CESS (Journal of Computer Engineering System and Science) **p-ISSN :2502-7131**
Vol. 6 No. 1 Januari 2021 **e-ISSN :2502-714x**

IMPLEMENTASI ALGORITMA LUC PADA APLIKASI KEAMANAN *SHORT MESSAGE SERVICE* (SMS) BERBASIS ANDROID (STUDI KASUS : BADAN NARKOTIKA NASIONAL PROVINSI SULAWESI TENGAH)

Yusuf Anshori[1], Deny Wiria Nugraha[2], Arief Pratomo[3]

[123]Jurusan Teknologi Informasi, Fakultas Teknik, Universitas Tadulako
Jl. Soekarno-Hatta KM.9, Tondo, Mantikulore, Kota Palu, Sulawesi Tengah 94148
[1]iyus.jr@gmail.com, [2]deny.wiria.nugraha@gmail.com, [3]ariefpratmo@gmail.com

Abstrak— ***Short Message Service*** **(SMS) masih memiliki masalah keamanan terhadap upaya pencurian pesan, penyadapan pesan, pembajakan pesan oleh pihak yang tidak bertanggung jawab untuk mendapatkan suatu informasi. Kriptografi adalah ilmu untuk menjaga kerahasiaan informasi dari aspek yang dapat mengancam keamanan informasi dengan algoritma dan teknik matematika tertentu. Algoritma Luc merupakan algoritma kriptografi kunci publik yang dikembangkan dengan menggunakan fungsi lucas. Penelitian ini bertujuan untuk mengimplementasikan algoritma Luc pada aplikasi keamanan SMS yang berbasis android di Badan Narkotika Nasional Provinsi Sulawesi Tengah untuk menjaga privasi dalam berkomunikasi antara pengirim dan penerima pesan. Aplikasi ini dibuat menggunakan pengembang android studio dengan bahasa pemrograman java. Berdasarkan hasil pengujian pertama untuk sampel pesan *plain* pada kata KOTA PALU yang menggunakan kunci publik 13 menghasilkan pesan *cipher* P=Se!cQa. Hasil pengujian kedua untuk sampel pesan pada kata KOTA PALU yang menggunakan kunci publik 29 menghasilkan pesan *cipher* AD*`_$(NQ.**

Kata Kunci— **Kriptografi, SMS, Android, Algoritma Luc, Enkripsi, Dekripsi, Kunci Publik.**

Abstract— ***Short Message Service*** **(SMS) still has security problems with attempts to steal messages, intercept messages, hijack messages by parties who are not responsible for obtaining information. Cryptography is the science of maintaining the confidentiality of information from aspects that can threaten information security with certain mathematical algorithms and techniques. The Luc algorithm is a public key cryptographic algorithm that was developed using the lucas function. This study aims to implement the Luc algorithm on an Android-based SMS security application in the National Narcotics Agency of Central Sulawesi Province to maintain privacy in communication between the sender and recipient of the message. This application was created using an Android studio developer with the Java programming language. Based on the results of the first test for a sample of plain messages in the word KOTA PALU that uses public key 13 produces the message cipher P=Se!CQa. The second test result for the message sample on the word KOTA PALU that uses public key 29 generates the AD*`_$(NQ cipher message.**

Keywords— **Cryptography, SMS, Android, Luc Algorithm, Encryption, Decryption, Public Key.**

I. PENDAHULUAN

Perkembangan teknologi dan informasi yang sangat pesat menghasilkan cara berkomunikasi jarak jauh dengan menggunakan pesan singkat pada ponsel genggam yang biasa disebut *Short Message Service* (SMS). Saat ini meskipun banyaknya layanan pesan instan canggih yang mengandalkan koneksi internet, SMS dapat menjadi pilihan alternatif untuk melakukan komunikasi ketika tidak adanya jangkauan koneksi internet di sebuah lokasi atau tempat. Namun, layanan ini masih sangat rentan terhadap upaya pencurian, penyadapan, pembajakan dan banyak hal lain oleh pihak yang tidak bertanggung-jawab untuk mendapatkan informasi.

Badan Narkotika Nasional adalah sebuah Lembaga Pemerintah Non Kementerian (LPNK) Indonesia yang mempunyai tugas melaksanakan tugas pemerintahan di bidang pencegahan, pemberantasan penyalahgunaan dan peredaran gelap psikotropika, prekursor, dan bahan adiktif lainnya.

Dalam permasalahannya Badan Narkotika Nasional Provinsi Sulawesi Tengah dalam upaya penugasan untuk melakukan pencegahan dan pemberantasan penyalahgunaan narkoba, akan menggunakan SMS ketika lokasi tersebut sulit mendapatkan akses internet.

Pada umumnya, aplikasi SMS masih sangat jarang memiliki keamanan dalam proses pengiriman dan penerimaan pesan. Salah satu penyadapan yang biasanya terjadi adalah Man-in-middle Attack. Sebuah serangan dimana penyerang akan berada pada tengah- tengah komunikasi antara pengirim dan penerima. Seluruh pesan yang terkirim antara pengirim dan penerima harus melalui penyerang terlebih dahulu. Penyerang dengan bebas melakukan pencegatan, penyadapan, pengubahan, bahkan dapat memalsukan komunikasi antara pengirim dan penerima. Beberapa orang dengan berbagai cara mencoba mencuri informasi yang bukan hak mereka.

Oleh karena itu, diperlukan adanya suatu mekanisme baru yang dapat menjaga kerahasiaan pesan tersebut. Penerapan metode untuk mengamankan pesan ada bermacam-macam, salah satu dari metode yang ada yaitu kriptografi. Kriptografi adalah ilmu untuk menjaga kerahasiaan informasi dari aspek-aspek yang dapat mengancam keamanan suatu informasi dengan metode dan teknik matematika tertentu. Dalam perancangan aplikasi keamanan SMS ini, penulis akan mengimplementasikan operasi fungsi lucas yang ada pada algoritma Luc untuk melakukan penyandian pesan.

Berdasarkan uraian di atas, penulis berencana mengambil suatu solusi untuk menyelesaikan masalah tersebut melalui skripsi yang berjudul "Implementasi Algoritma Luc Pada Aplikasi Keamanan Short Message Service (SMS) Berbasis Android (Studi Kasus: Badan Narkotika Nasional Provinsi Sulawesi Tengah)".

II. TINJAUAN PUSTAKA

A. Tinjauan Pustaka

Beberapa hasil penelitian terdahulu yang dijadikan referensi pada penelitian yang dilakukan oleh penulis adalah sebagai berikut:

1. Menurut penelitian yang dilakukan oleh Anggraini (2016), yang berjudul *"*Implementasi Algoritma Luc Pada Pengamanan Citra Digital Berbasis *Desktop"*. Penelitian tersebut menggunakan algoritma Luc untuk diimplementasikan ke dalam proses enkripsi dan dekripsi pada pengamanan citra digital. Hasil proses enkripsi dengan menggunakan algoritma Luc mampu mengaburkan piksel citra sehingga informasi di dalam citra digital tersebut tidak dapat diketahui. Penelitian tersebut membahas mengenai citra digital *truecolor* yang memiliki format gambar .bmp dimana citra yang diteliti maksimal berukuran 300x300 piksel. Persamaan penelitian ini dengan penelitian yang dilakukan oleh penulis yaitu kedua-duanya menggunakan algoritma luc. Di sisi lain, perbedaan penelitian ini dengan penelitian yang dilakukan oleh penulis, penelitian ini diimplementasikan pada citra digital yang berbasis *desktop*, sementara penelitian yang dilakukan oleh penulis diimplementasikan pada SMS berbasis android. [1]
2. Penelitian yang sama juga dilakukan oleh Alpha (2017), yang berjudul "Kriptografi *Visual* Dengan Implementasi Algoritma Luc Pada Citra Berwarna". Dalam penelitian tersebut algoritma Luc akan digunakan sebagai algoritma kriptografi *visual* untuk mengenkripsi dan mendekripsikan satu gambar berwarna. Pengujian dilakukan dengan 4 citra yang berbeda. Dilakukan penilaian dengan SSIM (*Structural SIMilarity*) untuk membandingkan nilai matriks citra awal dengan citra hasil dari dekripsi. Hasil dari pengujian SSIM adalah 1 untuk semua citra. Hal ini membuktikan bahwa citra awal dan citra hasil dekripsi mempunyai nilai matriks yang sama. Penelitian tersebut mempunyai tujuan untuk merealisasikan penyembunyian gambar menggunakan kriptografi *visual* dengan algoritma Luc melalui bantuan perangkat lunak MATLAB dan membandingkan gambar sebelum dilakukan enkripsi menggunakan algoritma Luc dengan gambar hasil dari dekripsinya. Ditahap implementasinya, pada proses enkripsi dimana mengubah pesan awal (*plain image*) menjadi sebuah pesan yang bersifat *privacy* (*cipher image*), sedangkan dekripsi adalah proses mengubah kembali pesan *privacy* (*cipher image*) menjadi pesan awal (*plain image*). Persamaan penelitian ini dengan penelitian yang dilakukan oleh penulis yaitu kedua-duanya menggunakan algoritma Luc untuk mengamankan pesan. Di sisi lain, perbedaan penelitian ini dengan penelitian yang dilakukan oleh penulis, penelitian ini mengimplementasikannya masih berbasis *dekstop*, sementara penelitian yang dilakukan oleh penulis yaitu berbasis android. [2]
3. Penelitian yang sama juga dilakukan oleh Yusfrizal (2015), yang berjudul "Penerapan Algoritma RC6 untuk Perancangan Aplikasi Pengamanan SMS Pada *Mobile Device* Berbasis Android". Dalam penelitian tersebut dapat mengimplementasikan kriptografi simetris RC6 untuk enkripsi dan dekripsi pesan dimana untuk pembangkitan kunci algoritma RC6 melakukan proses pembangunan kunci yang identik dengan algoritma RC5, yang membedakan hanyalah pada algoritma RC6, jumlah kata yang diambil dari kunci yang dimasukan oleh pengguna ketika melakukan enkripsi ataupun dekripsi lebih banyak. Tujuan dari proses pembangunan kunci tersebut adalah untuk membangun suatu *array* S yang berukuran 2r+4 dari kunci masukan pengguna sepanjang b *bytes* ($0 \leq b \leq 255$), *array* tersebut akan digunakan baik dalam proses enkripsi maupun dekripsi. Proses untuk membangun kunci-kunci internal menggunakan dua buah konstanta yang disebut dengan "*magic constant*". Persamaan penelitian ini dengan penelitian yang dilakukan oleh penulis

yaitu kedua-duanya mengamankan SMS dengan berbasis android. Di sisi lain, perbedaan penelitian ini dengan penelitian yang dilakukan oleh penulis, penelitian ini menggunakan algoritma RC6, sementara penelitian yang dilakukan oleh penulis menggunakan algoritma Luc. [3]

4. Penelitian yang sama juga dilakukan oleh Kusumawati (2018), yang berjudul "Kriptosistem Kunci Publik Luc Serta Implementasinya Pada Program Lazarus". Penelitian tersebut mengkaji kriptosistem kunci publik Luc sebagai perkembangan dari kriptosistem kunci publik RSA serta implementasinya pada program Lazarus. Penelitian diawali dengan membahas mengenai kriptosistem kunci publik RSA. Selanjutnya dibahas mengenai tanda tangan digital serta skema tanda tangan menggunakan algoritma RSA. Langkah berikutnya dibahas mengenai salah satu kelemahan pada algoritma RSA yakni penipuan tanda tangan RSA menggunakan *adaptive chosen message attack*. Serangan ini mungkin terjadi karena proses enkripsi dan dekripsi pada algoritma RSA menggunakan perkalian bilangan asli. Persamaan penelitian ini dengan penelitian yang dilakukan oleh penulis yaitu kedua-duanya menggunakan algoritma Luc. Di sisi lain, perbedaan penelitian ini dengan penelitian yang dilakukan oleh penulis, penelitian ini implementasi penggunaanya dibuat berbasis *desktop*, sementara penelitian yang dilakukan oleh penulis yaitu berbasis android. [4]
5. Penelitian yang sama juga dilakukan oleh Syamsinar (2017), yang berjudul "Implementasi Kombinasi Algoritma Asimetris Rivest Shamir Adleman Dan Algoritma Simetris AES Pada Aplikasi Pesan Singkat". Dalam penelitiannya dibutuhkan sistem yang dapat memberikan pengamanan terhadap isi pesan rahasia pada aplikasi pesan singkat. Penelitian tersebut mencoba membangun dan mengembangkan sebuah aplikasi pesan singkat menggunakan sistem *hybrid cryptosystem*. Aplikasi ini menggunakan teknik kriptografi dengan menggabungkan dua algoritma yakni algoritma asimetris *Rivest Shamir Adleman* untuk melakukan enkripsi dan deskripsi terhadap kunci simetris/sesi dan algoritma simetris *Advanced Encryption Standard* untuk melakukan enkripsi dan deskripsi terhadap pesan *plaintext* dan kunci simetris/sesi. Kemudian kedua algoritma ini akan diimplementasikan pada aplikasi pesan singkat menggunakan bahasa *java* dengan IDE *eclipse*. Persamaan penelitian ini dengan penelitian yang dilakukan oleh penulis yaitu kedua-duanya mengamankan SMS dengan berbasis android. Di sisi lain, perbedaan penelitian ini dengan penelitian yang dilakukan oleh penulis, penelitian ini menggunakan algoritma RSA dan AES, sementara penelitian yang dilakukan oleh penulis menggunakan algoritma Luc. [5]

III. ALGORITMA LUC

Algoritma Luc merupakan metode kriptografi dengan menggunakan dua kunci yang berbeda dalam kriptosistemnya. Untuk mengenkripsi pesan, digunakan fungsi enkripsi dengan menggunakan kunci publik. Hasil enkripsi merupakan pesan yang terenkripsi dari pihak yang berhak atas informasi didalamnya [6]. Selanjutnya untuk membaca pesan yang telah terenkripsi digunakan fungsi dekripsi dengan menggunakan kunci privat (*Private Key*) yang akan menghasilkan pesan yang sama dengan pesan awal sebelum dienkripsi. Untuk melakukan proses enkripsi, pesan harus dikonversikan kedalam bentuk angka ASCII (*American Standard Code for Information Interchange*). Dalam penerapan algoritma Luc terdapat 3 tahap utama yaitu pembangkitan kunci, proses enkripsi dan proses dekripsi yang penulis deskripsikan sebagai berikut:

1. *Pembangkitan Kunci*
   a. Kunci Publik
      a. Pilih dua bilangan prima acak, misal P dan *Q* dimana P ≠ Q.
      b. Hitung nilai N = P x Q. Nilai N akan digunakan dalam menghitung modulo pada proses enkripsi dan dekripsi.
      c. Hitung semua faktor prima terhadap (P-1), (P+1), (Q-1), dan (Q+1). Hasil faktor prima tidak akan dihimpun pada bilangan prima yang ada pada (P-1), (P+1), (Q-1), dan (Q+1).
      d. Pilih salah satu bilangan yang sama secara acak dari 4 buah hasil himpunan yang didapatkan pada poin (3) sebagai kunci publik (*e*).
   b. Kunci Privat
      1) Masukkan bilangan prima P dan Q.
      2) Masukkan kunci publik (*e*).
      3) Hitung determinan $D = C^2 - 4$.
      4) Cari simbol legendre dari $\frac{D}{P}$ dan $\frac{D}{Q}$.
      5) Hitung nilai *r* = LCM [(P - $\frac{D}{P}$) , (Q - $\frac{D}{p}$)].
      6) Hitung nilai *d* dengan menggunakan persamaan $ed \equiv 1 \bmod r$.
         $ed \equiv 1 \bmod r$
         $d = \frac{(k.r)+1}{e}$

Dimana *k* adalah bilangan prima acak yang menghasilkan bilangan bulat dari proses pembagian tersebut sehingga nilai *d* atau kunci dekripsi mempunyai kemungkinan sesuai dengan nilai *r*. Nilai *d* yang diperoleh merupakan kunci dekripsi (kunci privat) dari kunci enkripsi (kunci publik).

Proses pembangkitan kunci publik dilakukan dengan rahasia terutama nilai bilangan prima P dan Q, serta nilai *r* yang dipakai untuk dekripsi. Namun pendistribusian kunci publik tidak bersifat rahasia, karena tujuan dari kunci publik adalah untuk enkripsi.

