JOIN (Jurnal Online Informatika)
Volume 4 No. 1 | Juni 2019 : 39-45
DOI: 10.15575/join.v4i1.264

Short Message Service Encoding using the Rivest Shamir-Adleman

Deny Prasetyo[1], Eko Didik Widianto[2], Ike Pertiwi Windasari[3]
[1,2,3]Computer System Department, Universitas Diponegoro, Semarang, Indonesia
[1]deny_pr@student.ce.undip.ac.id, [2]didik@live.ac.id, [3]ikerpertiwi@gmail.com

Abstract- SMS (Short Message Service) is one of the data exchange features in cellphones, including Android smartphones, which are the most widely used smartphone platforms today. However, the security of the SMS is questionable because the message sent must pass through a third party frist, namely SMSC (Short Message Service Center), so that message can be tapped or misused. One of the ways to reduce this risk is to encrypt or keep the original message secret by applying the cryptography algorithm. This research is to develop a system that serves to encrypt and decrypt SMS for Android-based smartphone users. The application is created to encrypt dan decrypt SMS using the Java programming language on Android, that is applied to smartphone integrated with Android Studio and used RSA cryptography algorithm. The application can be used to encrypt and decrypt SMS using the RSA algorithm on the Android-based smartphone. This application can send an SMS with a size of 86 characters and using QR Code to exchange the public key.

***Keywords*- Android, Cryptography, Rivest-Shamir-Adleman, Short Message Service**

I. INTRODUCTION

Short Messages Service or SMS is a wide-common feature that applicated in wireless communication which provides message delivery in an alphanumeric form between customer terminals with the external system such as email, paging, and voice mail[1]. SMS is one of the cellphone features including Android-based smartphones. Meanwhile, the security issues appeared in this feature in which the text that has been sent is through the third party at first, they are called as SMSC. This third party is a device that operates a store and forwards SMS traffic including a determination or SMS final destination search route[2]. This can cause the sent-text could be read or hijacked by an unauthorized party.

In this case, a system or application that could secure the contents of the SMS is highly necessary. The implementation is to adapt the cryptography algorithm to encode the text's content. Meanwhile, cryptography is a major that studying how to secure messages or data with certain techniques so it cannot read or hijacked by other unauthorized people[3]–[6]. The cryptography algorithm is divided into two forms that are the symmetrical algorithm and asymmetric. Furthermore, the cryptography algorithm is using one key in the process of both encryption and decryption of the messages while the asymmetric algorithm is using two different keys to its process or commonly called as public and private keys[7].

Atmojo[8] is succeeding in inventing SMS encryption applications that using RC6 symmetric cryptography algorithm. This application can both encrypt and decrypt the messages with similar keys and by inserting iteration in its process. Moreover, the process of encrypting and decrypting is needed to inserting key manually. Wicaksono [9] argued that the RSA is a solid encryption method to overcome security issues of message delivery to one network in electronic media and also there is no effective leak technique to this algorithm.

This research is aimed to create the encryption application of SMS using the Android-based RSA algorithm with the key substitution through QR Code. The RSA algorithm was created in 1978 by Ron Rivest, Adi Shamir, and Leonard Adleman[10]. This algorithm is included as the asymmetric algorithm in which the encrypt-decryption in this algorithm is using prime number and modulo arithmetic. Moreover, the public key (PU) is formed from e and n, n = p, q, while e is the integers between 1 and $\Phi(n)$, $\Phi(n) = (p-1)(q-1)$. Meanwhile, the private key (PR) was formed from d dan n, $d \equiv e\text{-}1 \pmod{\Phi(n)}$. As the encryption, calculation using the formula $C = M^e \bmod n$, and to decrypt the message is using formula $M = C^d \bmod n$. p and q is the prime number, n is parameter security, is the exponent of encryption, d is the exponent of decryption, M is plaintext and C is the ciphertext[11].

II. METHOD

This application is designed using Java language program and the IDE android studio. The basis of data in this application is saved locally using SQLite. Meanwhile, the system development is using Waterfall method which has five phases such are analyzing necessity, system design, implementation, assessment/test and maintenance[12].

A. Necessity Analysis

In the development of software, there are several system requirements consisting of functional and non-functional elements. Functional requirements which is describe the services, features or functions provided by the system that will be used by a user. Moreover, functional requirements analysis is important to know whether the needs of users have been accomplished. Non-functional

Received: October 18, 2018; Revised: April 25, 2019; Received: July 30, 2019; Published: August 31,2019

ISSN 2527-1682 (Print)
ISSN 2527-9165 (Online)

requirements are requirements that describe a set of constraints, characteristics, and properties in the system, both in the development and operational environment, or quality attributes that should exist in the system.

