SISTEM KEAMANAN PESAN PADA ANDROID GINGERBREAD (2.3.4) DENGAN ALGORITMA LUC

SKRIPSI

Oleh
Arif Fajar Irawan
NIM. 051810101009

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2013

SISTEM KEAMANAN PESAN PADA ANDROID GINGERBREAD (2.3.4) DENGAN ALGORITMA LUC

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk
menyelesaikan pendidikan di Program Studi Matematika (S1)
dan mencapai gelar Sarjana Sains

Oleh

Arif Fajar Irawan
NIM 051810101009

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2013

PERSEMBAHAN

Skripsi ini saya persembahkan untuk;

1) Allah SWT yang dengan tuntunan serta limpahan kasih-Nya mengajariku arti dan kekuatan dalam hidup;
2) ayahanda Askarudin dan ibunda Kasmaslikah tercinta, beliau berdua segalanya bagiku, terimakasih atas dorongan, semangat dan doanya;
3) Bapak dan ibu Guru di TK Al-Muawanah Banyuwangi, MI Al-Muawanah I Banyuwangi, MTs Al-Huda Banyuwangi, MAN 2 Jember dan Universitas Jember yang telah memberikan ilmu dan membimbing dengan penuh kesabaran;
4) Almamater Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember semoga skripsi ini bermanfaat dan dapat menambah referensi ilmu pengetahuan khususnya di bidang Matematika dan Teknologi Informatika.

MOTTO

"Janganlah kamu berputus asa dari rahmat ALLAH"[1]

Stop Dreaming Start Action

"Sesungguhnya, Aku mengingatkan kepadamu supaya kamu tidak termasuk orang-orang yang tidak berpengetahuan"[2]

1. QS. Az-Zumar : 53
2. QS. Hud : 46

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : **Arif Fajar Irawan**

NIM : **051810101009**

menyatakan dengan sesungguhnya bahwa karya tulis ilmiah berjudul: ” *Sistem Keamanan Pesan Pada Android GingerBread (2.3.4) Dengan Algoritma Luc”* adalah benar-benar hasil karya sendiri, kecuali jika disebutkan sumbernya dan belum pernah diajukan pada institusi manapun, serta bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, Mei 2013

Yang menyatakan,

Arif Fajar Irawan

NIM 051810101009

SKRIPSI

SISTEM KEAMANAN PESAN PADA ANDROID GINGERBREAD (2.3.4) DENGAN ALGORITMA LUC

Oleh

Arif Fajar Irawan

NIM 051810101009

Pembimbing

Dosen Pembimbing Utama	: Drs. Rusli Hidayat M.Sc
Dosen Pembimbing Anggota	: Bagus Juliyanto S.Si

PENGESAHAN

Skripsi berjudul “Sistem Keamanan Pesan Pada Android GingerBread (2.3.4) dengan Algoritma Luc” telah diuji dan disahkan oleh Fakultas Matematika dan Ilmu Pengetahuan Alam pada :

Hari :

Tanggal :

Tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam

Tim Penguji

Ketua,	Sekretaris,
Drs. Rusli Hidayat M.Sc NIP 196610121993031001	Bagus Juliyanto S.Si NIP 198007022003121001
Penguji I,	Penguji II,
Kusbudiono S.Si. M.Si NIP 197704302005011001	Prof. Drs. I Made Tirta, M.Sc., Ph.D NIP 195912201985031002

Mengesahkan

Dekan,

Prof. Drs. Kusno, DEA, Ph.D.
NIP 196101081986021001

RINGKASAN

Sistem Keamanan Pesan Pada Android GingerBread (2.3.4) dengan Algoritma Luc; Arif Fajar Irawan, 051810101009; 2013; 38 halaman; Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Algoritma Luc merupakan salah satu algoritma kriptografi asimetris yang menggunakan dua kunci yang berbeda pada kriptosistemnya. Tujuan dari skripsi ini adalah meningkatkan keamanan pesan pada telepon seluler berbasis Android.

Penulisan skripsi ini dilakukan dalam dua tahap. Tahap pertama yaitu indentifikasi masalah. Langkah-langkah yang dilakukan pada tahap ini adalah mencari referensi yang berkaitan dengan keamanan pesan, fungsi Lucas, algoritma Luc, menyusun algoritma dan *software* pendukung Android. Dalam algoritma Luc, proses awal adalah menentukan dua bilangan prima p dan q kemudian dihitung $N = p.q$, selanjutnya adalah menentukan kunci public e dengan memilih salah satu bilangan yang relative prima terhadap (p-1), (p+1), (q-1), (q+1). Objek penelitian ini adalah telepon seluler berbasis Android, dimana Android merupakan teknologi modern yang menyerupai komputer sehingga mampu melakukan penghitungan rumit. Tahap kedua adalah pembuatan program serta uji program, dalam skripsi ini pembuatan program memanfaatkan program *Eclipse Juno* yang berbasis bahasa pemrograman Java. Pengujian program dilakukan dengan mengirim pesan teks ke telepon seluler lain (GingerBread). Proses enkripsi dilakukan pada saat pengiriman pesan, pesan yang telah diketik dalam *TextBox* di-enkripsi dengan menggunakan kunci public yang telah disepakati. Pesan yang diterima masuk dalam *Inbox* pesan, dengan isi pesan yang telah ter-enkripsi, untuk dapat membaca pesan tersebut dilakukan proses dekripsi dengan membangkitkan kunci privat terlebih dahulu.

Dari hasil enkripsi maupun dekripsi dapat dinyatakan bahwa algoritma Luc dapat diimplementasikan dalam bidang telekomunikasi terutama pada telepon seluler berbasis Android, hal ini disebabkan karena algoritma Luc menggunakan pembagian modulo pada setiap langkah dan menghasilkan nilai yang tidak terlalu besar, sehingga proses enkripsi maupun dekripsi dapat dilakukan dengan cepat.

PRAKATA

Syukur alhamdulillah penulis panjatkan ke hadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga skripsi yang berjudul "*Sistem Keamanan Pesan Pada Android GingerBread (2.3.4) Dengan Algoritma Luc"* dapat diselesaikan. Skripsi ini disusun untuk memenuhi salah satu syarat dalam menyelesaikan pendidikan strata satu (S1) pada Fakultas Matematika dan Ilmu Pengetahuan Alam.

Skripsi ini tidak mungkin terwujud tanpa adanya bantuan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terimakasih kepada;

1. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember beserta staff dan karyawan;
2. Drs. Rusli Hidayat M.Sc selaku Dosen Pembimbing Utama dan Bagus Juliyanto S.Si selaku Dosen Pembimbing Anggota serta Anggota yang dengan sabar meluangkan waktu, tenaga, pikiran serta perhatian dalam penulisan skripsi ini;
3. Kusbudiono S.Si. M.Si selaku Dosen Penguji I dan Prof. Drs. I Made Tirta, M.sc., Ph.D selaku Dosen Penguji II yang telah banyak memberikan saran dan kritik membangun kepada penulis;
4. ibunda dan ayahanda tercinta serta kakak tersayang atas doa, dukungan, cinta, kasih sayang, dan kesabaran selama ini hingga penyusunan skripsi;
5. teman senasib dan seperjuanganku Tri saktika Aji, Fiqih Maulana Yusuf, Cintya Carolina, Hujjatul Islam Al Wafi, Bilal el Bizarroby, Dani Catur Prasetya dan seluruh anggota UKMS Titik atas bantuan dan kerjasama serta dukungannya selama ini.....KEEP OUR SPIRIT
6. teman-teman angkatan 2004 dan adik – adik angkatan terimasih atas persahabatannya selama ini.

