivitas manusia. Kemajuan ini membuat segalanya menjadi lebih mudah, banyak dari kita yang ketagihan atau bahkan terbuai dengan kemudahan akses terhadap teknologi saat ini. Tidak hanya pejabat tinggi atau pejabat pemerintah saja yang sangat perlu mengamankan data teks, namun semua pengguna aplikasi keamanan data teks juga sangat perlu mengamankan pesannya. Mengurangi risiko pesan disadap atau dicuri secara digital oleh pihak ketiga untuk tujuan kriminal dan perilaku.

Keamanan data sangat penting untuk menjaga kerahasiaan informasi, terutama informasi yang

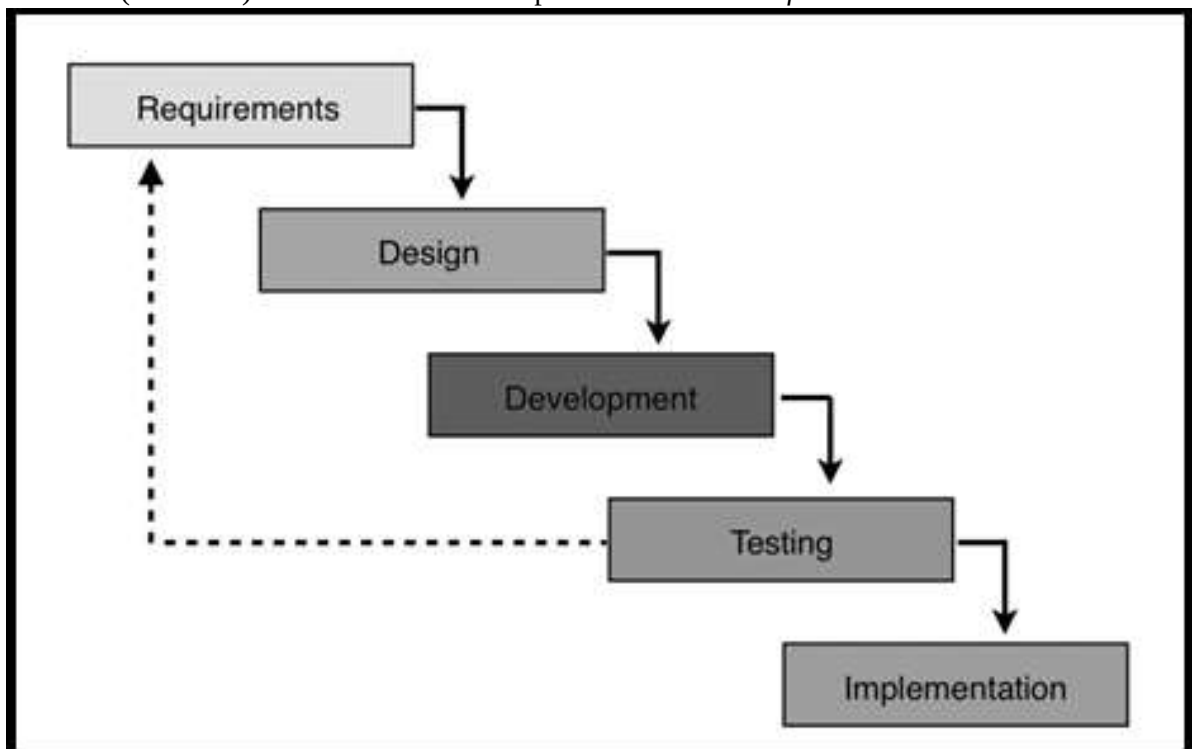
mengandung informasi sensitif yang isinya hanya dapat diketahui oleh pihak yang berwenang. Untuk mencegah kejahatan ini terjadi, maka perlu dilakukan pengamanan data pribadi. Salah satu metode yang digunakan untuk mengamankan data adalah penggunaan kriptografi, khususnya dengan menyediakan konten informatif (teks biasa) untuk konten yang tidak dipahami melalui proses enkripsi, dan untuk memulihkan salinan aslinya. asli, prosedur dekripsi dilakukan disertai prosedur yang benar. penggunaan kunci tersebut.

Banyak terjadi kasus penipuan yang mengatasnamakan seseorang yang bertujuan untuk menimbulkan kerugian bagi orang tersebut dan banyak pihak akibat tindakan seseorang yang menyadap data teks tersebut dan memodifikasinya, kemudian memodifikasinya dan menggunakannya untuk tujuan lain. keperluan pribadi atau lainnya. Salah satu upaya yang dapat digunakan untuk meningkatkan keamanan pesan adalah penggunaan kriptografi. Salah satu algoritma enkripsi yang paling banyak digunakan adalah algoritma RSA.

Oleh karena itu, untuk mencapai tingkat keamanan yang tinggi, metode RSA menggunakan kunci publik. Dari sekian banyak algoritma kriptografi kunci publik yang telah diciptakan, yang paling populer adalah algoritma ini. Keamanan algoritma RSA terletak pada kesulitan relatif dalam penguraian bilangan prima. Faktorisasi dilakukan untuk mendapatkan kunci privat. Selama faktorisasi bilangan prima yang besar tidak menemukan algoritma yang mampu menyelesaikannya, maka keamanan algoritma RSA terjamin.