B. System Design

Based on the necessity design that can generally be illustrated by which the system is using UML modeling or Unified Modelling Language (UML). It is a tool to determine and visualize the system of software. This is also one of standard diagram which portrayed and mapped visually the computer application or the design of the system database and its structure [12], [13].

1. Usecase Diagram

Usecase diagram is used to support the development team to visualize the functional necessity of the system including the relations of "actor" (people who interact with the system) for such a crucial process as well as the relations inter-different Usecase. Figure 1 shows a usecase diagram from the SMS encryption application system that use RSA algorithm. In the usecase diagram that showed in figure 1, there is an actor-user

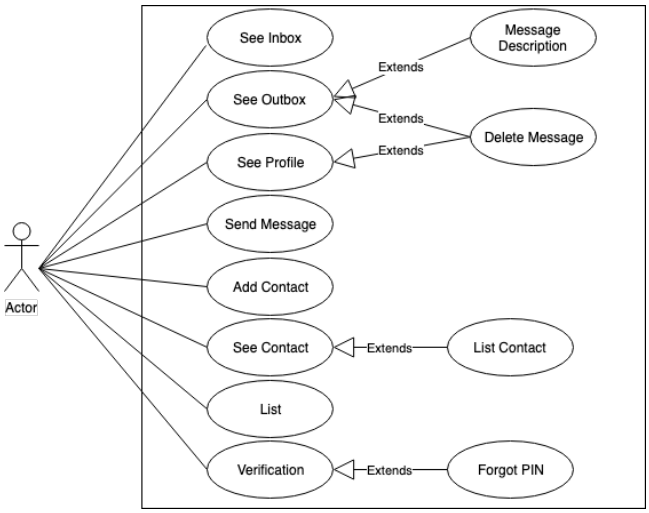

Figure 1. Use case diagram of encryption application of SMS using RSA SMS algorithm

2. Activity Diagram

The activity diagram can use to illustrate the activity or operational process in step-by-step the workflow of its components in the system. The activity diagram shows the whole flow of control. In figure 2 shows the activity diagram of delivering and encrypting a message. Furthermore, figure 3 shows the activity diagram of opening inbox and decrypting a message. The sequence of its activity is started from opening the application and ends with signing up or saving the data into database application at the same time.

Figure 2 shows an activity diagram from the process of delivering message encryption starting to open the page and select contact and it ends in to deliver and encrypt the message.

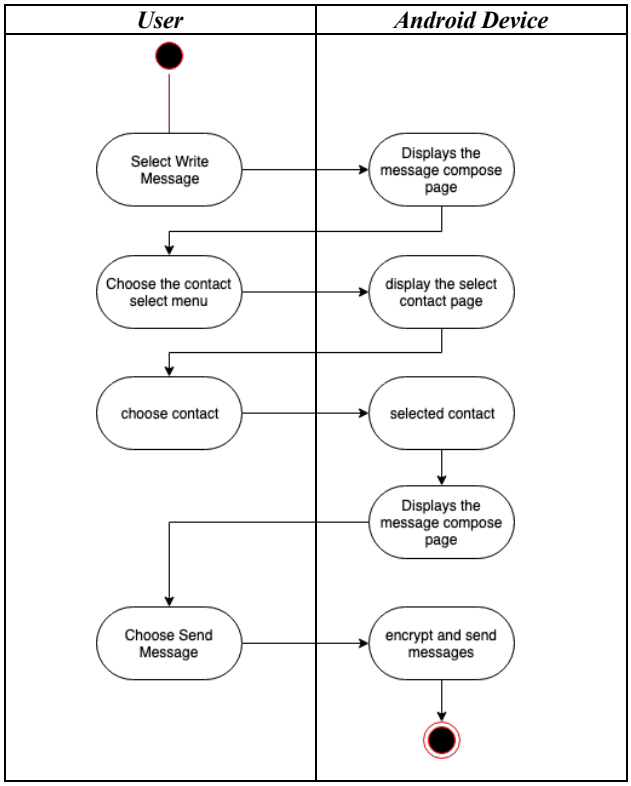

Figure 2. Activity diagram deliver and encrypt a message

Figure 3 shows an activity diagram from opening the inbox and decrypt the message starting in the inbox page and it ends in decrypt message