7. teman seperjuangan di kost-an Jawa VII dan Karimata 36 B, terimakasih atas kebersamaan, persahabatan, dukungan dan semangat yang diberikan selama ini.
8. Miranti Puspitasari yang memberikan banyak inspirasi
9. semua pihak yang tidak dapat disebutkan satu per satu.

Penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga tulisan ini dapat bermanfaat.

Jember, Maret 2013

Penulis

DAFTAR ISI

Halaman

DAFTAR TABEL

DAFTAR GAMBAR

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Penggunaan teknologi telepon genggam (*handphone*) sebagai alat telekomunikasi pada saat ini telah mengubah cara pandang masyarakat dalam berkomunikasi. Telepon genggam mempunyai beberapa fungsi komunikasi yang dapat digunakan antara lain, *video Call*, *SMS, MMS, Chatting,* Internet, dan lain-lain. Berkembangnya teknologi telepon genggam dapat dilihat dengan munculnya berbagai sistem operasi yang lengkap layaknya komputer, diantaranya adalah Android. Android adalah sebuah sistem operasi untuk perangkat telepon yang berbasis linux yang mencakup sistem operasi, *middleware*, aplikasi dan menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Android berkembang pesat karena mempunyai *platform* yang sangat lengkap baik dalam system operasi, Aplikasi dan Tool Pengembangannya, Market aplikasi serta mendapat dukungan yang sangat tinggi dari komunitas *Open Source* di dunia (Safaat, 2012).

Meskipun Android memiliki fitur yang lengkap, namun layanan *SMS* (*Short Message Service*) sebagai layanan pertukaran informasi atau pesan pendek menjadi komunikasi favorit karena saat ini semua telepon genggam memiliki layanan ini dan yang paling penting adalah biaya *SMS* relatif murah. Namun demikian *SMS* tidak menjamin integritas dan keamanan pesan yang disampaikan. Pesan yang bersifat personal atau rahasia tidak dijamin sampai ke penerima tanpa diketahui informasinya oleh pihak yang tidak bertanggung-jawab. Beberapa resiko yang dapat mengancam keamanan pesan pada layanan *SMS* antara lain *SMS Spoofing, SMS Snooping*, dan *SMS Interception*. *SMS Spoofing* merupakan pengiriman sms di mana nomor pengirim yang tertera bukanlah nomer pengirim yang sebenarnya, masalah berikutnya

adalah *SMS Snooping* lebih sering terjadi karena kelalaian pengguna telepon seluler. Contohnya ketika seseorang meminjamkan telepon selulernya pada orang lain untuk menggunakan telepon selulernya. Pada saat itu orang tersebut dapat dengan sengaja atau tidak membuka isi pesan yang ada pada *inbox SMS*. Celah keamanan terbesar pada layanan komunikasi *SMS* adalah pada saat *SMS* tersebut sedang dikirim melalui jaringan *SMS* tersebut. *SMS* bekerja pada jaringan nirkabel yang memungkinkan terjadinya pencurian isi pesan *SMS* ketika dalam proses transmisi dari pengirim ke penerima, kasus ini disebut *SMS interception.*

Dengan adanya beberapa keterangan diatas maka dibutuhkan sebuah sistem keamanan pada layanan *SMS* yang mampu menjaga integritas dan keamanan isi pesan untuk menutupi celah keamanan *SMS* (terutama untuk *SMS Snooping, SMS Intercept* dan campur tangan operator). Schneier (1996) menyatakan, "*cryptography that will stop your kid sister from reading your files, and cryptography that will stop major governments from reading your files*". pernyataan tersebut mengandung arti bahwa kriptografi dapat mencegah kebocoran informasi dari pihak yang tidak berkepentingan atau tidak berhak atas informasi tersebut bahkan oleh pemerintah pusat sekalipun.

Kriptografi adalah seni untuk mengamankan informasi dengan menggunakan teknik penyandian. Proses penyandian informasi asli (*plainteks*) yang menghasilkan informasi yang tersandikan (*chiperteks*) disebut *enkripsi*, sedangkan proses menguraikan chiperteks menjadi informasi asli disebut *dekripsi*. Saat ini telah banyak metode kriptografi yang muncul, salah satunya adalah Algoritma Luc, algoritma tersebut menggunakan dua buah kunci yaitu kunci umum (untuk melakukan enkripsi) dan kunci rahasia (untuk melakukan dekripsi). Operasi pada Algoritma Luc dilakukan dalam domain bilangan, oleh karena itu sebelum dilakukan *enkripsi*, teks terlebih dahulu di konversi kedalam bentuk angka (Saputra *et al.* 2006). Algoritma Luc sebenarnya hampir sama dengan metode kriptografi yang lain yaitu metode RSA (Rivest, Shamir, Adleman), hanya saja fungsi pangkat pada metode RSA diganti

dengan fungsi Lucas dimana pertambahan nilai barisan Lucas sampai dengan n suku sangat cepat, sehingga dikembangkan fungsi modulo N > 2.

Saputra *et al.* (2006) telah melakukan penelitian tentang Kriptografi Teks Dengan Menggunakan Algoritma Luc.Penelitian tersebut menghasilkan enkripsi berupa teks yang telah disandikan dalam bentuk bilangan. Dwi (2012) juga melakukan penelitian keamanan pesan dengan judul Penerapan Algoritma Vigenere Cipher pada Aplikasi *SMS* Android, penelitian tersebut menghasilkan sebuah aplikasi yang dapat melakukan enkripsi dan dekripsi pada Android.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka rumusan masalah dalam penulisan skripsi ini adalah

a. Bagaimana menentukan proses enkripsi dan dekripsi teks dengan menggunakan algoritma Luc?
b. Bagaimana membuat perangkat lunak yang dapat melakukan enkripsi dan dekripsi *SMS* pada Android Gingerbread (2.3.4) dengan algoritma Luc?

1.3 Batasan Masalah

Karena keterbatasan pengetahuan penulis, maka ruang lingkup permasalahan dalam merancang perangkat lunak ini adalah sebagai berikut :

a. Pesan atau teks terdiri dari huruf kapital.
b. Konversi plaintext menjadi chipertext dengan menggunakan algoritma Luc.
c. Konversi karakter hanya dalam jangkauan nilai ASCII.

1.4 Tujuan

Tujuan penyusunan tugas akhir (skripsi) ini adalah :

a. Menentukan proses enkripsi dan dekripsi teks dengan menggunakan algoritma Luc.

b. Merancang perangkat lunak yang dapat melakukan enkripsi dan dekripsi *SMS* pada Android Gingerbread (2.3.4) menggunakan bahasa pemrograman *JAVA*.

1.5 Manfaat

Manfaat dari penyusunan tugas akhir (skripsi) ini yaitu :

a. Meningkatkan keamanan informasi yang terkandung dalam *SMS*.
b. Aplikasinya dapat digunakan dalam berbagai bidang, misalnya transaksi online, *SMS* banking, dan lain sebagainya.