2. METODE

Metode pengembangan sistem yang digunakan dalam penelitian ini menggunakan model waterfall. Model air terjun adalah model pengembangan perangkat lunak di mana alur kerja dieksekusi secara berurutan (berurutan). Berikut ini adalah tahapan dari model *waterfall* :



Gambar 1. Model Waterfall

3. HASIL DAN PEMBAHASAN

Enkripsi :

- Plainteks disusun menjadi blok-blok $x_1, x_2, \dots$, sedemikian sehingga setiap blok merepresentasikan nilai di dalam rentang 0 sampai $r - 1$.

- Setiap blok x_i dienkripsi menjadi blok y_i dengan rumus :
$$y_i = x_i^{PK} \bmod r \dots\dots\dots (8)$$

Dekripsi :

- Setiap blok cipherteks y_i didekripsi kembali menjadi blok x_i dengan rumus :
$$x_i = y_i^{SK} \bmod r \dots\dots\dots (9)$$

Misalkan plainteks yang akan dienkripsikan adalah $X = \text{HARI INI}$, atau dalam sistem desimal (pengkodean ASCII) adalah 7265827332737873.

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

$$\begin{array}{ll} x_1 = 726 & x_4 = 273 \\ x_2 = 582 & x_5 = 787 \\ x_3 = 733 & x_6 = 003 \end{array}$$

Nilai-nilai x_i ini masih terletak di dalam rentang 0 sampai 3337 – 1 (agar transformasi menjadi satu-ke-satu).

- Blok-blok plainteks dienkripsikan sebagai berikut:

$$\begin{array}{l} 726^{79} \bmod 3337 = 215 = y_1 \\ 582^{79} \bmod 3337 = 776 = y_2 \\ 733^{79} \bmod 3337 = 1743 = y_3 \\ 273^{79} \bmod 3337 = 933 = y_4 \\ 787^{79} \bmod 3337 = 1731 = y_5 \\ 003^{79} \bmod 3337 = 158 = y_6 \end{array}$$

Jadi, cipherteks yang dihasilkan adalah $Y = 215\ 776\ 1743\ 933\ 1731\ 158$

- Dekripsi dilakukan dengan menggunakan kunci rahasia :
 $SK = 1019$

- Blok-blok cipherteks didekripsikan sebagai berikut:

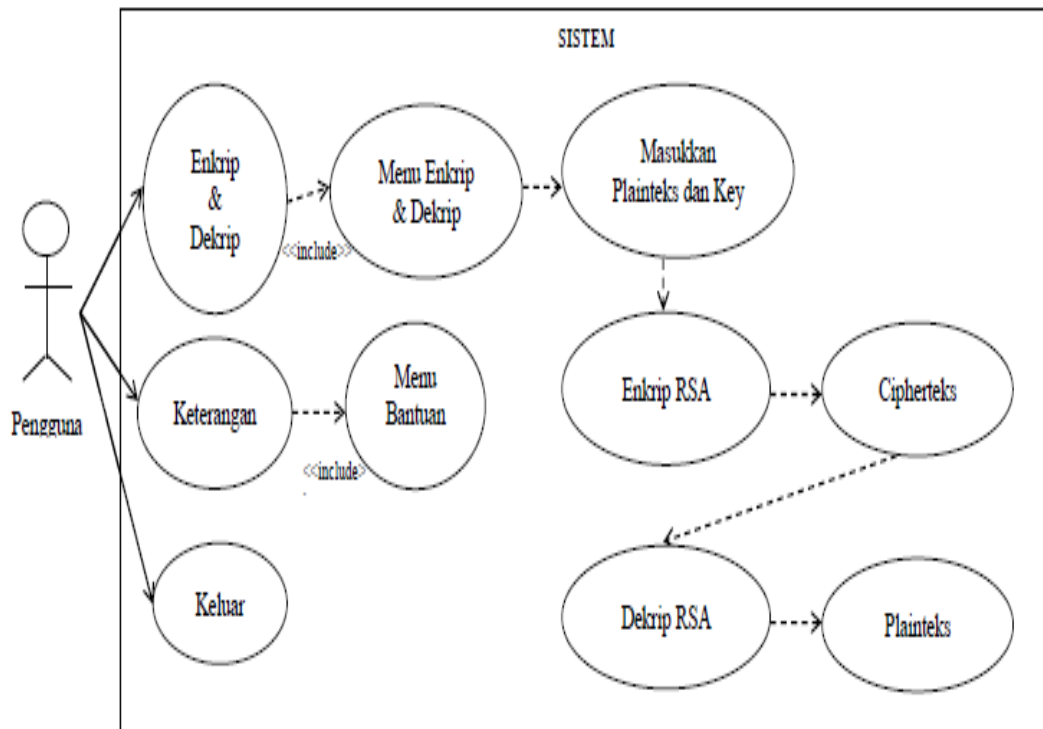
$$\begin{array}{l} 215^{1019} \bmod 3337 = 726 = x_1 \\ 776^{1019} \bmod 3337 = 582 = x_2 \\ 1743^{1019} \bmod 3337 = 733 = x_3 \\ \dots \end{array}$$