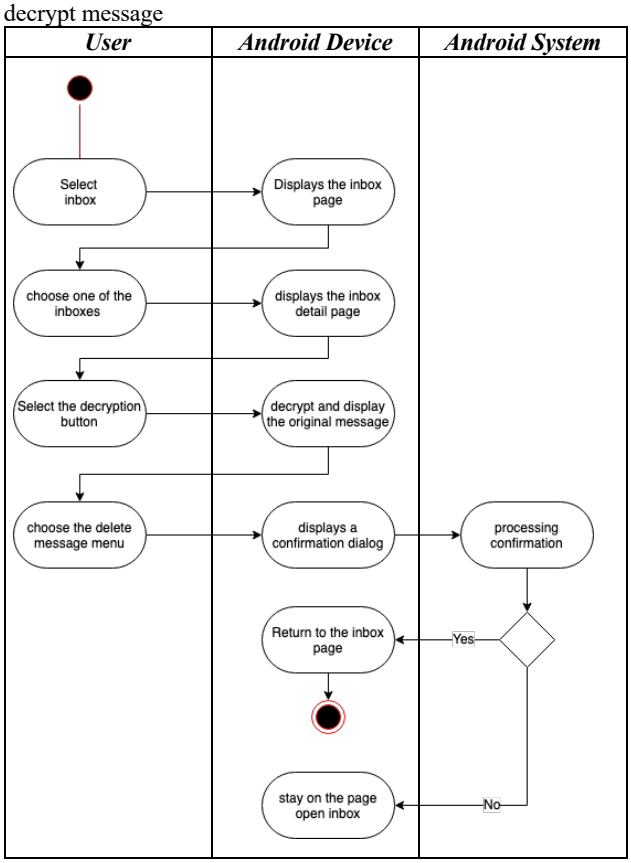

Figure 3. Acivity diagram opening inbox and decrypt the message

3. Database Design

The scheming of a database requires saving data that will be displayed in this application. The mentioned design is to create suitable with the system requirement [13]. The basis data of the process design to surveillance application using *Entity Relationship Diagram* (ERD) method.

Figure 4 shows the relation inter-entity wholly. The ERD from the system involving four entities that are inbox, outbox, contact, dan user.

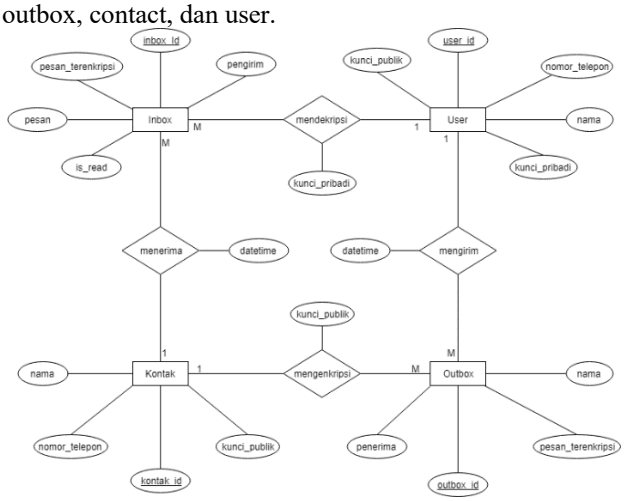

Figure 4. Relation inter-entity overall

III. RESULT AND DISCUSSION

A. Implementation

The implementation is one of the phases where the applying process from the existing design in this application. This implementation is aimed to make the interface application as well as the system previous design. The implementation is one of the phases where the applying process from the existing design in this application. This implementation is aimed to make the interface application as well as the system previous design. The Implementation is using Java Language program and IDE Android studio

1. Interface Implementation in Android Device

The interface display is made suitable with the interface previous design. The 'list page' is to submit the user data and creating both public and private key then saving it on the application basis. The 'create PIN' and security questions is to create PIN and save the security question. The illustration shows in figure 5.