BAB 2. TINJAUAN PUSTAKA

2.1. Landasan Matematika

Perkembangan kriptografi akan dipengaruhi oleh perkembangan matematika, terutama dalam hal algoritma (Kromodimoeljo, 2009). Beberapa teori dalam matematika yang berkaitan dengan kriptografi adalah :

2.1.1 Bilangan Prima

Bilangan bulat positif yang hanya mempunyai satu pembagi positif. Setiap bilangan bulat positif lainnya mempunyai minimal dua pembagi positif karena pasti dapat dibagi oleh 1 dan bilangan itu sendiri.

Definisi 2.1 Misalkan $n \in \mathrm{N}$ dengan $n > 1$, maka n disebut bilangan prima jika pembagi positif n adalah 1 dan n.

Contoh : bilangan bulat positif 2, 3, 5, 89, dan 101 adalah bilangan-bilangan prima (Riyanto, 2007).

2.1.2 *Greatest Common Divisor* (GCD)

Greatest Common Divisor (GCD) atau biasa disebut dengan *Faktor Persekutuan Terbesar* (FPB) adalah pembagi terbesar dari dua buah bilangan.

Defnisi 2.2 Jika $d|a$ dan $d|b$ maka d adalah pembagi persekutuan (*common divisor*) dari a dan b. Untuk setiap pasangan bilangan bulat a dan b kecuali jika $a = b = 0$, pembagi persekutuan terbesar dari a dan b adalah bilangan bulat unik d dimana:

a. d merupakan pembagi persekutuan dari a dan b,
b. jika c merupakan pembagi persekutuan dari a dan b, maka $c \leq d$.

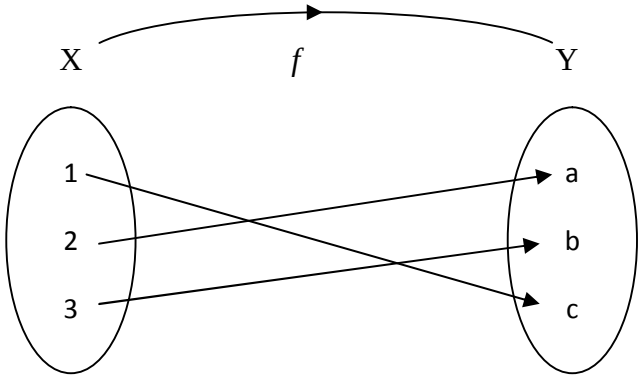
Definisi 2.3 Dua bilangan bulat *a* dan *b* , dimana salah satu dari keduanya tidak sama dengan 0, dikatakan *relatif prima* jika gcd(*a*,*b*) = 1 (Kromodimoeljo, 2009).

2.1.3 *Least Common Multiple* (*LCM*)

Diberikan *a*, *b* ∈ Z, dengan *a* dan *b* tidak sama dengan 0. *Least Common Multiple* (*LCM*) dari *a* dan *b* dinotasikan dengan [*a*, *b*], didefinisikan sebagai bilangan bulat positif terkecil yang dapat dibagi oleh *a* dan *b*.

2.2.4 Fungsi Bijektif

Jika *f* adalah fungsi yang injektif dan surjektif maka *f* disebut bijektif (berkorespondensi satu-satu). Misal X = {1,2,3}, Y = {a,b,c} diberikan fungsi *f*(1) = c, *f*(2) =a, *f*(3)=b



Gambar 2.1 Fungsi Bijektif

Fungsi *f* merupakan fungsi yang bijektif sebab :

a. *f* merupakan fungsi yang injektif
   setiap elemen *y*∈ Y mempunyai kawan tepat satu elemen *x*∈ X
b. *f* merupakan fungsi yang surjektif
   setiap elemen *y*∈ Y dikawankan dengan elemen *x*∈ X.

2.1.5 Fungsi Saling Invers Satu Sama Lain

Misalkan *f* dan *g* fungsi bijektif. Fungsi *f* dan *g* dikatakan saling invers satu sama lain jika $f(g(x)) = x$, *x* adalah elemen dalam domain fungsi *g*, dan $g(f(x)) = x$, dimana *x* adalah elemen dalam domain fungsi *f*.

2.1.6 Modulo

Definisi 2.4 Diberikan suatu bilangan bulat positif *m*. Untuk bilangan bulat *a* dan *b*, maka *a* dikatakan kongruen terhadap *b* mod *m* jika $m \mid (a-b)$.

Jika *a* kongruen terhadap *b* mod *m*, maka dapat dinyatakan dengan $a \equiv b \pmod{m}$ (atau, *a- b* habis dibagi oleh *m*). Jika $m \nmid (a-b)$, dinyatakan dengan $a \equiv b \pmod{m}$, dibaca *a* tidak kongruen dengan *b* mod *m*. Bilangan bulat positif *m* disebut *modulus*. Bentuk jamak dari *modulus* adalah *moduli* (Kromodimoeljo, 2009).

Teorema 2.1 (*Chinese Reminder Theorem*) *Jika terdapat beberapa persamaan dengan modulus berbeda sebagai berikut*

$$x \equiv a_1 (mod\ m_1),$$
$$x \equiv a_2 (mod\ m_2),$$
$$\vdots$$
$$x \equiv a_i (mod\ m_j),$$

dimana setiap pasangan modulus adalah relatif prima ($gcd(m_i,m_j) = 1$ *untuk* $i \neq j$*), maka terdapat solusi untuk x. jika* x_1 *dan* x_2 *merupakan solusi untuk x, maka* $x_1 \equiv x_2 \pmod{M}$ *dimana* $M = m_1 m_2 \ldots m_r$.

Bukti :

Pembuktian bahwa sistem persamaan seperti diatas mempunyai solusi untuk *x* bersifat konstruktif, jadi menghasilkan algoritma untuk mencari solusi. Didefnisikan $M_i = M = m_i$, jadi M_i merupakan produk dari semua modulus kecuali m_i. Karena $\gcd(m_i,M_i) = 1$, maka terdapat bilangan bulat N_i dimana

$M_i N_i \equiv 1 \pmod{m_i}$. Maka suatu solusi untuk *x* adalah

$$x = \sum_{j=1}^{r} a_j M_j N_j$$

Untuk setiap *i*, karena semua suku kecuali suku *i* dapat dibagi dengan *mi*, maka hanya suku *i* yang tidak ≡ 0 (mod m_i), jadi

$$x \equiv a_i M_i N_i \equiv a_i \pmod{m_i}$$

seperti yang dikehendaki. Untuk menunjukkan bahwa solusi *x* unik modulo *M*, kita tunjukkan bahwa jika x_1 dan x_2 adalah solusi untuk *x*, maka $x_1 \equiv x_2 \pmod{M}$. Untuk setiap *i*, $x_1 \equiv x_2 \equiv a_i \pmod{m_i}$, atau $x_1 - x_2 \equiv 0 \pmod{m_i}$. Jadi $x_1 - x_2 \equiv 0 \pmod{M}$, yang berarti $x_1 \equiv x_2 \pmod{M}$. ■

2.1.7 Fungsi Euler Phi

Definisi 2.5 Untuk bilangan bulat $n \geq 1$, $\emptyset(n)$ menyatakan banyaknya semua bilangan bulat positif yang lebih kecil atau sama dengan *n*, dan relatif prima terhadap *n*. Bila *n* merupakan bilangan prima maka $\emptyset(n) = n - 1$.

Definisi 2.6 Untuk $n, m \in N$ dan $g : N \quad C$, $g(n)$ dikatakan multiplikatif jika $g(nm) = g(n)g(m)$ ketika $\gcd(n,m) = 1$. Sedangkan $g(n)$ dikatakan multiplikatif lengkap jika $(g(n,m) = g(n)g(m)$ untuk sebarang $n, m \in N$.