2. Proses Enkripsi

Misalkan si pengirim ingin mengirim pesan kepada penerima, maka si pengirim pesan terlebih dahulu harus mempunyai kunci publik (*e*) yang didapatkan dari pembangkitan kunci publik. Selanjutnya proses enkripsi dapat dijelaskan sebagai berikut:

a. *Plaintext* (M) adalah isi pesan yang akan dikirim si pengirim pesan ke si penerima pesan.
b. Nilai *e* dan N didapatkan dari pembangkitan kunci publik.
c. *Plaintext* (M) yang akan dikirim kepada si penerima pesan, terlebih dahulu diatur menjadi blok-blok $m_1, m_2, \ldots m_i$ yang mempunyai dua karakter tiap bloknya.
d. Setiap blok yang telah didapatkan (m_i) diubah ke dalam ASCII kemudian dienkripsi dengan fungsi ≡ $c_i = V_e\ (m_i, 1)$ mod *N*.
e. Setiap blok yang telah dienkripsi c_i digabungkan kembali *ciphertext* yang utuh (*C*).

3. Proses Dekripsi

Misalkan si penerima pesan telah menerima *ciphertext* (*C*) dari si pengirim, maka langkah untuk melakukan dekripsi sebagai berikut:

a. *Ciphertext* yang telah diterima si pengirim diatur menjadi blok-blok $c_1, c_2, \ldots c_i$ yang memiliki dua karakter tiap bloknya.
b. Setiap blok yang ada (c_i) diubah ke dalam ASCII.
c. Hitung nilai determinan $D = C^2 - 4$.
d. Cari simbol legendre dari $\frac{D}{P}$ dan $\frac{D}{Q}$.
e. Hitung LCM [(P - $\frac{D}{P}$) , (Q - $\frac{D}{P}$)].
f. Hitung nilai *d* dengan menggunakan persamaan $ed \equiv 1 \bmod r$.
$$ed \equiv 1 \bmod r$$
$$d = \frac{(k.r)+1}{e}$$
g. Masukkan *d* untuk mendekripsi dengan fungsi ≡ $m_i = V_d\ (c_i, 1)$ mod *N*.

Setiap blok yang telah didekripsi (m_i) digabungkan kembali untuk menjadi *plaintext* semula (M).

IV. METODE PENELITIAN

A. Bahan Penelitian

Dalam penelitian ini, peneliti memiliki dua jenis data yaitu data primer dan data sekunder. Data primer pada penelitian ini adalah sampel pesan "KOTA PALU" yang didapatkan dari hasil wawancara kepada pegawai Kantor Badan Narkotika Nasional Provinsi Sulawesi Tengah. Data sekunder pada penelitian ini adalah bahan pustaka yang terkait dengan implementasi algoritma Luc pada aplikasi keamanan SMS berbasis android baik yang berasal dari jurnal, buku, skripsi, dan artikel *online*.

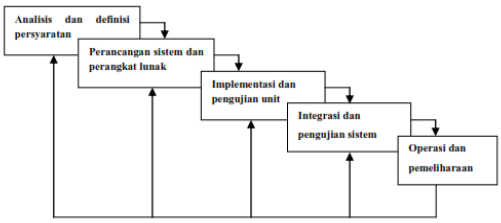
B. Alat Penelitian

Alat yang digunakan pada penelitian ini berupa perangkat lunak dan perangkat keras sebagai berikut:

1. Perangkat lunak (*Software*)
   a. *Android Studio* versi 3.4 sebagai IDE.
   b. *Java* sebagai bahasa pemrograman.
   c. *Android SDK* sebagai *framework* dari Android Studio.
   d. *Photoshop* untuk mendesain *prototype* aplikasi.
2. Perangkat keras (*Hardware*)
   a. Laptop dengan spesifikasi :
      1) Sistem operasi *Windows* 10 64 *bit.*
      2) *Processor intel core i3.*
      3) RAM 4 GB DDR 4.
      4) *Hard Disk Drive (*HDD*)* 500 *GB.*
   b. Dua buah smartphone android untuk penggunaan aplikasi dengan spesifikasi minimal android versi 6.0 (*Marshmallow*).

C. Metode Pengembangan Sistem

Pengembangan sistem pada penelitian ini menggunakan metode pengembangan *Waterfall. Waterfall model* merupakan salah satu model proses perangkat lunak yang mengambil kegiatan proses dasar seperti spesifikasi, pengembangan, validasi, evolusi, dan merepresentasikanya sebagai fase-fase proses yang berbeda seperti analisis dan definisi persyaratan, perancangan perangkat lunak, implementasi, pengujian unit, integrasi aplikasi, pengujian aplikasi, operasi, dan pemeliharaan. Adapun gambar metode pengembangan *waterfall* dapat dilihat pada Gbr. 1 berikut:



Gbr. 1 Metode *Waterfall*

1. Definisi Kebutuhan *(Requirements Definition)*
   Pada tahap ini, penulis mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh aplikasi yang akan dibangun. Adapun kebutuhan dari aplikasi ini yaitu dapat mengirim dan mengamankan SMS.
2. Perancangan Sistem dan Perangkat Lunak (*System and Software Design)*
   Pada tahap ini, penulis melakukan rancangan *software* yang akan dibuat yaitu dengan melihat analisis kebutuhan dari pengguna dimana diharapkan aplikasi ini dapat membantu keamanan pesan saat berkomunikasi *via* SMS. Kemudian dari kebutuhan tersebut dibuatlah beberapa fungsi seperti pembangkitan kunci publik, proses enkripsi,

pembangkitan kunci privat, proses dekripsi, buat SMS baru, penginputan nomor tujuan dari daftar kontak, dan pesan konfirmasi pengiriman kunci publik setelah pengiriman SMS. Setelah itu, penulis membuat rancangan *form* dari beberapa fungsi di atas sebelum nantinya masuk ke tahap pemrograman atau *coding*. Berikut adalah *form-form* dari aplikasi keamanan SMS dengan algoritma Luc:

a. *Form* Utama
Form ini adalah tampilan utama ketika aplikasi dijalankan. *form* utama bertujuan untuk menampilkan data dari daftar SMS yang masuk dari tiap nomor kontak. *form* utama juga memuat tombol pengaturan dan buat pesan baru.

b. *Form* Pengaturan Kunci Publik
Form ini adalah tempat untuk membangkitkan kunci publik. Proses pembangkitan kunci publik terlebih dulu dimulai dari pembangkitan sepasang bilangan prima P dan Q serta nilai N. Kemudian pembangkitan kunci publik akan menghasilkan beberapa kunci publik.

c. *Form* Pembuatan Pesan Baru
Form ini adalah tempat untuk membuat SMS baru. Pada *form* ini, proses enkripsi pesan terjadi ketika pengirim memasukkan pesan *plain* dan untuk mengirim pesan yang telah terenkripsi, pengirim harus memasukkan nomor tujuan. Setelah pesan terkirim, akan ada konfirmasi untuk pengiriman kunci publik.

d. *Form* Percakapan
Form ini berfungsi sebagai percakapan antara pengirim pesan dan penerima pesan. Pada *form* ini pesan yang sudah terenkripsi akan diberi aksi (*Long Click*) sebagai pengantar ke *form* selanjutnya untuk melakukan proses dekripsi pesan. Pada *form* ini setelah mengirim pesan, akan ada konfirmasi untuk pengiriman kunci publik.

e. *Pop-up menu* Pembangkitan Kunci Privat
Untuk membangkitkan kunci privat, penerima pesan harus memasukkan kunci publik yang telah dikirim oleh pengirim pesan.

f. *Form* Dekripsi Pesan
Pada *form* ini, pesan *cipher* yang diterima akan dilakukan proses dekripsi untuk mendapatkan pesan *plain* setelah kunci privat dibangkitkan.

3. Implementasi dan Pengujian Unit *(Implementation and Unit Testing)*
Pada tahap ini, Desain program atau *software design* diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang telah ditentukan. Penulis menggunakan Android Studio sebagai IDE dan bahasa pemrograman *java* untuk pembuatan aplikasi.

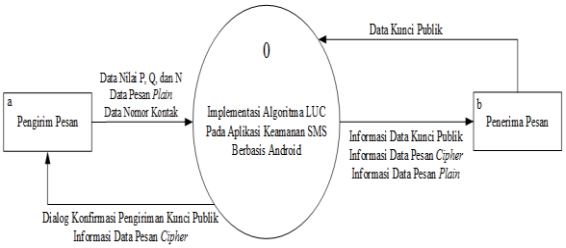
4. Integrasi dan Pengujian Sistem *(Integration and System Testing)*
Pada tahap ini, untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain diubah bentuknya menjadi dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses *coding*. Tahap ini merupakan tahap implementasi dari tahap desain. Penyatuan unit-unit program kemudian diuji secara keseluruhan (*system testing)*.