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

$$P = 7265827332737873$$

yang dalam karakter ASCII adalah :

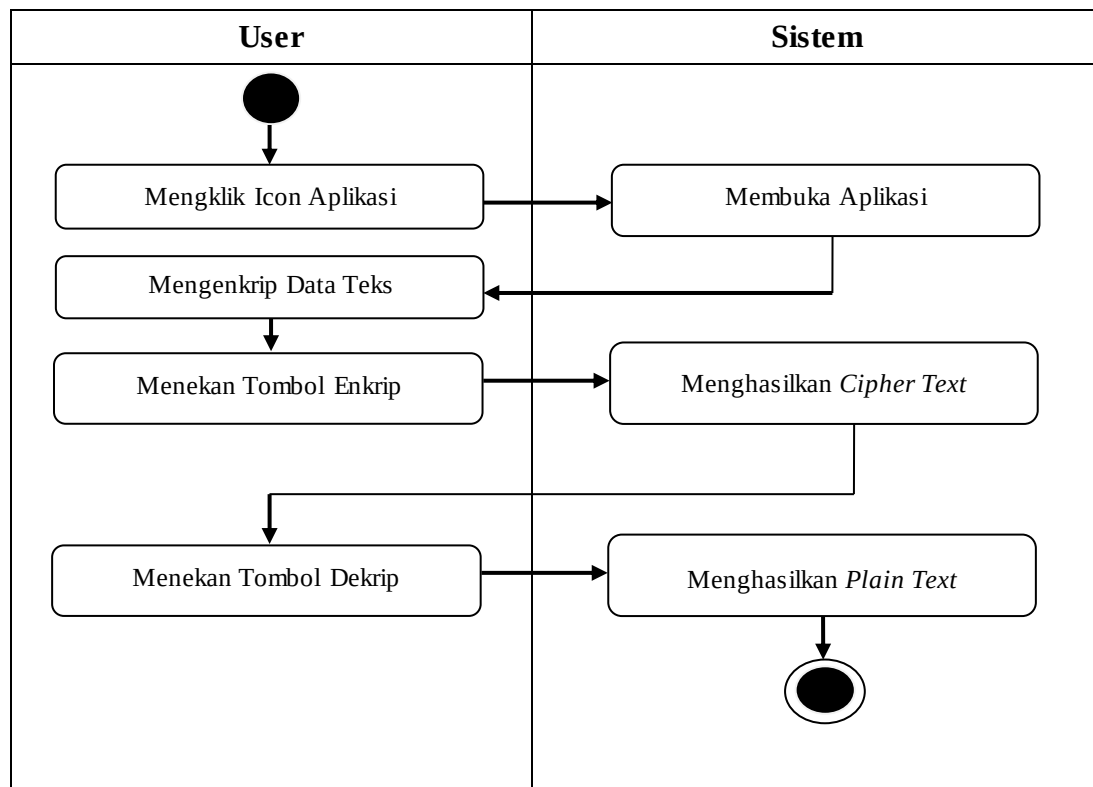
- . Maka digambarlah suatu bentuk diagram *Use Case* yang dapat dilihat pada gambar 3 berikut :



Gambar 2. Use Case Diagram

Pada diagram use case di atas terlihat pengguna akan memilih menu Encrypt & Decrypt untuk membuka aplikasi Encrypt & Decrypt. Jika pengguna memilih menu Deskripsi, maka akan muncul menu bantuan tentang cara menggunakan Encoding. & Decoding dan informasi tentang program pembuatan data biologis. Pengguna dapat memilih tombol Keluar untuk keluar dari aplikasi. Pengguna akan memilih menu untuk membuka aplikasi Enkripsi & Dekripsi, kemudian pengguna akan mengenkripsi data teks menggunakan metode RSA, kemudian pengguna akan menekan tombol Dekripsi untuk mendekripsi enkripsi teks.