Teorema 2.2 *Fungsi ∅ merupakan fungsi multiplikatif.*

Teorema ini menunjukkan bahwa $\emptyset(mn) = \emptyset(m)\emptyset(n)$ untuk semua bilangan-bilangan bulat $m \geq 1$ dan $n \geq 1$.

Bukti :

Jika $\gcd(m,n) = 1$. Hitung semua bilangan bulat antara 0 dan $mn \equiv 1$ yang relatif prima dengan *mn* (jadi tidak ada faktor bilangan yang lebih besar dari 1 yang juga merupakan faktor *mn*). Misal diberikan *j* untuk bilangan yang akan dihitung. Diberikan label j_1 untuk kemungkinan terkecil *j* modulo *m* dan j_2 untuk kemungkinan terkecil *j* modulo *n*, jadi $0 \equiv j_1 < m$, $0 \equiv j_2 < n$,

$$j \equiv j_1 \pmod{m},$$

$$j \equiv j_2 \pmod{n}.$$

Berdasarkan Teorema 2.1 untuk setiap pasangan j_1, j_2, hanya ada satu j antara 0 dan mn - 1 yang mengakibatkan kedua persamaan diatas berlaku. Juga perhatikan bahwa j relatif prima dengan mn jika dan hanya jika j relatif prima dengan m dan n. Jadi banyaknya j yang harus dihitung sama dengan banyaknya kombinasi pasangan j_1, j_2. Banyaknya j_1 yang relatif prima dengan m dimana $0 \leq j_1 < m$ adalah $\emptyset(m)$, sedangkan banyaknya j_2 yang koprima dengan n dimana $0 \leq j_2 < n$ adalah $\emptyset(n)$. Jadi banyaknya j adalah $\emptyset(m)\emptyset(n)$. ■

Contoh :

Ambil $m = 5$, $n = 6$, dan $\emptyset(mn) = \emptyset(30) = 8$. Dari seluruh bilangan bulat yang tidak lebih dari 30 hanya terdapat 8 bilangan yang merupakan relatif prima terhadap 30, yaitu 1, 7, 11, 13, 17, 19, 23, 29.

Sedangkan 30 = 5 . 6. Maka kita dapatkan pula $\emptyset(5) = 4$ yaitu 1, 2, 3, 4 dan $\emptyset(6) = 2$ yaitu 1 dan 5. Sehingga

$$\emptyset(30) = \emptyset(5.6) = \emptyset(5)\emptyset(6) = 4 \,.\, 2 = 8$$

(Kromodimoeljo, 2009).

Lemma 2.3 *Misalkan a dan n adalah bilangan bulat yang lebih besar dari 1 dan gcd(a,n) = 1. Jika* $a_1, a_2, \ldots, a_{\emptyset(n)}$ *merupakan bilangan – bilangan bulat positif yang lebih kecil dari n dan relatif prima terhadap n, maka* $aa_1, aa_2, \ldots, aa_{\emptyset(n)}$ *kongruen modulo n terhadap* $a_1, a_2, \ldots, a_{\emptyset(n)}$ *dalam suatu urutan tertentu.*

Teorema 2.3 (Euler) *Jika n bilangan bulat dengan n ≥ 1 dan gcd(a,n) = 1 maka* $a^{\emptyset(n)} \equiv 1$ *(mod n).*

Bukti :

Misalkan n bilangan bulat dengan $n > 1$, dan $a_1, a_2, \ldots, a_{\emptyset(n)}$ adalah bilangan-bilangan bulat positif yang lebih kecil daripada n dan relatif prima terhadap n.

Oleh karena gcd(a,n) = 1, dengan lemma 2.3 maka $aa_1, aa_2, \ldots, aa_{\emptyset(n)}$ kongruen modulo n terhadap $a_1, a_2, \ldots, a_{\emptyset(n)}$ dalam suatu urutan tertentu. Sehingga dapat ditulis

$$aa_1 \cong a_1'(\text{mod } n)$$
$$aa_2 \equiv a_2'(\text{mod } n)$$
$$\vdots$$
$$aa_{\emptyset(n)} \equiv a_{\emptyset(n)}'(\text{mod } n)$$

Di mana $a_1', a_1', \dots, a_{\emptyset(n)}'$ adalah bilangan-bilangan bulat $a_1, a_1, \dots, a_{\emptyset(n)}$ dalam suatu urutan tertentu. Hasil yang kita dapatkan dari kekongruensian $\emptyset(n)$ adalah

$$aa_1 aa_2 \dots aa_{\emptyset(n)} \equiv a_1' a_1', \dots, a_{\emptyset(n)}' \text{ (mod } n)$$
$$\equiv a_1 a_2 \dots a_{\emptyset(n)} \text{ (mod } n)$$

Sehingga

$$a_{\emptyset(n)}(a_1 a_2 \dots a_{\emptyset(n)}) \equiv a_1 a_2 \dots a_{\emptyset(n)} \text{ (mod } n).$$

Oleh karena gcd(a_i, n) = 1 untuk setiap i, berdasarkan Lemma 2.3 $\gcd(a_1 a_2 \dots a_{\emptyset(n)}, n) = 1$. Sehingga kedua ruas dapat dibagi dari kongruensi sebelumnya dengan faktor persekutuan $a_1 a_2 \dots a_{\emptyset(n)}$, dan didapatkan

$$a^{\emptyset(n)} \equiv 1 \text{ (mod } n)$$

(Kromodimoeljo, 2009).■

2.1.8 Simbol Legendre

Definisi 2.7 *(Quadratic Residue)*. Sebuah bilangan *a* yang relatif prima terhadap *n* adalah *quadratic residue* modulo *n* jika memenuhi

$$x^2 \equiv a(mod\ n) \qquad (2.3)$$

mempunyai solusi (jika hanya jika *a* adalah sebuah bilangan dalam modulo *n*), dan *a* disebut *quadratic nonresidue* modulo *n* jika persamaan (2.3) tidak mempunyai solusi (Hildebrand, 2011).

Contoh :

Misalkan *a* = 19 dan *n* = 5, maka 19 adalah *quadratic residue* karena memenuhi persamaan (2.3) dimana

$$2^2 \equiv 19 \bmod 5.$$

Definisi 2.8 Diberikan *p* adalah bilangan prima, dan *a* adalah bilangan bulat yang relatif prima terhadap *p* (*gcd*(*a*,*p*)) = 1. Simbol legendre dari *a* modulo *p* dinotasikan dengan $\left(\frac{a}{p}\right)$, dan didefinisikan

$$\left(\frac{a}{p}\right) = \begin{cases} 0 \text{ jika } p|a; \\ 1 \text{ jika } a \text{ merupakan quadratic residue } (\bmod p); \\ -1 \text{ jika } a \text{ merupakan non-quadratic residue } (\bmod)p. \end{cases}$$

Jadi $\left(\frac{a}{p}\right)$ dapat digunakan untuk memberi indikasi apakah suatu bilangan bulat merupakan suatu *quadratic residue* (mod p).