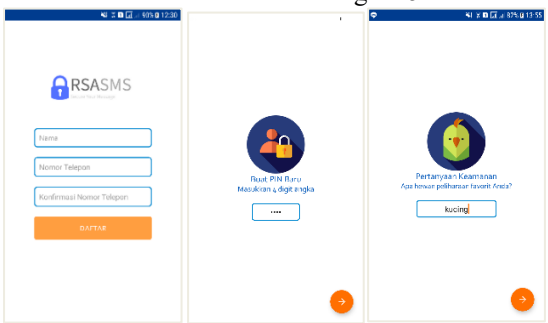

Figure 5. Sign-up page interface (left), Create PIN (middle) and security question (right)

The PIN verification page will appear when the application is opened twice and this page is to verify PIN that has been made by the user to get in into the application. This 'forgotten PIN page' is to check the security question that has inserted if a user forgets with the existing PIN. User's PIN is portrayed on the display page. The illustration shows in figure 6.

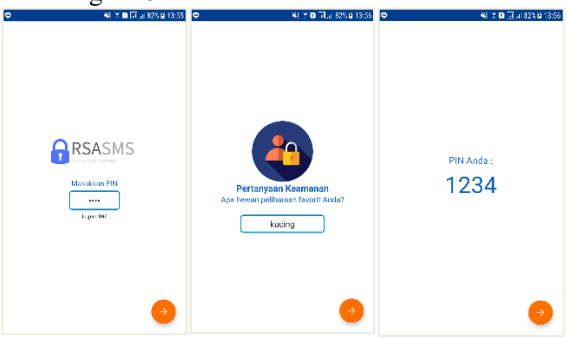

Figure 6. First Page interface (Inbox, Outbox, Profil)

The first page will display when the user is the success to verify PIN. This page consisting of the Inbox page, Outbox, and profile. The inbox including messages while the outbox contains sent messages. The profile page displays portrayed user complete information as shows in figure 7.

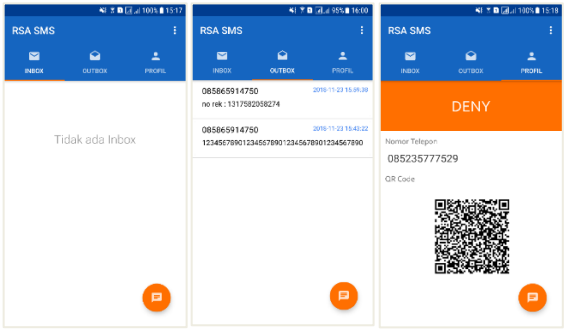

Figure 7. First-page interface (Inbox, Outbox, Profil)

The dropdown menu is located in the right-corner on the first page. This menu (figure 5) has several menu options to direct a user to another interface page.

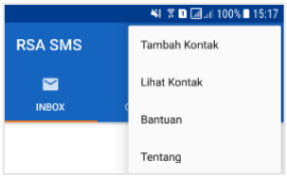

Figure Figure 8. Dropdown interface menu

This menu will appear when press the add contact menu is selected. This page will give the user direction to the QR code scan used to the scanning process and will automatically open add contact pace as seen in figure 9.

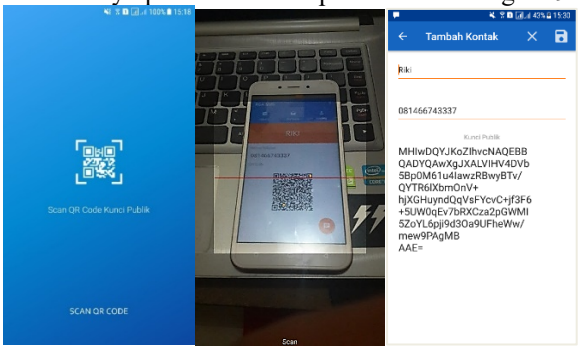

Figure 9. Start page interface Scan (left), Scan page QR Code, add contact page (left)

ISSN 2527-1682 (Print)
ISSN 2527-9165 (Online)

This 'select contact' will displayed when the 'write a message' button that in the main page. The 'select contact' consists of saved contact then select one of the contacts to open the 'write a message' page as can be seen in figure 10.

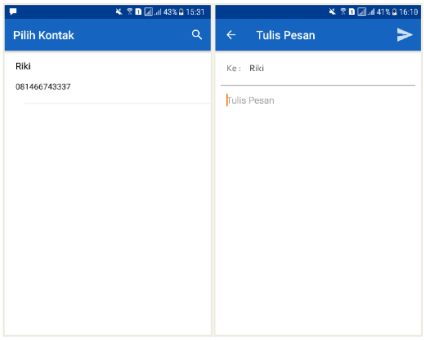

Figure 10. Select contact interface (left) and 'write a message' page (right)

The Open Inbox and Open Outbox pages will appear when one of the incoming or sent messages is selected. These two pages show the details of each message. There are several views on the Inbox Open Page and the Outbox Open Page, including the display of the Sender and Recipient Message, Encrypted and Original Message, Back Button, Decryption Button, and Delete button. Decryption button functions to translate encrypted messages into original messages. The display can be seen in Figure 11.