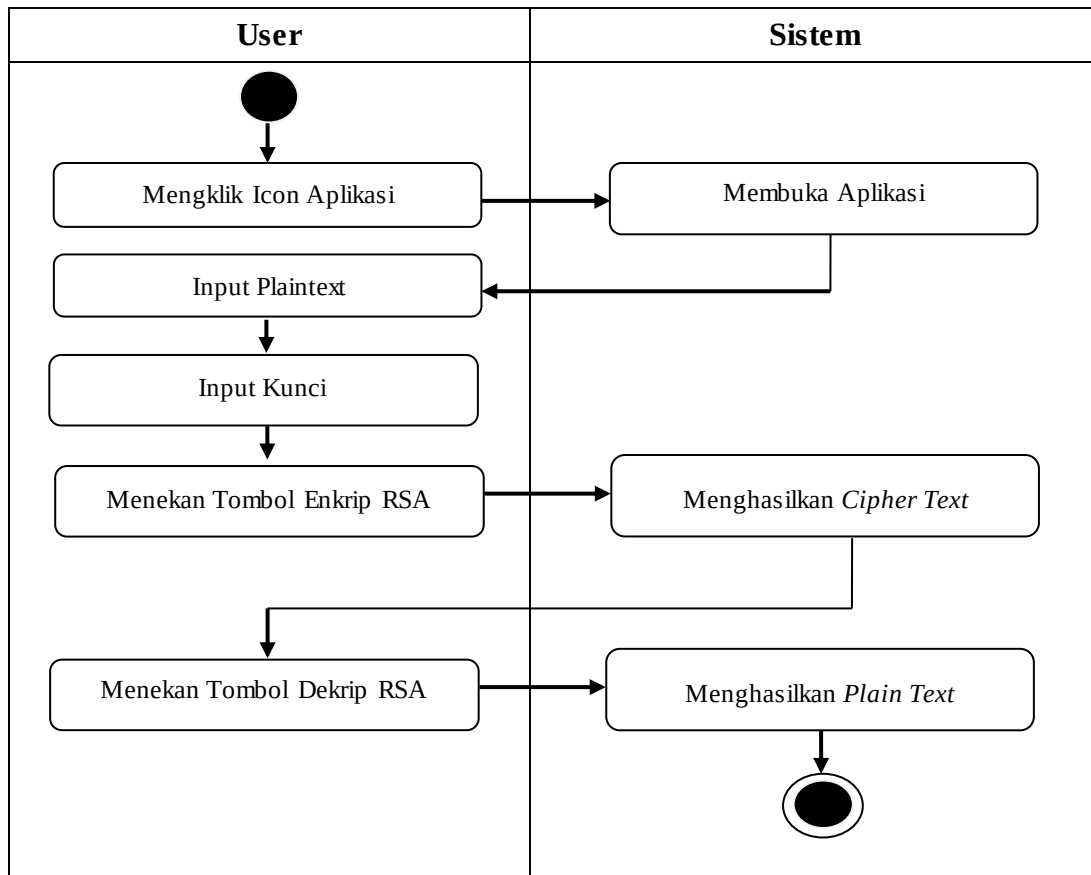
Diagram aktivitas menggambarkan berbagai aliran aktivitas dalam sistem yang dirancang, bagaimana setiap aliran dimulai, keputusan yang dapat dibuat, dan akhir operasi. Diagram aktivitas juga dapat menggambarkan proses paralel yang dapat terjadi dalam beberapa eksekusi. Diagram aktivitas pada aplikasi ini seperti di bawah ini.



Gambar 3. Activity Diagram Menu Enkrip & Dekrip

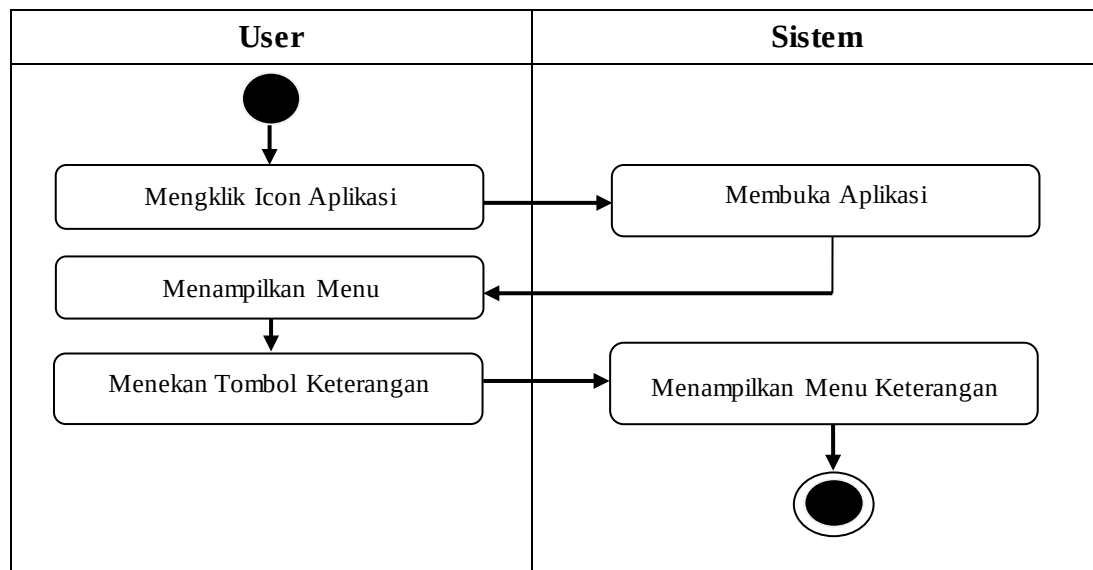
Dari diagram kerja di atas terlihat pengguna mengklik ikon aplikasi untuk membuka aplikasi Encrypt & Decrypt, kemudian pengguna mengenkripsi data teks dengan RSA, kemudian pengguna menekan tombol Decrypt untuk mendekripsi kode. budaya teks. Pengguna dapat menekan tombol keluar atau tombol menu untuk keluar dari aplikasi.