Teorema 2.4 (Simbol Legendre)

$$\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} (mod\ p)$$

Bukti :

Jika *p*|*a* maka kedua sisi dari persamaan akan sama dengan 0. Jika *p* tidak membagi *a* maka berdasarkan *Fermat's little theorem* akan didapatkan

$$\left(a^{(p-1)/2}\right)^2 = a^{p-1} \equiv 1 (mod\ p)$$

Jadi $a^{(p-1)/2} = \pm 1$ (Kromodimoeljo, 2009). ■

2.2. Kriptografi

Kriptografi (*cryptography*) berasal dari bahasa Yunani, terdiri dari dua suku kata yaitu *kripto* dan *graphia*. *Kripto* artinya menyembunyikan, sedangkan *graphia* artinya tulisan. Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi, seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data (Menezes *et al*, 1996).

Tujuan dari sistem Kriptografi dapat dijelaskan sebagai berikut :

a. kerahasiaan, adalah layanan yang digunakan untuk menjaga isi informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka/membaca informasi yang tersandikan,

b. integritas data, adalah berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak seperti penyisipan, penghapusan, dan pensubtitusian data lain ke dalam data yang sebenarnya,
c. autentikasi, adalah berhubungan dengan identifikasi/pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan melalui kanal harus diautentikasi keaslian, isi data, waktu pengiriman, dan lain-lain,
d. non-repudiasi, adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman atau terciptanya suatu informasi oleh pengirim.

(Menezes *et al*, 1996).

2.2.1 Chiper

Menurut Stinson (dalam Riyanto, 2007) algoritma kriptografi terdiri dari dua bagian, yaitu fungsi enkripsi dan dekripsi. Enkripsi adalah proses untuk mengubah teks asli (*plaintext*) menjadi teks telah disandikan (*ciphertext*), sedangkan dekripsi adalah kebalikannya yaitu mengubah teks yang disandikan (*ciphertext*) menjadi teks asli (*plaintext*). *Chiper* merupakan proses untuk enkripsi dan dekripsi data, terdapat 2 jenis algoritma Kriptografi berdasarkan jenis kuncinya, yaitu :

a. Kriptografi Simetri

Algoritma simetris atau disebut juga algoritma Kriptografi konvensional adalah algoritma yang menggunakan kunci yang sama untuk proses enkripsi dan proses dekripsi.

Algoritma Kriptografi Simetris dibagi menjadi 2 kategori yaitu algoritma aliran (*Stream Chipers*) dan algoritma blok (*Block Chipers*). Pada algortima aliran. Proses penyandian berorientasi pada satu bit atau satu byte data. Sedangkan pada algoritma blok, proses penyandiannya berorientasi pada sekumpulan *bit* atau *byte* data

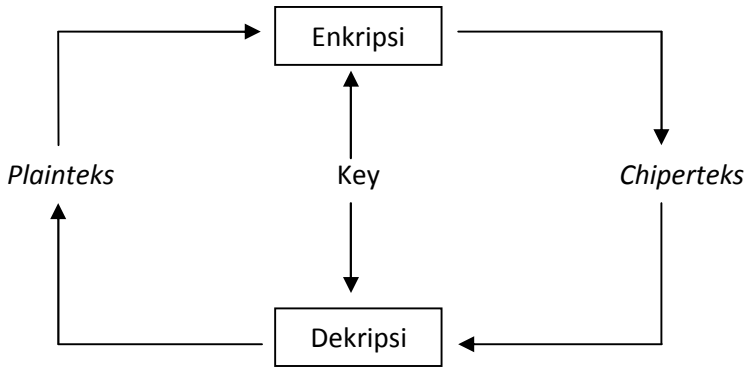
(per blok). Contoh algoritma kunci simetris adalah DES (*Data Encryption Standard*), blowfish, twofish, MARS, IDEA,3DES(DES diaplikasikan 3 kali), AES (*Advanced Encryption Standard*) yang bernama asli Rijndael.

b. Kriptografi Asimetris

Kriptografi asimetrik (*Asymetric Cryptography*) adalah algoritma yang menggunakan kunci yang berbeda untuk proses enkripsi dan dekripsi. Kunci enkripsi dapat disebarkan kepada umum dan dinamakan sebagai kunci public (*public key*). Sedangkan kunci dekripsi disimpan untuk digunakan sendiri dan dinamakan sebagai kunci privat (*private key*). Contoh algoritma yang menggunakan kunci asimetris adalah RSA (*Rivest Shamir Adleman*), ECC (*Elliptic Curve Cryptography*) dan algoritma LUC.

Pada kriptosistem asimetrik, setiap pelaku sistem informasi memiliki sepasang kunci, yaitu kunci public dan kunci privat. Kunci public digunakan sebagai kunci untuk enkripsi data dan dapat didistribusikan kepada umum, sedangkan kunci privat disimpan untuk dekripsi pesan yang tersandikan.

Dalam Gambar 2.2 berikut ini memperlihatkan proses aliran enkripsi dan dekripsi.



Gambar 2.2 Aliran proses *Enkripsi* dan *Dekripsi*

2.2.2 Barisan Lucas

Menurut Smith & Michael (1993) barisan Lucas merupakan deret U_n dan V_n yang dibangun oleh dua buah bilangan bulat positif *P* dan *Q*. Kemudian dibangun sebuah persamaan kuadrat :

$$X^2 - PX + Q = 0$$

Akar dari persamaan adalah $\left(P \pm \sqrt{P^2 - 4Q}\right)/2$. Bagian $\left(\sqrt{P^2 - 4Q}\right)$ disebut Diskriminan atau D. Dimisalkan kedua akar sebagai :

$\alpha = \frac{P + \sqrt{D}}{2}$ dan $\beta = \frac{P - \sqrt{D}}{2}$, sesuai dengan persamaan tersebut α dan β dapat diperlihatkan

$\alpha + \beta = P$, $\alpha\beta = Q$, $\alpha - \beta = \sqrt{D}$ diasumsikan pemilihan $D \neq 0$.

Kemudian barisan Lucas didefinisikan sebagai berikut :

$U_n(p,Q) = \frac{\alpha^n - \beta^n}{\alpha - \beta}$ dan

$V_n(P,Q) = \alpha^n + \beta^n$ (2.4)

Untuk $n \geq 2$:

$V_n(P,Q) = PV_{n-1}(P,Q) - QV_{n-2}(P,Q)$ dan (2.5)

$U_n(P,Q) = PU_{n-1}(P,Q) - QU_{n-2}(P,Q)$

Sebagai contoh dimisalkan $P = 3$, $Q = 1$. Maka sepuluh barisan Lucas pertama disajikan pada Tabel 2.1 berikut.

Tabel 2.1 Contoh sepuluh bilangan lucas pertama

n	0	1	2	3	4	5	6	7	8	9
$V_n(3,1)$	2	3	7	18	47	123	322	843	2207	5778
$U_n(3,1)$	0	1	3	8	21	55	144	377	987	2584

Fungsi-fungsi tersebut digunakan untuk mempercepat perhitungan iterasi *n*.

2.2.3 Rantai Lucas (*Lucas Chain*)

Teorema 2.5 (Rantai Lucas). *Diberikan sebuah bilangan bulat n, maka barisan Rantai Lucas dari n adalah* ($b_0, b_1, \dots, b_r$) *dimana* b_0 = *(0 atau 1),* b_1 = *(0 atau 1) dan* $b_r = 0$.