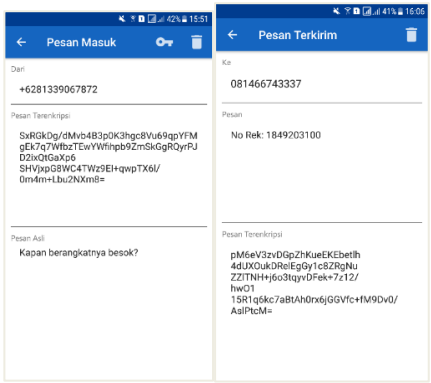

Figure 11. Selection page interface Kontak (left) and write message page (right)

The Contact List page displays a list of contacts that have been saved, while the Contact Open Page displays more complete information than the contacts that are opened. It can be seen in Figure 12.

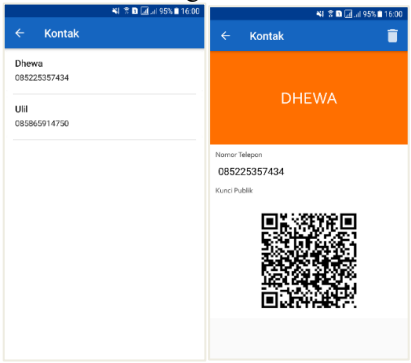

Figure 12. 'see contact' interface (left) and open contact page (right)

The Help page contains information about how to use the application. There are several views on this page, namely the Information Sending and Text SMS Reading Column, and the RSA Explanation Button. The RSA Explanation button when clicked will open the url address https://en.wikipedia.org/wiki/RSA_(cryptosystem) which contains information about the RSA algorithm. The display can be seen in figure 13.

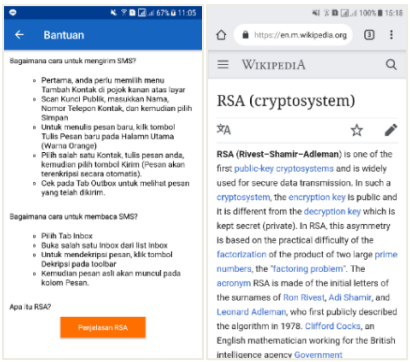

Figure 13. Help page interface (left) and explanation RSA (right)

The About page can be accessed from the Main Page Dropdown Menu. On this page there are two menus, namely the About Application Menu and the About Developer Menu. It can be seen in figure 14.

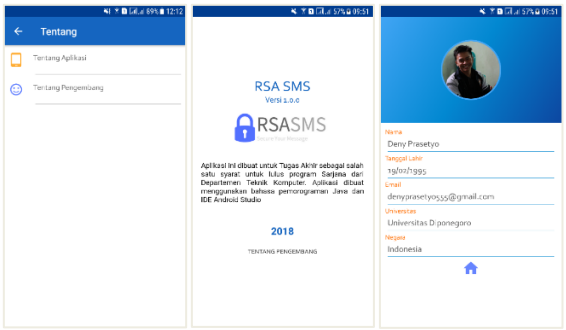

Figure 14. About page interface (left), application developer (right)

2. RSA Algorithm Implementation

This stage is to applying the RSA algorithm to the application. The RSA algorithm is used to create a public key pair as well as the message encryption and decryption process.

a. Key and SMS Size

The key size should be precise so that the length of the generated public key cannot be longer than 160 bytes, so it is necessary to select the length of the public key in the base64-encoded format that is most close to this limit. As can be seen in Table 1[14].

Table 1. Key and SMS size

Key Size (bit)	**Length of Public Key (*base64-encoded*)**
512	130
640	150
672	154
688	**159**
768	171
1924	21

The 688-bit key size is the best choice for the maximum length of SMS characters that have sent. So, the maximum number of characters that can be sent is the result of the size of the key divided by the number of bits in one byte, which is 688/8 bytes or 86 characters.