5. Operasi dan Pemeliharaan *(Operation and Maintenance)*
Pada tahap ini, perangkat lunak yang telah sukses dalam pengujian sistem dilakukan pemeliharaan yang meliputi pengembangan, perbaikan, dan penambahan fungsi ataupun persyaratan yang perlu dilakukan.

V. HASIL DAN PEMBAHASAN

A. Analisa Sistem

1. Diagram Konteks
Adapun gambar diagram konteks dapat dilihat pada Gbr. 2 berikut:



Gbr. 2 Diagram Konteks Implementasi Algoritma Luc Pada Aplikasi Keamanan SMS Berbasis Android

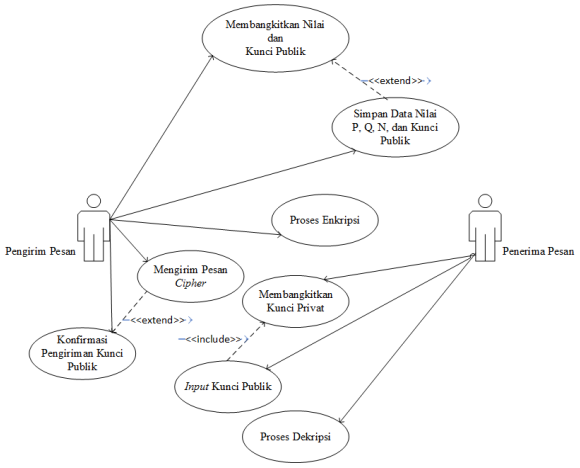
Keterangan Gbr. 2 dapat dilihat sebagai berikut:

a. Implementasi Algoritma Luc Pada Aplikasi Keamanan SMS Berbasis Android.
Implementasi dari algoritma Luc pada aplikasi keamanan SMS berbasis android ini akan menghasilkan pesan *cipher* yang dikirim dari pengirim pesan ke Penerima SMS.

b. Pengirim Pesan
Pengirim pesan memasukkan pesan *plain* agar diproses oleh aplikasi menjadi pesan *cipher* dan nomor kontak untuk tujuan pengiriman pesan.

c. Penerima Pesan
Penerima pesan akan menerima pesan *cipher* dan kunci publik kemudian diproses oleh aplikasi agar menghasilkan pesan *plain.*

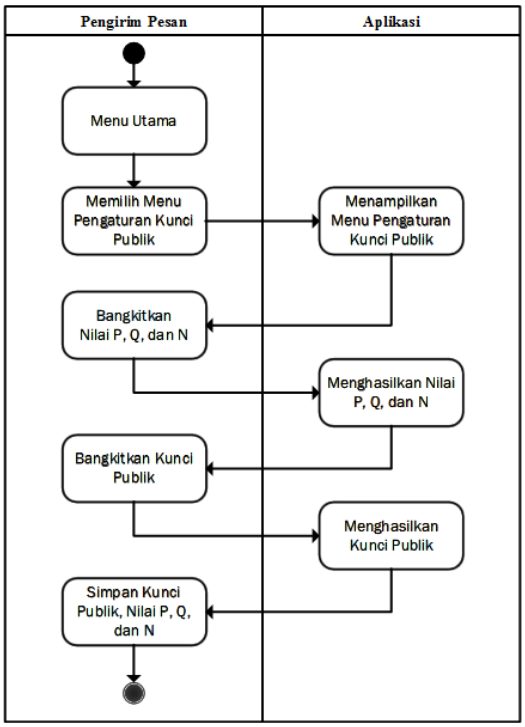
2. *Usecase* Diagram
Adapun gambar *usecase* diagram dapat dilihat pada Gbr. 3 berikut:



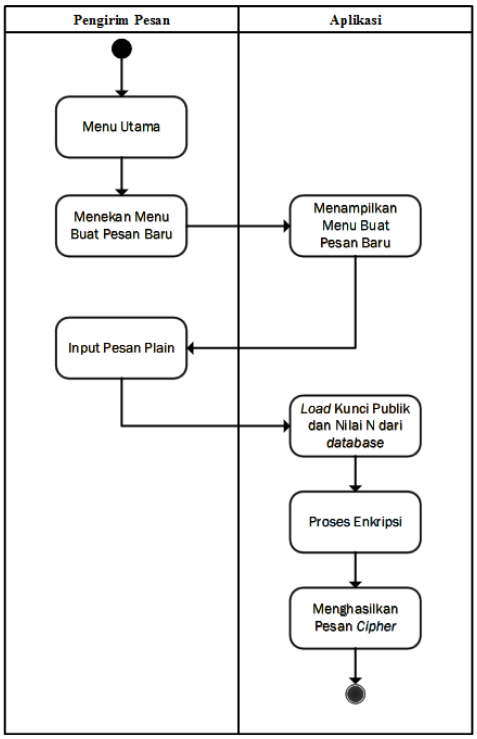
Gbr. 3 *Usecase* Konteks Implementasi Algoritma Luc Pada Aplikasi Keamanan SMS Berbasis Android

1. *Activity* Diagram

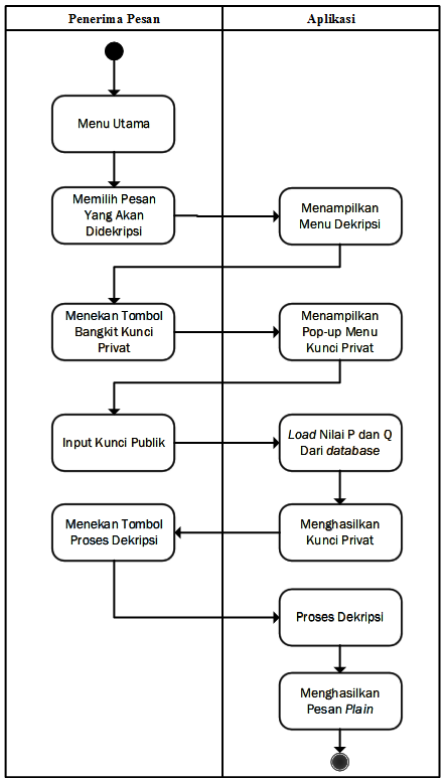
Adapun gambar *activity* diagram dapat dilihat pada Gbr. 4, 5, dan 6 berikut:



Gbr. 4 *Activity* Diagram Pembangkitan Kunci Publik Implementasi Algoritma Luc Pada Aplikasi Keamanan SMS Berbasis Android



Gbr. 5 *Activity* Diagram Proses Enkripsi Implementasi Algoritma Luc Pada Aplikasi Keamanan SMS Berbasis Android



Gbr. 6 *Activity* Diagram Proses Dekripsi Publik Implementasi Algoritma Luc Pada Aplikasi Keamanan SMS Berbasis Android

B. Pengujian Sistem

Pengujian adalah suatu proses pelaksanaan atau pengecekan suatu program dengan tujuan menemukan suatu kesalahan. Dalam penelitian ini, pengujian terbagi menjadi 2 bagian, yaitu pengujian fungsional sistem (*black*

box) dan pengujian algoritma Luc dalam melakukan penyandian pesan.