Pada saat enkripsi dan dekripsi RSA, pengguna mengklik ikon aplikasi untuk membuka aplikasi, kemudian memasukkan teks biasa, memasukkan kunci, lalu menekan tombol Enkripsi RSA untuk mengenkripsi teks biasa dengan RSA dan menghasilkan teks tersandi, kemudian pengguna menekan tombol Dekripsi untuk mendekripsi cipherteks. Pengguna dapat menekan tombol RSA Decrypt untuk mendekripsinya menjadi teks biasa.



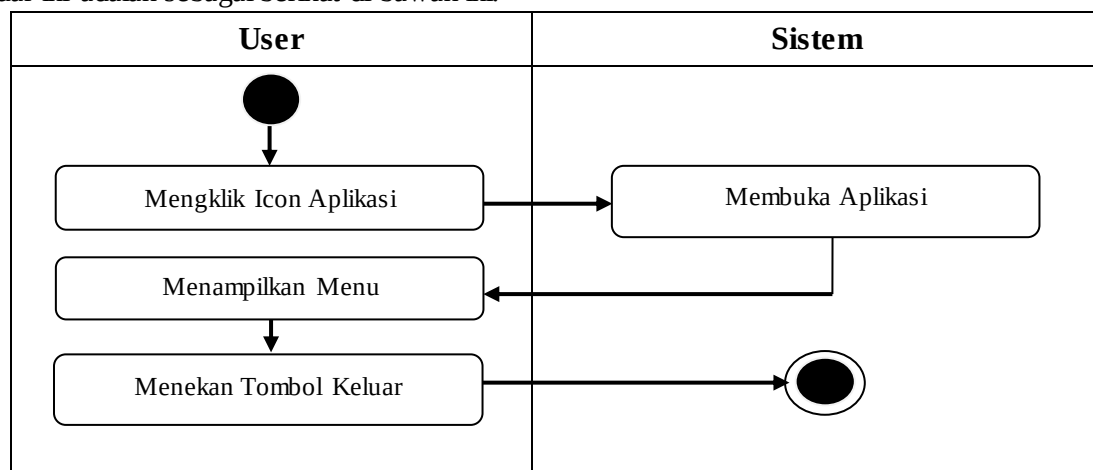
Gambar 4. Activity Diagram Proses Enkripsi dan Dekripsi RSA

User memilih menu Keterangan, maka aplikasi akan menampilkan menu keterangan yang berisikan informasi tentang pembuat program dan cara menggunakan aplikasi. Adapun diagram aktivitas pada menu Keterangan ini adalah sebagai berikut.



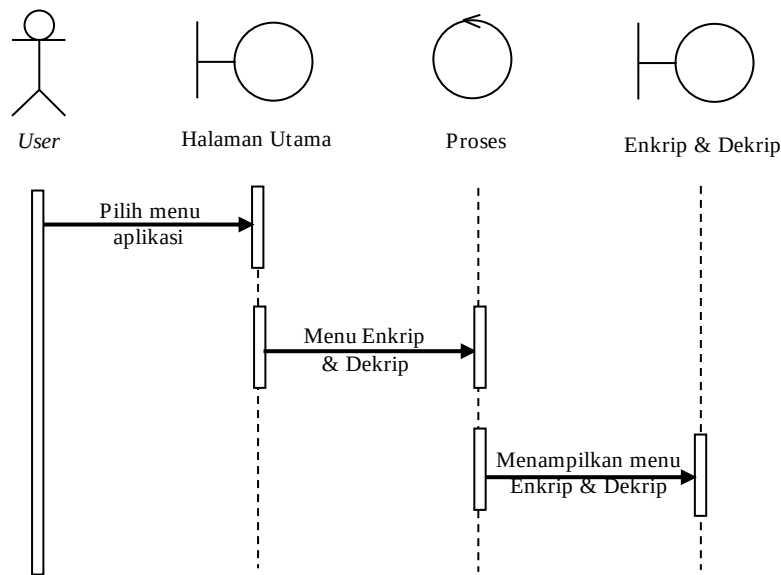
Gambar 5. Activity Diagram Menu Keterangan

User memilih menu Keluar, maka aplikasi tertutup. Adapun diagram aktivitas pada menu Keluar ini adalah sebagai berikut di bawah ini.