Bukti :

Untuk b_0 hingga b_r, misalkan z = n mod 2, jika $z = 1$ maka ($b_0 = 1$ dan $n/2$), selain itu jika $z = 0$ maka ($b_0 = 0$ dan $n/2$). Pembagian tersebut dilakukan secara berulang hingga b_r dan pada akhirnya $b_r = 0$.■

Definisi 2.9 Diberikan sebuah barisan ($a_0, a_1, \dots a_x$), panjang Rantai Lucas didefinisikan sebagai x (Wang *et al*, 1999).

2.2.4 Barisan Lucas Dalam Kriptografi

Menurut Saputra *et al*. (2006) Algoritma Luc merupakan metode kriptografi dengan menggunakan dua kunci yang berbeda dalam kriptosistemnya. Untuk mengenkripsi file teks digunakan fungsi enkripsi yang menggunakan sebuah kunci publik, hasil enkripsi merupakan file terenkripsi yang aman dari pihak yang tidak berhak atas informasi didalamnya. Selanjutnya untuk membaca file yang telah terenkripsi digunakan fungsi dekripsi dengan menggunakan kunci privat (*Private Key*) yang akan menghasilkan file teks yang sama dengan teks aslinya. Operasi pada Algoritma Luc dilakukan dalam domain bilangan, oleh karena itu sebelum dilakukan proses enkripsi, teks terlebih dahulu dikonversikan kedalam bentuk angka.

Pertambahan nilai barisan Lucas sampai dengan n suku sangat cepat, sehingga dikembangkan fungsi modulo $N > 2$. Dengan mengaplikasikan operasi modulo dalam setiap langkahnya didapatkan hasil yang sama. Sehingga memenuhi persamaan :

$$V_n(P \bmod N, Q \bmod N) = V_n(P,Q) \bmod N$$

Jika $Q = 1$ maka didapatkan fungsi : $V_n(P,1) \bmod N$.

Sehingga fungsi Lucas yang akan dipakai dalam algoritma Luc adalah :

$$V_{de}(P,1) \equiv P \bmod N$$

Aplikasi selanjutnya dalam kriptografi yaitu nilai *e* dan *d* disebut sebagai kunci, dengan *e* adalah kunci enkripsi dan *d* adalah kunci dekripsi. Misalkan M merupakan plainteks asli dengan M<*N*, sebuah plainteks M akan dienkripsi dengan fungsi Lucas sehingga :

$$V_e(\text{M mod } N,1) \equiv \text{C, dengan C adalah chiperteks}$$

Selanjutnya chiperteks didekripsi dengan barisan Lucas yang lain yaitu :

$$V_d(\text{C mod } N,1) \equiv \text{M}$$

Beberapa fungsi lucas lain yang digunakan dalam kriptosistem

$$V_{2n} = V_n^2 - 2 \pmod N \tag{2.6}$$

$$V_{2n+1} = PV_n^2 - V_nV_{n-1} - P \pmod N \tag{2.7}$$

$$V_{2n-1} = V_nV_{n-1} - P \pmod N \tag{2.8}$$

(Smith & Michael, 1993).

Dimana *n* adalah *e* atau *d*, beberapa fungsi diatas (2.6), (2.7), (2.8) bertujuan untuk mempercepat penghitungan iterasi *n*. Dalam pengembangan barisan Lucas sebagai algoritma dalam kriptografi, yang akan digunakan hanya fungsi Lucas $V_n(P,Q)$ pada persamaan (2.6) dan persamaan (2.8).

2.1.4 Algoritma Luc

Dalam menyelesaikan algoritma Luc terdapat tiga tahap utama yaitu algoritma pembangkitan kunci, proses enkripsi dan proses dekripsi.

a. Algoritma pembangkitan Kunci
    1) Algoritma Kunci Public
        a) Pilih dua bilangan prima sebarang, misal *p* dan *q* dimana $p \neq q$.
        b) Hitung nilai $N = p \times q$. Nilai *N* akan digunakan dalam menghitung modulo pada proses enkripsi dan dekripsi.
        c) Hitung semua bilangan yang relatif prima terhadap (*p*-1), (*p*+1), (*q*-1) dan (*q*+1).

d) Pilih salah satu bilangan secara acak dari hasil yang didapatkan pada poin (c) sebagai kunci public *e*.

2) Algoritma Kunci Privat

a) Masukkan dua bilangan prima *p* dan *q*.

b) Masukkan *e* yang dihitung pada tahap pembangkitan kunci public.

c) Hitung determinan $D = C^2 - 4$.

d) Cari simbol legendre dari $\frac{D}{p}$ dan $\frac{D}{q}$.

e) Hitung nilai $S(N) = LCM\ [\left(p - \frac{D}{p}\right), (q - \frac{D}{p})]$.

f) Hitung $ed \equiv 1 \bmod S(N)$.

Nilai *d* diperoleh dengan cara berikut

$$e.d \equiv 1 \bmod S(N)$$

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

Dengan *k* adalah bilangan peubah sebarang sehingga nilai *d* atau kunci dekripsi mempunyai 4 kemungkinan sesuai dengan nilai *S(N)*. Nilai*(d,N)* yang diperoleh merupakan kunci dekripsi (kunci privat) dari kunci enkripsi *(e,N)*.

Proses pembangkitan kunci dilakukan dengan rahasia terutama nilai bilangan prima *p* dan *q*, serta nilai *S(N)* yang dipakai untuk dekripsi. Namun pendistribusian kunci public tidak bersifat rahasia, karena tujuan dari kunci public adalah untuk enkripsi.

b. Proses Enkripsi

Proses enkripsi adalah proses pengacakan data atau pesan, misalkan A akan bertukar informasi dengan B, pihak A dan B sama-sama melakukan pembangkitan kunci seperti yang telah dijelaskan pada sub bab sebelumnya, kemudian A dan B

bertukar kunci public (A menerima kunci public dari B dan B menerima kunci public dari A) dimana pertukaran kunci tersebut tidak bersifat rahasia.

Dalam proses enkripsi dimisalkan B ingin mengirim data atau pesan kepada A, maka B terlebih dahulu harus mempunyai kunci public (*e*) yang diberikan oleh A. Selanjutnya proses enkripsi dapat dijelaskan sebagai berikut :

1) plainteks (M) adalah isi pesan atau informasi yang akan disampaikan oleh B kepada A.
2) nilai *e* dan *N* didapatkan dari kunci public yang telah diberikan A kepada B.
3) plainteks (M) yang akan disampaikan kepada A dipecah atau diatur menjadi blok-blok m_1, m_2, …m_i yang mempunyai dua karakter pada tiap blok.
4) setiap blok yang telah didapatkan (m_i) di ubah dalam bentuk ASCII kemudian di enkripsi dengan persamaan $c_i = V_e(m_i, 1) mod\ N$.
5) setiap blok yang telah dienkripsi (c_i) digabungkan kembali sehingga menjadi sebuah chiperteks yang utuh (C).

c. Proses Dekripsi

Proses dekripsi sebuah chiperteks hampir sama dengan proses enkripsi sebuah pesan, perbedaannya adalah persamaan yang dipakai adalah $m_i = V_d(c_i, 1) mod\ N$ serta kunci yang dipakai adalah kunci dekripsi (*d*,*N*) dimana kunci tersebut telah di ketahui pada proses pembangkitan kunci. Misalkan A telah menerima chiperteks (C) dari B dengan menggunakan kunci public yang telah diberikan kepada B, maka langkah-langkah dekripsi adalah sebagai berikut