1. Pengujian Fungsional Sistem

Setelah pembuatan aplikasi telah selesai, maka dilakukan pengujian terhadap aplikasi. Pengujian fungsional sistem ini dilakukan dengan metode *blackbox* untuk mengetahui apakah fungsi-fungsi yang terdapat dalam sistem berjalan dengan baik atau tidak. Hasil pengujian fungsi sistem dapat dilihat pada tabel berikut:

1.1 Pengujian *form* Pembangkitan Kunci Publik

TABEL 1
PENGUJIAN PROSES PEMBANGKITAN KUNCI PUBLIK

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Hasil**
Menekan tombol pembangkitan kunci publik tanpa nilai P, nilai Q, dan nilai N yang belum terisi	Tampil pesan peringatan untuk membangkitkan nilai tersebut.	Menampilkan pesan peringatan bahwa nilai belum dibangkitkan	Baik
Menekan tombol simpan data tanpa nilai P, nilai Q, nilai N, dan kunci publik yang belum terisi	Tampil pesan peringatan untuk membangkitkan nilai P, nilai Q, nilai N, dan kunci publik terlebih dahulu.	Menampilkan pesan peringatan bahwa nilai belum dibangkitkan	Baik
Menekan tombol simpan data tanpa kunci publik yang belum terisi	Tampil pesan peringatan untuk membangkitkan kunci publik.	Menampilkan pesan peringatan bahwa kunci publik belum dibangkitkan	Baik

TABEL 2
PENGUJIAN PROSES PEMBANGKITAN KUNCI PUBLIK (LANJUTAN)

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Hasil**
Menekan tombol pembangkitan kunci publik dengan nilai P, nilai Q, dan nilai N yang sudah terisi	Dapat membangkitkan kunci publik dan menyimpan ke *database*	Proses pembangkitan kunci publik berhasil dan dapat tersimpan ke *database*	Baik

1.2 Pengujian *form* Membuat Pesan Baru

TABEL 3
PENGUJIAN PROSES MEMBUAT PESAN BARU

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Hasil**
Menekan tombol enkripsi tanpa pesan *plain* yang belum terisi	Tampil pesan peringatan untuk mengisi kolom pesan *plain*	Menampilkan peringatan bahwa kolom pesan *plain* belum terisi	Baik
Menekan tombol kirim pesan tanpa nomor kontak dan pesan *cipher* yang belum terisi	Tampil pesan peringatan untuk mengisi nomor kontak dan pesan *cipher*	Menampilkan peringatan bahwa kolom nomor kontak dan pesan *cipher* belum terisi	Baik
Menekan tombol kirim pesan tanpa nomor kontak yang belum terisi	Tampil pesan peringatan untuk mengisi nomor kontak	Menampilkan peringatan bahwa kolom nomor kontak	Baik
Menekan tombol kirim pesan tanpa pesan *cipher* yang belum terisi	Tampil pesan peringatan untuk mengenkripsi pesan	Menampilkan peringatan bahwa pesan belum terenkripsi	Baik

TABEL 4
PENGUJIAN PROSES MEMBUAT PESAN BARU (LANJUTAN)

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Hasil**
Menekan tombol kirim pesan dengan nomor kontak dan pesan *cipher* yang sudah terisi	Dapat mengirimkan pesan *cipher* dan konfirmasi pengiriman kunci publik	Berhasil mengirimkan pesan *cipher* dan kunci publik	Baik

1.3 Pengujian *form* Percakapan

TABEL 5
PENGUJIAN PERCAKAPAN SMS

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Hasil**
Menekan tombol kirim pesan tanpa pesan *plain* yang belum terisi	Tampil pesan peringatan untuk mengisi kolom pesan *plain*	Menampilkan pesan peringatan bahwa kolom pesan *plain* belum terisi	Baik

TABEL 6
PENGUJIAN PERCAKAPAN SMS (LANJUTAN)

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Hasil**
Menekan tombol kirim pesan dengan pesan *plain* yang belum terisi	Dapat terkirim dan pesan konfirmasi untuk mengirim kunci publik	Proses mengirim pesan *cipher* berhasil dan kunci publik dapat terkirim	Baik

1.4 Pengujian *form* Dekripsi Pesan

TABEL 7
PROSES DEKRIPSI

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Hasil**
Menekan tombol proses dekripsi pesan tanpa kunci	Tampil pesan peringatan untuk mengisi kolom kunci privat	Menampilkan pesan peringatan bahwa kolom kunci privat belum terisi	Baik

privat yang belum terisi			

TABEL 8
PROSES DEKRIPSI (LANJUTAN)

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Hasil**
Menekan tombol proses dekripsi pesan dengan kunci privat yang terisi	Dapat mendekripsi pesan *cipher* dan menghasilkan pesan *plain*	Proses mendekripsi pesan *cipher* berhasil dan menghasilkan pesan *plain*	Baik

2. Pengujian Algoritma Luc

Pada pengujian penerapan algoritma Luc terdapat 4 langkah yaitu pembangkitan kunci publik, proses enkripsi, pembangkitan kunci privat, dan proses dekripsi yang penulis deskripsikan sebagai berikut:

1. Hasil dan uji coba penyandian pesan berupa kata "KOTA PALU" menggunakan kunci publik = 13.
   1) Pembangkitan Kunci Publik
      a. Pilih dua bilangan prima acak, misal *P* dan *Q* dimana $P \neq Q$. Dalam pengujian ini sepasang bilangan *P* dan *Q* dibangkitkan dimana bilangan prima *P* = 47 dan *Q* = 241.
      b. Hitung nilai N = *P* x *Q*. Nilai N akan digunakan dalam menghitung modulo pada proses enkripsi dan dekripsi. Maka N = *P* x *Q* = 47 x 241 = 11327.
      c. Hitung semua faktor prima terhadap (*P*-1), (*P*+1), (*Q*-1), dan (*Q*+1). Hasil faktor prima tidak akan dihimpun pada bilangan prima yang ada pada (*P*-1), (*P*+1), (*Q*-1), dan (*Q*+1). Maka,
      (*P* – 1) = (46) = {3, 5, 7, 11, 13, 17, 19, 29, 31, 37, 41, 43}
      (*P* + 1) = (48) = {5, 7, 11 ,13 , 17, 19, 23, 29, 31, 37, 41, 43, 47}
      (*Q* – 1) = (240) = {7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, … , 239}
      (*Q* + 1) = (242) = {3, 5, 7, 13, 17, 19, 23, 29, 31, 37, 41, 43, … , 241}
      d. Pilih salah satu bilangan yang sama secara acak dari 4 buah hasil himpunan yang didapatkan pada poin (c) sebagai kunci publik.

      Bilangan yang sama pada ke-4 himpunan yaitu {7, 13, 17, 19, 29, 31, 37, 41, 43} dan pada pengujian ini yang dipilih dari bilangan tersebut yaitu 13 maka kunci publik = 13.
   2) Proses Enkripsi

Tahap awal pada proses enkripsi adalah mengatur teks menjadi blok-blok yang terdiri dari dua karakter. Jika karakter terakhir tidak mempunyai pasangan, maka ditambahkan karakter spasi. Selanjutnya, setiap karakter dalam blok diubah menjadi nilai ASCII dan dihitung menggunakan persamaan $c_i = V_e\ (m_i, 1)$ mod N dimana *e* = 13, m_i adalah nilai ASCII tiap blok, N = 11327 dan c_i adalah hasil enkripsi tiap blok. Pada pengujian ini, penulis menggunakan sampel data kata "KOTA PALU", apabila dipisahkan dalam blok maka teks berubah menjadi "KO", "TA", "[spasi]P", "AL", dan "U[spasi". Selanjutnya adalah merubah tiap blok dalam bentuk ASCII, maka didapatkan bilangan ASCII KO = 7579, TA = 8465, [spasi]P = 3280, AL = 6576, dan U[spasi] = 8532. Dengan menggunakan kunci publik yang dibangkitkan pada tahap sebelumnya *e* = 13, maka selanjutnya adalah pembangkitan rantai lucas k[*x*] untuk proses enkripsi dapat dilihat pada tabel 9.

TABEL 9
PEMBANGKITAN RANTAI LUCAS UNTUK PROSES ENKRIPSI

x	k [*x*]	*e*
1	1	*e*-1 = 12
2	0	12/2 = 6
3	0	6/2 = 3
4	1	3-1 = 2
5	0	2/2 = 1

Didapatkan k[*x*] = {1, 0, 0, 1, 0} dimana k[*x*] adalah rantai lucas. Kemudian dilakukan *backward* maka k[*x*] = {0, 1, 0, 0, 1}. Proses enkripsi berdasarkan dengan nilai k[*x*] yang ada dapat dilihat pada tabel 10.

TABEL 10
PROSES ENKRIPSI KARAKTER KO DAN TA

Enkripsi karakter KO		Enkripsi karakter TA	
k [*x*]	Hasil	k [*x*]	Hasil
0	2022	0	1621
1	3055	1	7630
0	10902	0	7445
0	10718	0	5012
1	8002	1	6183

Pada tabel 10 di atas didapatkan hasil dari enkripsi yaitu ASCII 8002 dimana 8002 = PSTX(STX adalah karakter ASCII yang tidak dapat terbaca) dan 6183 = =S maka hasil enkripsi kata KOTA = P=S.

Adapun untuk contoh perhitungan dengan menggunakan persamaan 9, 10, dan 11 sebagai berikut.