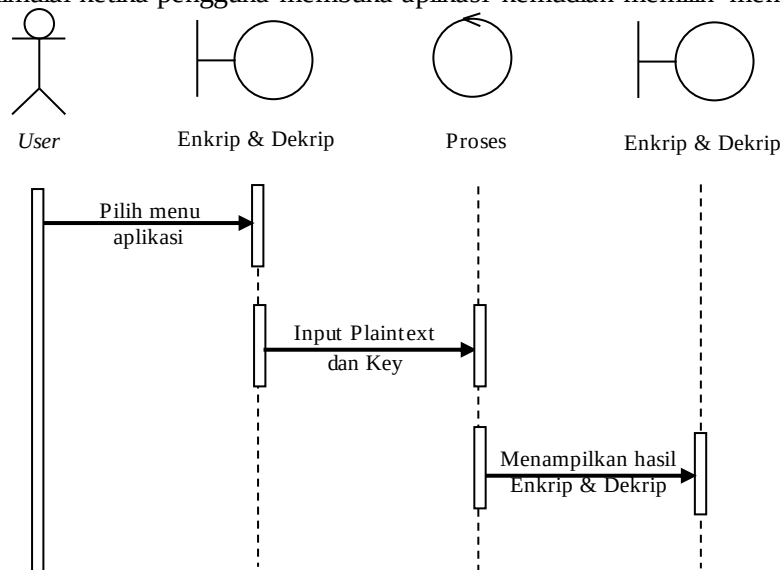
Gambar 6. Activity Diagram Menu Keluar

Adapun *sequence diagram* pada perancangan aplikasi ini terlihat pada Gambar III.6. berikut.



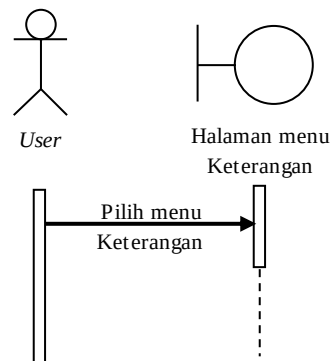
Gambar 7. *Sequence Diagram* Menu Enkrip & Dekrip

Diagram urutan pada Gambar 6. Tampilkan alur fitur dalam kasus penggunaan. Proses dalam use case dimulai ketika pengguna membuka aplikasi kemudian memilih menu Enkripsi dan Dekripsi.



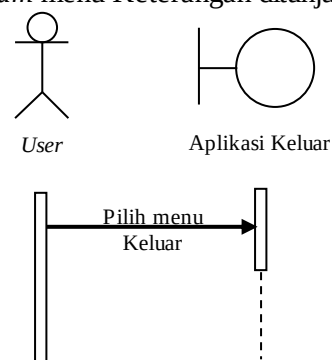
Gambar 8. *Sequence Diagram* proses Enkripsi & Dekripsi RSA

Sequence diagram pada Gambar 8 menunjukkan alur fungsionalitas dalam suatu use case. Proses dalam use case dimulai ketika pengguna membuka aplikasi, kemudian memasukkan plaintext dan kunci, mengenkripsi plaintext, dan menghasilkan ciphertext. Kemudian melakukan proses dekripsi untuk membuat ulang plaintext. Diagram urutan menu Deskripsi menggambarkan interaksi ketika memilih menu Deskripsi dalam suatu aplikasi. Diagram menu Urutan Deskripsi ditunjukkan pada Gambar 9.



Gambar 9. *Sequence Diagram* Tentang

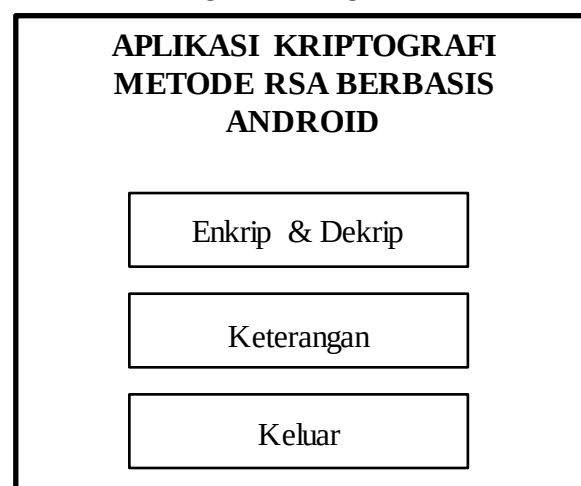
Sequence diagram menu Keluar menggambarkan interaksi saat memilih menu Keluar pada aplikasi. *Sequence diagram* menu Keterangan ditunjukkan pada gambar III.9.



Gambar 10. *Sequence Diagram* Keluar

Perancangan menu utama menampilkan 3 (tiga) pilihan menu, yaitu:

Menyandikan dan mendekode, mendeskripsikan, dan mengeluarkan. Jika menu Enkripsi & Dekripsi dipilih, maka akan muncul form enkripsi dan dekripsi untuk mengamankan data teks. Apabila menu Description dipilih maka akan ditampilkan form yang berisi petunjuk penggunaan aplikasi yang dirancang dan menjelaskan data biologis dari Program Builder:



Gambar 11. Form Menu Utama

Rancangan menu utama menampilkan 3 (tiga) pilihan menu yaitu: Enkrip & Dekrip,

Keterangan dan Keluar. Ketika menu Enkrip & Dekrip dipilih maka akan muncul *form* enkripsi dan dekripsi untuk mengamankan data teks, ketika menu Keterangan dipilih akan ditampilkan *form* yang berisikan tentang cara menggunakan aplikasi yang dirancang dan keterangan biodata dari pembuat program

The screenshot displays a mobile application interface titled "APLIKASI KRIPTOGRAFI METODE RSA BERBASIS ANDROID". It features a vertical sequence of six rectangular buttons: "Masukkan Plainteks", "Masukkan Kunci", "Enkrip RSA (Public Key)", "Cipherteks", "Dekrip RSA (Private Key)", and "Plainteks".