b. Creating Public and Private Keys

The key creation process is carried out in the RSA class. The key size made according to the explanation above is 688 bits. The program code below is used to create public and private key pairs in the application.

```
public void generateKeyPair() throws
NoSuchAlgorithmException,NoSuchPaddingException,
InvalidKeyException,IllegalBlockSizeException,
BadPaddingException {
   KeyPairGenerator kpg =
   KeyPairGenerator.getInstance("RSA");
   kpg.initialize(688);
   KeyPair kp = kpg.generateKeyPair();
   PublicKey publicKey = kp.getPublic();
   PrivateKey privateKey = kp.getPrivate();}
```

Method generateKeyPair()above will generate randomly generated public and private key pairs. The publicKey variable is used to hold the value of the public key while the privateKey variable is used to hold the private key.

c. The process of encrypt and decryption of the Message

The message encryption process takes place when the user clicks the Send button on the Message Write Page, the message written is encrypted first then sent to the recipient. The code below is used to encrypt the message before sending the message.

```
public String encrypt(String message, PublicKey
publicKey)throws
NoSuchAlgorithmException,NoSuchPaddingException,
InvalidKeyException,IllegalBlockSizeException,Ba
dPaddingException,UnsupportedEncodingException {
  Cipher cipher = Cipher.getInstance("RSA");
  cipher.init(Cipher.ENCRYPT_MODE, publicKey);
  byte[] data = Base64.decode(message,
  Base64.DEFAULT);
  byte[] encryptedBytes = cipher.doFinal(data);
  String encrypted = new String(encryptedBytes);
  }
```

Method `encrypt()` firstly is to change the message to Base64-encoded form, then encrypted using the recipient's public key, and re-encode it in the form of a String. The same thing applies in the message decryption process, which is written in the code below.

```
public String decrypt(String encryptedMessage,
PrivateKey privateKey)
throws NoSuchAlgorithmException,
NoSuchPaddingException,InvalidKeyException,
IllegalBlockSizeException,BadPaddingException,Un
supportedEncodingException {

     Cipher cipher = Cipher.getInstance("RSA");
     cipher.init(Cipher.DECRYPT_MODE,
     privateKey);
     byte[] data =
     Base64.decode(encryptedMessage,
     Base64.DEFAULT);
     byte[] decryptedBytes =
     cipher.doFinal(data);
     String decrypted = new
     String(decryptedBytes);
     return decrypted;
}
```

Method decrypt()is executed when the Decryption button is clicked on the Open Inbox Page. First the encrypted message is changed in the form base64-encoded, then decrypted using the private key of the recipient, the decrypted result is changed back in the form of a String, then displayed in the Decrypted Message column

B. Test/Assessment

1. Application Test into Android *Smartphone*

This test was performed on Android devices from version 5.0 Lollipop, version 6.0 Marshmallow to version 7.0 Nougat, and version 8.0 (Table .2)

Table 2. *Application Test into Android* Smartphone

Smartphone	**Spesification**	**Status**	**Notes**
Infinix X510	Android 5.1 Lolipop 720x1280px	succeed	Well-organized
Xiaomi Redmi Note 4X	Android 6.0.0 Marsmallow 720 x 1280 px	succeed	Well-organized
Samsung Galaxy A5	Android 7.0.0 Nougat 1080x1920px	succeed	Well-organized
Xiaomi Pocophone F1	Android 8.1.0 Oreo 1080x2246px	succeed	Well-organized

2. Encryption and Decryption Message Test

This test aims to see whether the encryption and decryption process on certain messages is well-organized. It can be seen in Table 3.

Table 3. Encryption and Decryption Message Test

Message	**Length (byte)**	***Ciphertext***	***Plaintext***
Kapan besok berangkatnya ?	21	WsMPSXQKzcVEwELRvYZcU4Fgy7guLNI04vBem94mWjdBQHIcctiRnfi5r4QSGV2B5wIXUWG4W4KqFw1x+mN80NIE2Phde2bCy1zWAeSt7EnrHz1UA8=	Kapan besok berangkatnya?
Kirim ke rekening ini aja: 007891417319. Saya tunggu sampe besok siang.	71	G2N62WwnBntY+wlWq5W4v7HsrlKKNOhZTyDaF7qx7me66PUnAmOFPBSB7rOxdUgu8wlxsDf11PjR4zryfJK5Cn6i3Ybz8Y2F4s7/L4z/272/bxOuj7og=	Kirim ke rekening ini aja: 007891417319. Saya tunggu sampe besok siang.
Kemarin saya ketemu orang itu. Berlagak mencurigakan, kemudian saya mengikutinya jalan	86	GCRTNsELLADJDc/tk4wAvUmdGGTEgi308ZfkgLRal8Ioesrkv+yP1TfWFAnr+loPsJU3/t6HzPt/ZewmTbRKuclCzETFCG+AKLCKwo4+lspfnmrK8hA=	Kemarin saya ketemu orang itu. Berlagak mencurigakan, kemudian saya mengikutinya jalan
Departemen Pertahanan	87	-	-

Message	Length (byte)	*Ciphertext*	*Plaintext*
Amerika melakukan uji coba serangan senjata nuklir terbaru mereka.			