V_n = M dan V_j = 2

untuk k[*x*] = 0 adalah $V_{2n} = (V_n)^2 - 2$ (mod N)

$V_{2n-1} = V_n V_j$ – M (mod N)

$V_n = V_{2n}$;

$V_j = V_{2n-1}$;

untuk k[*x*] = 1 adalah $V_{2n+1} = MV_n - V_j$ (mod N)

$V_n = V_{2n+1}$;

$V_j = V_{2n}$;

Dari persamaan di atas maka hasil tabel 4.11 sebagai berikut.

Iterasi - k[*x*] = 0

$V_{2n} = 7579^2 - 2$ (mod 11327)
= 57.441.241 – 2 mod 11327
= 57.441.239 mod 11327
= 2022

V_{2n-1} = 7579 x 2 – 7579 (mod 11327)
= 15.158 – 7579 mod 11327

= 7579 mod 11327
= 7579
V_n = 2022
V_j = 7579
Iterasi - k[x] = 1
V_{2n+1} = 7579 x 2022 – 7579 mod 11327
= 15.324.738 – 7579 mod 11327
= 15.317.159 mod 11327
= 3055

V_n = 3055
V_j = 2022
Iterasi - k[x] = 0
V_{2n} = 3055^2 – 2 (mod 11327)
= 9.333.025 – 2 mod 11327
= 9.333.023 mod 11327
= 10902
V_{2n-1} = 3055 x 2022 – 7579 (mod 11327)
= 6.177.210 – 7579 mod 11327
= 6.169.631 mod 11327
= 7743
V_n = 10902
V_j = 7743
Iterasi - k[x] = 0
V_{2n} = 10902^2 – 2 (mod 11327)
= 118.853.604 – 2 mod 11327
= 118.853.602 mod 11327
= 10718
V_{2n-1} = 10902 x 7743 – 7579 (mod 11327)
= 84.414.186 – 7579 mod 11327
= 84.406.607 mod 11327
= 9130
V_n = 10718
V_j = 9130
Iterasi - k[x] = 1
V_{2n+1} = 7579 x 10718 – 9130 mod 11327
= 81.231.722 – 9130 mod 11327
= 81.222.592 mod 11327
= 8002

Pada tabel berikutnya juga menggunakan perhitungan dari persamaan seperti di atas untuk mendapatkan hasil enkripsi maupun dekripsi.

TABEL 11
PROSES ENKRIPSI KARAKTER [SPASI]P DAN AL

Enkripsi karakter [spasi]P		Enkripsi karakter AL	
k [x]	Hasil	k [x]	Hasil
0	9075	0	8615
1	6691	1	10664
0	5175	0	9141
0	3595	0	9927
1	10133	1	9981

Dari tabel 11 di atas didapatkan hasil dari enkripsi yaitu ASCII 10133 dimana 10133 = e! dan 9981 = cQ maka hasil enkripsi kata [spasi]PAL = e!cQ.

TABEL 12
PROSES ENKRIPSI KARAKTER U[SPASI]

Enkripsi karakter U[spasi]	
k [x]	Hasil
0	7720
1	3330
0	11092
0	9915
1	9704

Pada tabel 4.13 di atas didapatkan hasil dari enkripsi yaitu ASCII 9704 dimana 9704 = aEOT(EOT adalah karakter ASCII yang tidak dapat terbaca) maka hasil enkripsi kata U[spasi] = a .

Dari hasil keseluruhan enkripsi pada tabel 4.11, 4.12, dan 4.13 didapatkan nilai ASCII dari gabungan keseluruhan bloknya yaitu 800261831013399819704 = P=Se!cQa

3) Pembangkitan Kunci Privat

Kemudian untuk melakukan proses dekripsi penerima pesan harus membangkitkan kunci privat ketika telah menerima pesan *cipher* dan kunci publik kemudian aplikasi akan me*load* nilai P dan nilai Q dari *database*. Untuk melakukan proses perhitungannya dimana $D = C^2 - 4$ dan pada tabel 4.11, 4.12, dan 4.13 memiliki nilai ASCII masing-masing 8002, 6183, 10133, 9981, dan 9704 maka:

a. $D = (8002)^2 - 4$. Simbol legendre untuk $\frac{D}{P}$ adalah $\frac{64032000}{47}$ = -1.
dimana 64032000 mod 47 = 46. Kemudian hasil mod dibentuk ke model pecahan $\frac{46}{47}$. karena hasil mod bernilai genap dan lebih kecil daripada *P*, harus dibagi 2 maka $\frac{46}{47} : \frac{2}{1} = \frac{23}{47}$. Oleh karena itu, dari penyederhanaan tersebut didapatkan simbol legendre $\frac{23}{47}$ adalah -1.
sedangkan simbol legendre untuk $\frac{D}{Q}$ adalah $\frac{64032000}{241}$ = -1.
dimana 64032000 mod 241 = 228. Kemudian $\frac{228}{241} : \frac{2}{1} = \frac{114}{241}$ setelah itu $\frac{114}{241} : \frac{2}{1} = \frac{57}{241}$. Kemudian apabila pembilang lebih kecil dan bernilai ganjil daripada *Q*, maka 241 mod 57 = 13. Oleh karena itu, dari penyederhanaan tersebut didapatkan simbol legendre $\frac{13}{57}$ adalah -1.
Selanjutnya adalah mencari LCM ($P - \frac{D}{P}$) , ($Q - \frac{D}{Q}$) dimana *r* = LCM (P – 1),(Q - 1) maka:
r = LCM (47 – (-1)), (241 – (-1))
= LCM (48), (242)
= 5808
Kemudian mencari nilai *d* sebagai berikut.
$ed \equiv 1 \bmod r$
$d = \frac{1+(k.r)}{e}$

$d = \frac{1+(9x\,5808)}{13}$
$d = 4021$
maka kunci privat *d* adalah 4021.

b. $D = (6183)^2 - 4$. Simbol legendre untuk $\frac{D}{P}$ adalah $\frac{38229485}{47} = 1$.
dimana 38229485 mod 47 = 14. maka dimodelkan menjadi $\frac{14}{47}$. Oleh karena itu, dari penyederhanaan tersebut didapatkan simbol legendre $\frac{14}{47}$ adalah 1.
sedangkan simbol legendre untuk $\frac{D}{Q}$ adalah $\frac{38229485}{241}$ = -1.
dimana 38229485 mod 241 = 137. maka dimodelkan menjadi $\frac{137}{241}$ kemudian 241 mod 137 = 104. dimodelkan menjadi $\frac{104}{137} : \frac{2}{1} = \frac{52}{137} : \frac{2}{1} = \frac{26}{137} : \frac{2}{1} = \frac{13}{137}$ kemudian 137 mod 13 = 7. Oleh karena itu, dari penyederhanaan tersebut didapatkan simbol legendre $\frac{7}{13}$ adalah -1.
Selanjutnya adalah mencari LCM ($P - \frac{D}{P}$) , ($Q - \frac{D}{Q}$) dimana *r* = LCM (P – 1),(Q - 1)
maka:
r = LCM (47 – 1), (241 – (-1))
= LCM (46), (242)
= 5566
Kemudian mencari nilai *d* sebagai berikut.
$ed \equiv 1 \bmod r$
$d = \frac{1+(k.r)}{e}$
$d = \frac{1+(6x\,5566)}{13}$
$d = 2569$
maka kunci privat *d* adalah 2569.

c. $D = (10133)^2 - 4$. Simbol legendre untuk $\frac{D}{P}$ adalah $\frac{102677685}{47} = 1$.
dimana 102677685 mod 47 = 28. maka dimodelkan menjadi $\frac{28}{47}$. Oleh karena itu, dari penyederhanaan tersebut didapatkan simbol legendre $\frac{28}{47}$ adalah 1.
sedangkan simbol legendre untuk $\frac{D}{Q}$ adalah $\frac{102677685}{241}$ = -1.
dimana 102677685 mod 241 = 117. kemudian 241 mod 117 = 7.
117 mod 7 = 5. maka dimodelkan $\frac{5}{7}$. Oleh karena itu, dari penyederhanaan tersebut didapatkan simbol legendre $\frac{5}{7}$ adalah -1. (hasil dapat dilihat pada lampiran 5).
Selanjutnya adalah mencari LCM ($P - \frac{D}{P}$) , ($Q - \frac{D}{Q}$) dimana *r* = LCM (P – 1),(Q - 1)
maka:
r = LCM (47 – 1), (241 – (-1))
= LCM (46), (242)
= 5566
Kemudian mencari nilai *d* sebagai berikut.
$ed \equiv 1 \bmod r$
$d = \frac{1+(k.r)}{e}$
$d = \frac{1+(6x\,5566)}{13}$
$d = 2569$
maka kunci privat *d* adalah 2569.