Gambar 12. Form Enkrip & Dekrip

Rancangan *form* Keterangan berisi keterangan cara menggunakan aplikasi dan keterangan tentang biodata pembuat aplikasi. Rancangan form Keterangan dapat dilihat pada Gambar 13 berikut.

The screenshot shows a single screen with a large rectangular box containing the word "Keterangan" in bold, centered text.

Gambar 13. Form Keterangan

A. Tampilan Hasil

a) Tampilan Awal Aplikasi

Dengan menggunakan emulator pada *Eclipse Galileo*, diperoleh tampilan aplikasi enkripsi dan dekripsi yang dirancang. Tampilan awal ini adalah tampilan pertama sekali aplikasi dijalankan. *User* dapat menekan tombol kunci, ditarik ke tombol *speaker* untuk membuka menu aplikasi. Adapun tampilan awal aplikasi dapat dilihat pada Gambar 14 di bawah ini.



Gambar 14. Tampilan Awal Aplikasi

b) Tampilan Menu Utama Aplikasi

Pada gambar berikut ini adalah tampilan awal dari aplikasi enkripsi dan dekripsi yang berisi sub menu Enkrip dan Dekrip, Keterangan dan Keluar. Adapun tampilan awal dari aplikasi enkripsi dan dekripsi dan dapat dilihat pada Gambar 15

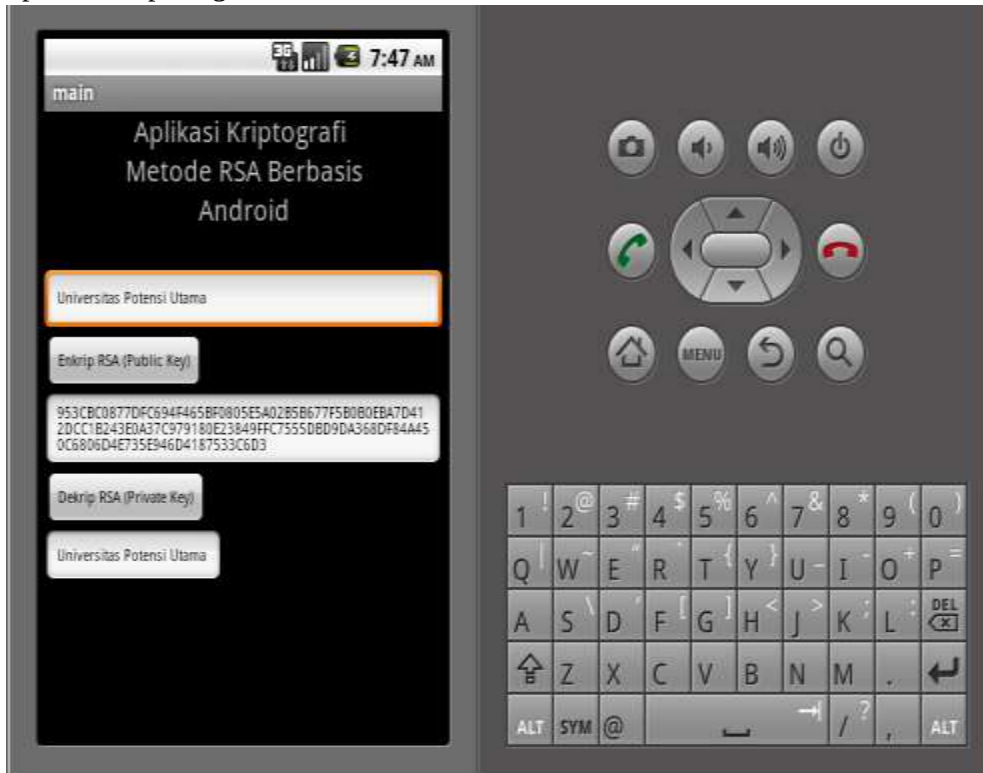


Gambar 15. Tampilan Menu Utama Aplikasi

c) Tampilan Enkrip dan Dekrip

Pada tampilan Enkrip dan Dekrip, terdapat kotak yang berisikan *plaintext*, isi *plaintext* tersebut lalu dienkrip menggunakan *public key* yang secara otomatis teregenarsi oleh aplikasi dengan menekan tombol Enkrip RSA (*Public Key*). Setelah dienkrip maka akan ditampilkan *ciphertext* sebagai hasil proses enkripsi. Lalu tekan tombol Dekrip RSA (*Private Key*) menggunakan *private key*, maka akan

ditampilkan kembali *plaintext*. Pada gambar berikut ini adalah tampilan dari aplikasi Enkrip dan Dekrip, dapat dilihat pada gambar 16.