3. The Test of Application Function in Android

Function testing is a test carried out on all functions contained in the application made. This test uses a black-box method by displaying the name of the test, the form of testing, until the test results[15]. Table 4 shows the test results on the register function and create a new PIN.

Table 4. the test of sign-up function and create a PIN

Test	Form of The Test	Expected Result	Test Result
Sign-up into system	Fill name and mobile number and click sign-up button	The sign-up success notification is appeared then the system displays the Login page with saved PIN	succeed
Create new PIN	Fill the PIN column and press the next button	PIN saved, sistem displays the question page of security question	succeed
Create Security Question	Fill the answer column and press next button	The answers are saved and system displays main page	succeed

The Testing of the PIN verification function and forgot the PIN on the application is successfull. Can be seen in Table 5.

Table 5. The test of the PIN Verification function and forgot the PIN

Test	Form of the Test	Expected result	Test Result
PIN verification	Fill the PIN and press the forgot PIN button	System displayed main page	Succeed
Forgot PIN	Fill the answer column then press next button	System displayed forgot PIN page then show PIN page	Succeed

The test of the function of adding new contacts to the application is also well-organized. The test results can be seen in Table 6.

Table 6. The test of add new contact function

Test	Form of the Test	Expected Result	Test Result
Add new contact	Click add contact	*Scan* page appeared	succeed
Conducting *scanning QR Code*	Press the login button and fill the login form	Start scan page appeared	succeed
Saving new contact	Click save button	Contact saved notification then go back to the main page	succeed

As the test of encryption function and send messages to the application are organized successfully. The test results can be seen in Table 7.

Table 7. encryption function test

Test	Form of the Test	Expected Result	Test Result
Write new message	Click the write message button	Displaying add contact page	succeed
Select destination contact	Click one of the contacts in select contact page	Displaying write a message page and the selected contact is displayed in the recipient column	succeed
Sending message	Write message in the message column then clicked the send button	The message is succeeded to encrypted and sent back to the main page	succeed

The process of testing the decryption function and delete messages on the application was successful. The test results can be seen in Table 8.

Table 8. decryption function and delete message test

Test	Form of the Test	Expected Result	Test Result
Open the incoming message	Clicked one of the messages in inbox page	Displaying open inbox page	succeed
Decrypting incoming message	Clicked the decryption button select 'yes' in the confirmation message	Confirmation pop up appeared and decrypted message in the decrypted message column	succeed
Deleting Incoming messages	Clicked delete button then select ';yes' in the confirmation message	Deleted messages notification appeared	succeed

The testing of view and delete outbox functions on the application is also successful. The test results can be seen in Table 9.

Table 9. test of open and delete *outbox* function

Test	Form of the Test	Expected result	Test Result
Open sent message	Clicked one of the messages in Out*box page*	Outbox page appeared	succeed
Delete sent message	Clicked delete button then select 'yes' in the confirmation message	Deleted message notification appeared	succeed

The test of functions see and delete contacts on the application are successful. The test results can be seen in Table 10.

Table 10. Test of functions see and delete contacts

Test	Form of the test	Expected result	Test Result
See the contact list	Click see contact Menu *Dropdown*	Shows Contact List page	succeed
See contact details	Click one of the contact in the contact list page	Displaying open contact page	succeed
Delete contact	Click delete button then press 'yes' in the confirmation message	Contact is deleted and go back to the main page	succeed

The test of see the Help Page function is well-organized and the test result can be seen in table 11.

Table 11. Test of the Help Page

Test	Form of the Test	Expected Result	Test Result
Open the Help Page	Click help in the menu in*Dropdown*	Shows help page	succeed
See the description of RSA	Click the description button of RSA	Shows the url page related RSA	succeed

The testing of function in seeing the About Page in the application was successful. The test results can be seen in Table 12.