d. $D = (9981)^2 - 4$. Simbol legendre untuk $\frac{D}{P}$ adalah $\frac{99620357}{47} = 1$.
dimana 99620357 mod 47 = 3. maka dimodelkan $\frac{3}{47}$. Oleh karena itu, dari penyederhanaan tersebut didapatkan simbol legendre $\frac{3}{47}$ adalah 1.
sedangkan simbol legendre untuk $\frac{D}{Q}$ adalah $\frac{102677685}{241}$ = -1.
dimana 99620357 mod 241 = 115. kemudian 241 mod 115 = 11. kemudian 115 mod 11 = 5. maka dimodelkan $\frac{5}{7}$. Oleh karena itu, dari penyederhanaan tersebut didapatkan simbol legendre $\frac{5}{7}$ adalah -1.
Selanjutnya adalah mencari LCM ($P - \frac{D}{P}$) , ($Q - \frac{D}{Q}$) dimana *r* = LCM (P – 1),(Q - 1)
maka:
r = LCM (47 – 1), (241 – (-1))
= LCM (46), (242)
= 5566
Kemudian mencari nilai *d* sebagai berikut.
$ed \equiv 1 \bmod r$
$d = \frac{1+(k.r)}{e}$
$d = \frac{1+(6x\,5566)}{13}$
$d = 2569$
maka kunci privat *d* adalah 2569.

e. $D = (9704)^2 - 4$. Simbol legendre untuk $\frac{D}{P}$ adalah $\frac{94167612}{47} = -1$.
dimana 94167612 mod 47 = 10. maka dimodelkan $\frac{10}{47}$. Oleh karena itu, dari penyederhanaan tersebut didapatkan simbol legendre $\frac{10}{47}$ adalah -1.
sedangkan simbol legendre untuk $\frac{D}{Q}$ adalah $\frac{94167612}{241}$ = 1.
dimana 94167612 mod 241 = 236. kemudian $\frac{236}{241} : \frac{2}{1} = \frac{118}{241} : \frac{2}{1} = \frac{59}{241}$
kemudian 241 mod 59 = 5. maka dimodelkan $\frac{5}{59}$. Oleh karena itu, dari penyederhanaan tersebut didapatkan simbol legendre $\frac{5}{59}$ adalah 1.

Selanjutnya adalah mencari LCM ($P - \frac{D}{P}$) , ($Q - \frac{D}{Q}$) dimana r = LCM (P – 1),(Q - 1) maka:

r = LCM (47 – (-1)), (241 – 1)
= LCM (48), (240)
= 240

Kemudian mencari nilai *d* sebagai berikut.

$ed \equiv 1 \bmod r$

$d = \frac{1+(k.r)}{e}$

$d = \frac{1+(2x\ 240)}{13}$

$d = 37$

maka kunci privat *d* adalah 37.

4) Proses Dekripsi

Tahap awal pada proses dekripsi adalah mengatur teks menjadi blok-blok yang terdiri dari dua karakter. Pada tabel 4.11, 4.12, dan 4.13 di atas didapatkan pesan *cipher* PSTX (STX adalah karakter ASCII yang tidak dapat terbaca), =S, e!, cQ, dan aEOT(EOT adalah karakter ASCII yang tidak dapat terbaca) yang dimana mempunyai nilai ASCII masing-masing = 8002, 6183, 10133, 9981, dan 9704. maka proses dekripsi dilakukan dengan persamaan $m_i = V_d$ $(c_i, 1) \bmod N$ Adapun penjelasan proses dekripsi tiap bloknya sebagai berikut.

a. Karakter PSTX (STX adalah karakter ASCII yang tidak dapat terbaca)

Kunci privat pada karakter ini yaitu d = 4021, maka selanjutnya membangkitkan rantai lucas k[x] untuk proses dekripsi dapat dilihat pada tabel 14.

TABEL 13
PEMBANGKITAN RANTAI LUCAS UNTUK KARAKTER PSTX

x	k [x]	d
1	1	d-1 = 4020
2	0	4020/2 = 2010
3	0	2010/2 = 1005
4	1	1005-1 = 1004
5	0	1004/2 = 502
6	0	502/2 = 251
7	1	251-1 = 250
8	0	250/2 = 125
9	1	125-1 = 124
10	0	124/2 = 62
11	0	62/2 = 31
12	1	31-1 = 30
13	0	30/2 = 15
14	1	15-1 = 14
15	0	14/2 = 7
16	1	7-1 = 6
17	0	6/2 = 3
18	1	3-1 = 2
19	0	2/2 = 1

Didapatkan k[x] = {1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1} dimana k[x] adalah rantai lucas. Kemudian dilakukan *backward* maka k[x] = {0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1}. Proses dekripsi berdasarkan dengan nilai k[x] yang ada dapat dilihat pada tabel 13.

TABEL 14
PROSES DEKRIPSI KARAKTER PSTX

Dekripsi karakter PSTX	
k [x]	Hasil
0	471
1	376
0	5450
1	2761
0	48
1	1786
0	6907
1	6944
0	95
0	9023
1	2291
0	4278
1	10446
0	5923
0	2208
1	3055
0	10902
0	10718
1	7579

Pada tabel 4.15 di atas didapatkan hasil dari dekripsi yaitu ASCII 7579 dimana 7579 = KO

b. Karakter =S

Kunci privat pada karakter ini yaitu d = 2569, maka selanjutnya membangkitkan rantai lucas k[x] untuk proses dekripsi dapat dilihat pada tabel 15.

TABEL 15
PEMBANGKITAN RANTAI LUCAS UNTUK KARAKTER =S

x	k [x]	d
1	1	d-1 = 2568
2	0	2568/2 = 1284
3	0	1284/2 = 642
4	0	642/2 = 321
5	1	321-1 = 320
6	0	320/2 = 160
7	0	160/2 = 80
8	0	80/2 = 40
9	0	40/2 = 20
10	0	20/2 = 10
11	0	10/2 = 5
12	1	5-1 = 4
13	0	4/2 = 2
14	0	2/2 = 1

Didapatkan k[x] = {1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0} dimana k[x] adalah rantai lucas. Kemudian dilakukan *backward* maka k[x] = {0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1}. Proses dekripsi berdasarkan dengan nilai k[x] yang ada dapat dilihat pada tabel 16.

TABEL 16
PROSES DEKRIPSI KARAKTER =S

Dekripsi karakter =S	
k [x]	Hasil
0	862
0	6787
1	8940
0	286
0	2505
0	11192
0	6896
0	4068
0	11202
1	3034

0	7630
0	7445
0	5012
1	8465

Pada tabel 15 di atas didapatkan hasil dari dekripsi yaitu ASCII 8465 dimana 8465 = TA

c. Karakter e!

Kunci privat pada karakter ini yaitu d = 2569, maka selanjutnya membangkitkan rantai lucas k[x] untuk proses dekripsi dapat dilihat pada tabel 17.

TABEL 17
PEMBANGKITAN RANTAI LUCAS UNTUK KARAKTER E!

x	k [x]	d
1	1	d-1 = 2568
2	0	2568/2 = 1284
3	0	1284/2 = 642
4	0	642/2 = 321
5	1	321-1 = 320
6	0	320/2 = 160
7	0	160/2 = 80
8	0	80/2 = 40
9	0	40/2 = 20
10	0	20/2 = 10
11	0	10/2 = 5
12	1	5-1 = 4
13	0	4/2 = 2
14	0	2/2 = 1

Didapatkan k[x] = {1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0} dimana k[x] adalah rantai lucas. Kemudian dilakukan *backward* maka k[x] = {0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1}. Proses dekripsi berdasarkan dengan nilai k[x] yang ada dapat dilihat pada tabel 18.

TABEL 18
PROSES DEKRIPSI KARAKTER E!

Dekripsi karakter e!	
k [x]	Hasil
0	9759
0	663
1	8164
0	2826
0	739
0	2423
0	3541
0	11017
0	5482
1	404
0	4636
0	5175
0	3595
1	3280

Pada tabel 17 di atas didapatkan hasil dari dekripsi yaitu ASCII 3280 dimana 3280 = [spasi]P

d. Karakter cQ

Kunci privat pada karakter ini yaitu d = 2569, maka selanjutnya membangkitkan rantai lucas k[x] untuk proses dekripsi dapat dilihat pada tabel 19.