Gambar 16. Tampilan Enkrip dan Dekrip

d) Tampilan Menu Keterangan

Pada gambar berikut ini adalah tampilan dari menu Keterangan yang berisikan tentang pembuat aplikasi dan cara menggunakan program. Adapun menu Keterangan pada aplikasi enkripsi dan dekripsi dan dapat dilihat pada Gambar 17.



Gambar 17. Tampilan Menu Keterangan

Adapun uji coba sistem yang telah dilakukan dapat dilihat pada tabel 1 sebagai berikut :

Tabel 1 Uji Coba Sistem

No	Nama Proses	Prosedur Pengujian	Hal Yang Diharapkan	Hasil Pengujian
1.	Tombol Enkrip dan Dekrip	Menekan tombol Enkrip dan Dekrip	Menampilkan menu Enkripsi dan Dekripsi	Sukses
3.	Tombol Keterangan	Menekan tombol Keterangan	Menampilkan menu Keterangan	Sukses
4.	Tombol Keluar	Menekan tombol Keluar	Keluar dari aplikasi	Sukses

4. KESIMPULAN

Keamanan data dengan kriptografi RSA menggunakan kunci publik memiliki keamanan yang tinggi karena sulit untuk mereduksi bilangan besar menjadi faktor prima dalam algoritma RSA.

a) Sistem yang dibangun adalah sistem keamanan data, yang berdampak pada pengamanan informasi berupa data teks. Dalam hal ini, pengamanan data teks dicapai dengan menggunakan metode RSA sebagai proses enkripsi dan dekripsinya.

b) Aplikasi pengamanan data teks ini akan dibangun pada platform mobile pada platform Android.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Universitas Potensi Utama yang telah banyak memberikan masukan dan saran dalam penyelesaian Skripsi ini.

REFERENSI

- [1] Haryanto, E. V. (2021). Desain Steganografi Untuk Keamanan Gambar Dengan Algoritma RSA dan LSB Berbasis Android. *CSRID (Computer Science Research And Its Development Journal)*, 11(3), 179–190.
- [2] Hendini, A. (2016). Pemodelan Uml Sistem Informasi Monitoring Penjualan Dan Stok Barang (Studi Kasus: Distro Zhezha Pontianak). *Jurnal Khatulistiwa Informatika*, 4(2).
- [3] Kuswanto, J., & Radiansah, F. (2018). Media Pembelajaran Berbasis Android Pada Mata Pelajaran Sistem Operasi Jaringan Kelas XI. *Jurnal Media Infotama*, 14(1).
- [4] Permana, A. A. (2018). Penerapan Kriptografi Pada Teks Pesan Dengan Menggunakan Metode Vigenere Cipher Berbasis Android. *Jurnal Al-Azhar Indonesia Seri Sains Dan Teknologi*, 4(3), 110. <https://doi.org/10.36722/Sst.V4i3.280>
- [5] Ratna, D. (2018). Implementasi Algoritma Rail Fence Cipher Dalam Keamanan Data Gambar 2 Dimensi. *Pelita Informatika: Informasi Dan Informatika*, 7(1), 38–42.
- [6] Siregar, H. F., & Siregar, Y. H. (2018). *Perancangan Aplikasi Komik Hadist Berbasis Multimedia*. 2(2), 113–121.
- [7] Siringoringo, R. (2020). Analisis Dan Implementasi Algoritma Rijndael (Aes) dan Kriptografi RSA Pada Pengamanan File. *Kakifikom Kumpul. Artik. Karya Ilm. Fak*, 2(01), 31–42.
- [8] Sulaiman, R., & Vebu, M. (2018). Peningkatan Keamanan Pesan Berbasis Android Menggunakan Algoritma Kriptografi Rsa. *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, 7(2), 116. <https://doi.org/10.32736/Sisfokom.V7i2.574>
- [9] Waluyo, S., & Kanahebi, D. V. (2021). Sistem Pengamanan File Menggunakan Algoritma Rc4 Berbasis Webbase Studi Kasus: Pt. Tjipta Jaya Bersama. *Semnas Ristek (Seminar Nasional Riset Dan Inovasi Teknologi)*, 5(1).
- [10] Yusfrizal, Y. (2019). Rancang Bangun Aplikasi Kriptografi Pada Teks Menggunakan Metode Reverse Cipher Dan Rsa Berbasis Android. *Jtik (Jurnal Teknik Informatika Kaputama)*, 3(2), 29–37.
- [11] Yusfrizal, Y., & Ramadhan, M. H. (2018). Sistem Pakar Diagnosa Awal Gangguan Attention Deficit Hyperactivity Disorder Pada Anak Dengan Metode Certainty Factor. *Jtik (Jurnal Teknik Informatika Kaputama)*, 2(2), 9–19.