Table 12. Test of the function of 'about page'

Test	Form of the Test	Expected Result	Test Result
Open the about page	Click 'about' in the menu of *Dropdown*	Displayed about page	succeed
Membuka Halaman Tentang Aplikasi	Click menu related to the application	Displayed page related to the application	succeed
Membuka Halaman Tentang Pengembang	Click menu related to the developer in about page	Displayed developer about page	succeed

C. Discussion

The SMS encryption application using the RSA cryptographic algorithm based on Android "RSA SMS" can well-organized on Android smartphones from Lollilop version 5.1 to Oreo 8.0. Appearance on the application made has a white background, with blue and orange. The RSA key size used is 688 bits, so the resulting public key has a length of 159 bytes. With this key length, 86 characters can be written in one SMS.

Moreover, Users need to register and login first in order to proceed to the main page of the application. While the data entered due to register then user data will be stored in the application database. Data stored in the form of the name, telephone number, public key, and private key. Your name, telephone number and a public key will be displayed in the form of a QR Code on the Profile Page, whereas because they are confidential, the private key will be displayed on any page.

The encryption process occured when the user presses the Send Message Button. Message encryption uses the public key of the selected contact. The decryption process occurs when the user presses the Decryption Button on the Message Open Page. The process of decrypting the message using the user's private key stored in the database.

The RSA SMS application testing that uses the black box testing method. Its kind of testing on each function of the RSA SMS application is successfully organized on which the main functions of this application are list, log in, add contacts, send messages and encrypt, decrypt messages, delete contacts, view the list of sent messages, view the list of incoming messages, and delete sent messages.

IV. CONCLUSION

This research is finally delineated that the SMS encryption application using the Android-based RSA algorithm that has been successfully designed and built using the Java programming language and Android Studio IDE. These applications that created can send SMS, receive SMS and can encrypt and decrypt SMS properlyonAndroid-based smartphones. The RSA key size used is 688 bits, so the maximum number of characters that can be sent in one SMS is 86 characters. The functions contained in the application were tested using the black-box testing method which was successful and ran as expected.

It is highly required that future advance research to conduct some of the restraints in the message or RSA keys in order to make the message is highly secured from hijacking or sabotaged. Above all, it is also required to conduct future research involving another cryptography algorithm so that the security of this system is an unpenetrated increase.

V. REFERENCES

[1] Abdiansah, "Membangun SMS-Gateway Untuk Pengisian Pulsa Elektronik," *J. Sist. Inf.*, vol. 1, no. 2, pp. 62–71, 2009.

[2] A. Saputra, *Membangun Aplikasi SMS dengan PHP dan MySQL*. Jakarta: Elex Media Komputindo, 2011.

[3] W. Stallings, *Cryptography and Network Security: Principles and Practices*. 2005.

[4] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. Taylor & Francis Inc, 1996.

[5] - Munsyi, A. Sudarsono, and M. U. H. Al Rasyid, "Secure Data Exchange Using Authenticated Attribute-Based Encdryption with Revocation for Environmental Monitoring," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 8, no. 5, p. 1948, Oct. 2018.

[6] E. M. D. L. Reyes, A. M. Sison, and R. P. Medina, "File encryption based on reduced-round AES with revised round keys and key schedule," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 16, no. 2, pp. 897–905, Nov. 2019.

[7] D Ariyus, *Kriptografi Keamanan Data dan Komunikasi*. Yogyakarta: Graha Ilmu, 2006.

[8] W. P. Atmojo, R. R. Isnanto, and R. Kridalukmana, "Implementasi Aplikasi Kriptografi Pada Layanan Pesan Singkat (SMS) Menggunakan Algoritma RC6 Berbasis Android," *J. Teknol. dan Sist. Komput.*, 2016.

[9] P. Wicaksono, "Enkripsi Menggunakan Algoritma RSA, Makalah Ilmu Komputer," 2013.

[10] W. Stallings, "Network and Internetwork Security Principle and Practice," *Prentice Hall*, 1995.

[11] J. Katz, *Introduction to Modern Cryptography*. 2007.

[12] M. Schader and A. Korthaus, *The Unified Modeling Language: technical aspects and applications*. 2012.

[13] W. Ieee, "Managing the Development of Large Software Systems Dr. Winston W. Royce Introduction," no. August, pp. 1–9, 1970.

[14] Hendra and Sukiman, "Aplikasi Pengaman Pertukaran SMS pada Perangkat Android dengan Metode RSA."

[15] B. Beizer and J. Wiley, "Black Box Testing: Techniques for Functional Testing of Software and Systems," *IEEE Softw.*, 2005.