TABEL 19
PEMBANGKITAN RANTAI LUCAS UNTUK KARAKTER CQ

x	k [x]	d
1	1	d-1 = 2568
2	0	2568/2 = 1284
3	0	1284/2 = 642
4	0	642/2 = 321
5	1	321-1 = 320
6	0	320/2 = 160
7	0	160/2 = 80
8	0	80/2 = 40
9	0	40/2 = 20
10	0	20/2 = 10
11	0	10/2 = 5
12	1	5-1 = 4
13	0	4/2 = 2
14	0	2/2 = 1

Didapatkan k[x] = {1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0} dimana k[x] adalah rantai lucas. Kemudian dilakukan *backward* maka k[x] = {0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1}. Proses dekripsi berdasarkan dengan nilai k[x] yang ada dapat dilihat pada tabel 20.

TABEL 20
PROSES DEKRIPSI KARAKTER CQ

Dekripsi karakter cQ	
k [x]	Hasil
0	10721
0	4470
1	511
0	598
0	6465
0	10920
0	7069
0	7362
0	10674
1	1568
0	663
0	9141
0	9927
1	6576

Pada tabel 19 di atas didapatkan hasil dari dekripsi yaitu ASCII 6576 dimana 6576 = AL

e. Karakter aEOT (EOT adalah karakter ASCII yang tidak dapat terbaca)

Kunci privat pada karakter ini yaitu d = 37, maka selanjutnya membangkitkan rantai lucas k[x] untuk proses dekripsi dapat dilihat pada tabel 21.

TABEL 21
PEMBANGKITAN RANTAI LUCAS UNTUK KARAKTER AEOT

x	k [x]	d
1	1	d-1 = 36
2	0	36/2 = 18
3	0	18/2 = 9
4	1	9-1 = 8
5	0	8/2 = 4
6	0	4/2 = 2
7	0	2/2 = 1

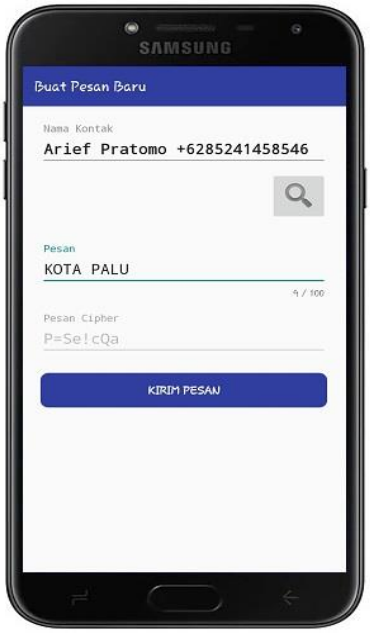
Didapatkan k[x] = {1, 0, 0, 1, 0, 0, 0} dimana k[x] adalah rantai lucas. Kemudian dilakukan *backward* maka k[x] = {0, 0, 0, 1, 0, 0, 1}. Proses dekripsi berdasarkan dengan nilai k[x] yang ada dapat dilihat pada tabel 22.

TABEL 22
PROSES DEKRIPSI KARAKTER AEOT

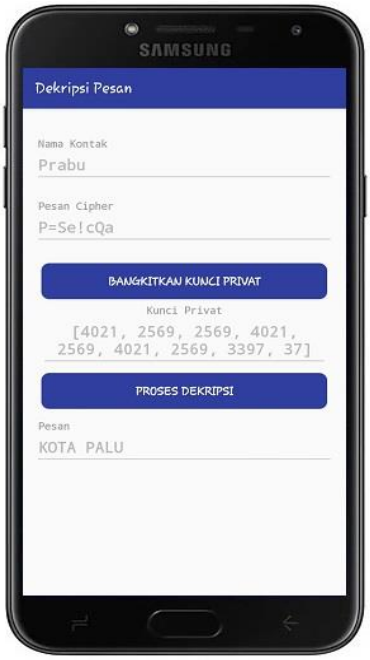
Dekripsi karakter aEOT	
k [x]	Hasil
0	6263
0	11093
0	9446
1	3659
0	11092
0	9915
1	8532

Pada tabel 4.23 di atas didapatkan hasil dari dekripsi yaitu ASCII 8532 dimana 8532 = U[spasi].
Dari hasil keseluruhan dekripsi pada tabel 4.15, 4.17, 4.19, 4.21, dan 4.23 didapatkan nilai ASCII dari gabungan keseluruhan bloknya yaitu 75798465 328065768532 = KOTA PALU.

Pada pengujian algoritma Luc penulis melakukan proses perhitungan manual yang bermaksud untuk mencocokan dengan hasil perhitungan yang dilukukan oleh aplikasi, dan hasil yang didapatkan perhitungan manual maupun perhitungan aplikasi pada pengujian ini menghasilkan hasil yang sama. Untuk melihat hasilnya pada aplikasi dapat dilihat pada gambar 7 dan 8 berikut.



Gbr. 7 Hasil Uji Coba Enkripsi Pesan *Plain* "KOTA PALU" Menggunakan Kunci Publik 13



Gbr. 8 Hasil Uji Coba Dekripsi Pesan *Cipher* "P=Se!cQa"

VI. KESIMPULAN DAN SARAN

A. *Kesimpulan*

Berdasarkan hasil pengujian dan analisis implementasi algoritma Luc pada aplikasi keamanan SMS berbasis android yang telah dilakukan oleh penulis, dapat disimpulkan bahwa :

1. Algoritma Luc dapat diimplementasi pada aplikasi keamanan SMS berbasis android untuk untuk keamanan dalam berkomunikasi.
2. Algoritma Luc dapat melakukan penyandian pesan pada aplikasi keamanan SMS berbasis android untuk meminimalisir kejahatan penyadapan informasi.
3. Hasil pengujian sampel pesan "KOTA PALU" pada aplikasi ini menggunakan kunci publik 13 yang menghasilkan pesan *cipher* P=Se!cQa dan proses dekripsi berjalan dengan benar dan sesuai dengan perhitungan manual.

B. *Saran*

Dari hasil penelitian ini masih terdapat beberapa kekurangan, sehingga masih diperlukan perbaikan untuk pengembangan lebih lanjut, diantaranya:

1. Diharapkan pada penelitian berikutnya dapat mengkaji lebih dalam tentang kriptografi algoritma Luc ini sehingga dapat mengembangkan penginputan pesan untuk huruf kecil agar terenkripsi dan dekripsi dengan baik.
2. Menambahkan fitur seperti hapus pesan, *copypaste* pesan, dan pencarian langsung nomor kontak seperti aplikasi SMS pada umumnya.
3. Diharapkan pada penelitian berikutnya dapat mengimplementasikan algoritma lainnya seperti ElGamal, McEliece, dll.

DAFTAR PUSTAKA

[1] Anggraini, N. 2016. Implementasi Algoritma Luc Pada Pengamanan Citra Digital Berbasis Desktop, *Skripsi* Program Studi Ilmu Komputer, Universitas Sumatera Utara, Medan.
[2] Alpha, C. D. 2017. Kriptografi Visual Dengan Implementasi Algoritma LUC Pada Citra Berwarna, *Skripsi* Program Studi Teknik Elektro, Universitas Kristen Maranatha, Bandung.
[3] Yusfrizal. 2015. Penerapan Algoritma RC6 Untuk Perancangan Aplikasi Pengamanan SMS Pada Mobile Device Berbasis Android, *Skripsi* Jurusan Teknik Informatika, Universitas Potensi Utama, Medan.
[4] Kusumawati, B. 2018. Kriptosistem Kunci Publik Luc Serta Implementasinya Pada Program Lazarus, *Skripsi* Program Studi Pendidikan Matematika, Universitas Sanata Dharma, Yogyakarta.
[5] Syamsinar. 2017. Implementasi Kombinasi Algoritma Asimetris Rivest Shamir Adleman Dan Algoritma Simetris AES Pada Aplikasi Pesan Singkat, *Skripsi* Jurusan Teknik Informatika, Universitas Islam Negeri Makassar.
[6] Ramadani, D. 2018. Implementasi Algoritma LUC Dalam Penyandian Teks, *Jurnal Skripsi*, STMIK Budi Darma, Medan.