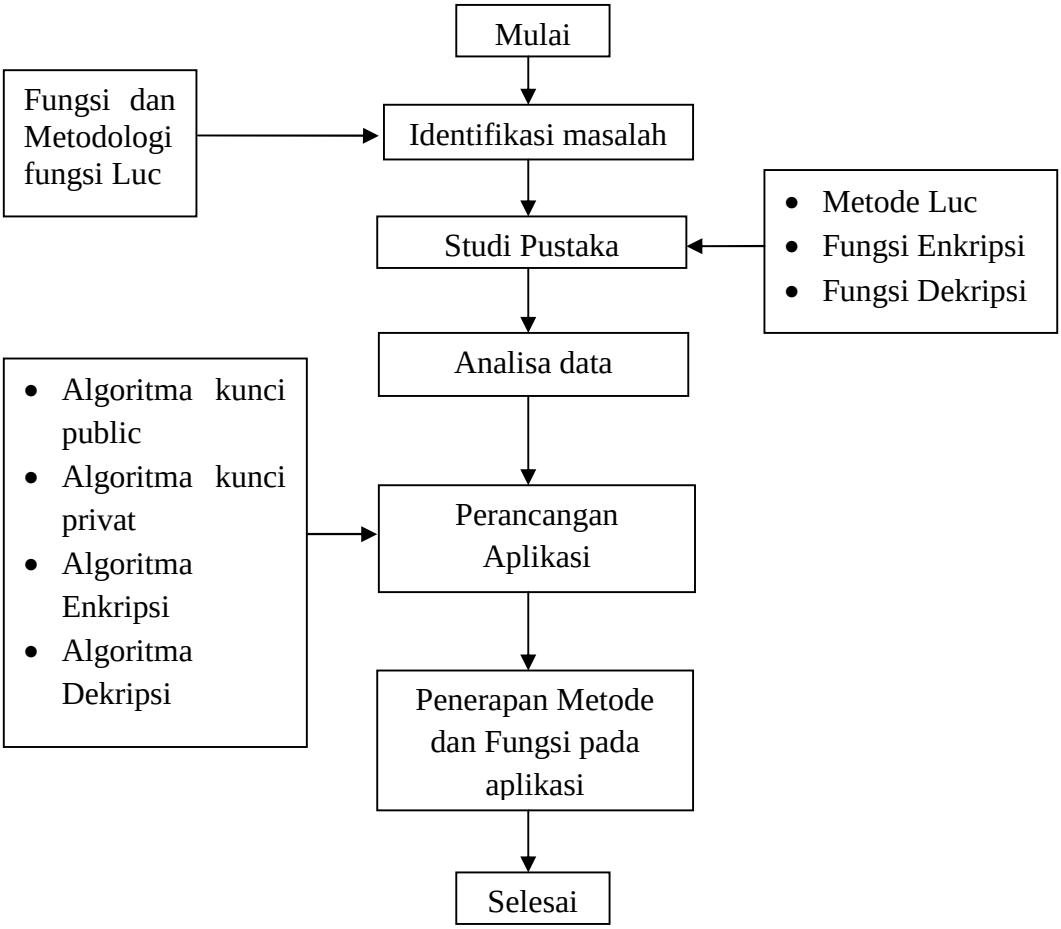
1) Chiperteks (C) adalah isi pesan atau informasi yang telah dienkripsi oleh B dan diterima oleh A.
2) Nilai *N* didapatkan dari kunci privat yang telah dicari pada tahap pembangkitan kunci.
3) Chiperteks yang telah diterima dari B dipecah atau diatur menjadi blok-blokc_1, c_2, … c_i yang mempunyai dua karakter pada tiap blok.

4) Setiap blok yang telah didapatkan (c_i) di ubah dalam bentuk ASCII.
5) Hitung nilai deskriminan $D = c^2 - 4$.
6) Cari simbol legendre dari $\frac{D}{p}$ dan $\frac{D}{q}$.
7) Hitung LCM $\left(p - \frac{D}{p}, q - \frac{D}{q}\right)$.
8) Cari nilai *d* dari *ed* ≡ 1 mod *S(N)*.
9) Gunakan *d* dalam persamaan dekripsi $m_i = V_d(c_i, 1) mod\ N$.
10) Setiap blok yang telah didekripsi (m_i) digabungkan kembali sehingga menjadi sebuah plainteks yang utuh (M).

BAB 3. METODOLOGI PENELITIAN

3.1 Kerangka Berfikir

Secara garis besar langkah-langkah dalam menyelesaikan Sistem Keamanan Pesan pada Android dapat dilihat pada Gambar 3.1



Gambar 3.1 Skema Langkah Penyelesaian Penelitian

a. Identifikasi Masalah

   Identifikasi masalah merupakan tahap awal dari penelitian ini. Masalah yang diidentifikasi adalah keamanan pesan dalam Android.

b. Studi pustaka

   Studi pustaka dilakukan untuk melengkapi pengetahuan dasar yang dimiliki peneliti, sehingga peneliti dapat menyelesaikan penelitian ini, dimana pada tahap ini dikumpulkan beberapa referensi yang berkaitan dengan metodologi-metodologi yang digunakan dalam perancangan aplikasi keamanan pesan dalam Android.

c. Analisa data

   Analisa data berfungsi untuk menganalisa permasalahan keamanan data yang dibutuhkan pada saat ini. Dalam hal ini analisa dilakukan dengan pengambilan fungsi-fungsi yang telah diperoleh dari studi pustaka untuk diimplementasikan di dalam aplikasi SMS.

d. Perancangan Aplikasi

   Perancangan aplikasi merupakan tindak lanjut dari analisa data, dimana pada tahap ini dibuat sampel-sampel fungsi, diantaranya pembangkitan kunci public, pembangkitan kunci privat, fungsi enkripsi, fungsi dekripsi yang dapat diaplikasikan dalam suatu algoritma sehingga menjadi aplikasi yang mampu mengimplementasikan fungsi-fungsi diatas. Beberapa proses dalam perancangan aplikasi adalah:

   1) Mengimpor *package java* yang digunakan dalam aplikasi seperti *java Math*, *SmsManager*;
   2) Memasukkan fungsi pembangkitan kunci;
   3) Memasukkan fungsi enkripsi;
   4) Memasukkan fungsi dekripsi.

e. Penerapan metode dan fungsi pada aplikasi

   Pada tahap ini dilakukan proses mengimplementasikan fungsi-fungsi yang didapat pada tahap perancangan aplikasi yaitu fungsi pembangkitan

kunci, fungsi enkripsi dan fungsi dekripsi sehingga dihasilkan aplikasi yang mampu mengimplementasikan fungsi-fungsi tersebut.

3.2 Perangkat Penelitian

Dalam melakukan penelitian, penulis menggunakan tiga perangkat :

3.2.1 Perangkat Lunak

Perangkat lunak atau *Software* yang digunakan penulis adalah *Eclipse* versi *Juno* berbasis bahasa pemrograman Java.

3.2.2 Media

Media yang digunakan penulis untuk uji coba aplikasi system keamanan pesan pada Android Gingerbread ini adalah Motorola XT530 yang mempunyai spesifikasi sebagai berikut

a. *Processor* : 800MHz
b. *RAM* : 512 Mb
c. *Memory Internal* : 130 Mb
d. *Memory External* : 2 Gb
e. *OS* : Android Gingerbread 2.3

BAB 4. HASIL DAN PEMBAHASAN

Berdasarkan dengan apa yang telah diuraikan pada bab 3, pada bab ini akan dibahas mengenai membangun fungsi dan implementasi Algoritma Luc pada aplikasi keamanan pesan pada Android. Langkah awal untuk membangun sistem keamanan pesan dengan menggunakan algoritma Luc yaitu membangkitkan fungsi kunci public, proses enkripsi dan proses dekripsi. Beberapa tahap yang harus dilakukan untuk mendapatkan fungsi tersebut adalah sebagai berikut:

4.1 Membangun Fungsi

a. Identifikasi masalah

Identifikasi masalah merupakan tahap awal dari penelitian ini, dimana masalah yang diidentifikasi adalah tentang keamanan SMS dalam Android serta masalah-masalah yang dapat timbul dalam proses pengiriman SMS seperti yang dijelaskan dalam BAB I.

b. Studi pustaka

Studi pustaka dilakukan untuk melengkapi pengetahuan dasar yang dimiliki peneliti, sehingga peneliti dapat menyelesaikan penelitian ini, dimana pada tahap ini dikumpulkan beberapa referensi yang berkaitan dengan metodologi-metodologi yang digunakan dalam perancangan aplikasi keamanan pesan dalam Android. Dalam hal ini, peneliti menggunakan algoritma Luc sebagai salah satu solusi atas masalah keamanan pesan pada Android.

c. Analisa data

Dalam analisa data dilakukan identifikasi terhadap fungsi-fungsi yang didapatkan untuk menyelesaikan masalah keamanan pesan dalam Android,

termasuk hal-hal yang berkaitan dengan pengolahan pesan pada Android. Fungsi-fungsi yang dianalisa adalah fungsi untuk pembangkitan kunci privat maupun public, fungsi yang digunakan pada proses enkripsi, dan fungsi yang digunakan pada proses dekripsi.

4.2 Implementasi Algoritma

a. Perencanaan aplikasi

Aplikasi merupakan kumpulan dari beberapa algoritma yang mempunyai fungsi-fungsi tertentu, karena itu dalam perencanaan aplikasi dibangun algoritma-algoritma dari fungsi-fungsi yang telah didapatkan pada tahap analisa data. Aplikasi ini ditulis dengan menggunakan *software Eclipse Juno* dengan menggunakan bahasa pemrograman java. Secara garis besar langkah-langkah pembuatan aplikasi ini adalah sebagai berikut:

1) Mengimpor *package java* yang berfungsi untuk menerima dan mengirim pesan.
2) Mengimpor *package java Math* agar dapat melakukan perhitungan matematis karena pada penerapan fungsi terdapat perhitungan yang menggunakan akar, pembulatan keatas serta penghitungan untuk pangkat *n*.
3) Memasukkan fungsi pembangkitan kunci. Dalam hal ini terdapat dua fungsi yaitu:
   a) Pembangkitan kunci public

      Pembangkitan kunci public cukup sederhana karena untuk kunci *N* hanya dengan mengalikan dua bilangan prima yang telah ditentukan. Selanjutnya untuk menentukan nilai *e* maka dicari bilangan yang berelatif prima dengan (*p*-1), (*p*+1), (*q*-1), (*q*+1). Dengan menggunakan fungsi pembagian modulo maka pencarian bilangan yang relatif prima dapat ditemukan.

b) Pembangkitan kunci privat

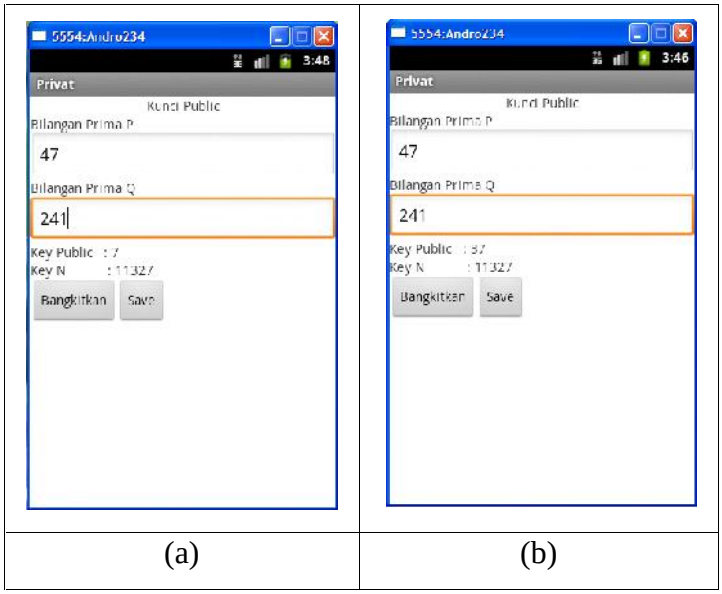
Bagian utama dari pembangkitan kunci privat adalah proses mencari nilai *d* dan membutuhkan kunci public *e*.

4) Algoritma proses enkripsi dapat dilihat pada subbab 2.1.4. tahap pertama dalam proses enkripsi adalah menentukan Rantai Lucas dan disimpan dalam *k[x]* yang mempunyai nilai 0 atau 1, tahap selanjutnya adalah proses dekripsi dengan menggunakan persamaan 2.6 atau 2.8 tergantung nilai *k*[*x*].
5) Proses dekripsi kurang lebih sama dengan proses enkripsi, perintah yang digunakan pada tahap dekripsi ini sama dengan proses enkripsi hanya nilai *e* dalam proses enkripsi diganti dengan nilai *d* yang didapatkan pada proses pembangkitan kunci privat. Pada visualisasi program, nilai yang dimasukkan adalah kunci public *e* karena pembangkitan kunci privat membutuhkan kunci public *e*, sedangkan nilai *d* tidak ditampakkan karena setiap pasangan huruf mempunyai kunci privat yang berbeda, sehingga kunci privat dihitung dalam program.

b. Penerapan metode dan fungsi pada aplikasi

Algoritma yang telah dihasilkan pada tahap sebelumnya diterapkan pada bahasa pemrograman *Java* untuk membangun sebuah aplikasi yang mampu mengimplementasikan fungsi-fungsi yang telah didapatkan.

Penerapan algoritma pada bahasa pemrograman membutuhkan variabel-variabel tertentu untuk mendefinisikan nilai-nilai yang diperlukan dalam fungsi agar mendapatkan hasil yang diinginkan. Misalnya pada fungsi pembangkitan kunci diperlukan nilai dua bilangan prima *p* dan *q*, dimana $p \neq q$. Maka dalam aplikasi dibutuhkan sebuah *TextEdit* untuk input nilai *p* dan *q*. Selanjutnya dibutuhkan sebuah *TextEdit* atau *TextView* untuk menampilkan hasil pembangkitan kunci secara acak sesuai dengan nilai yang relatif prima terhadap *p* dan *q*. Contohnya dapat dilihat pada Gambar 4.1, dengan menggunakan bilangan prima yang sama dapat dibangkitkan kunci privat yang berbeda.



(a) (b)

(a) Kunci public 7; (b) Kunci public 37

Gambar 4.1 *Layout* pembangkitan kunci

Gambar 4.2 dibawah ini merupakan tampilan proses enkripsi dan dekripsi dalam aplikasi yang membutuhkan beberapa *TextEdit* untuk *input* dan menampung *Output*. Pada proses enkripsi digunakan kunci privat (*e*,*N*) = (7, 11327), sedangkan pada proses dekripsi diperlukan nilai *p* = 47, *q* = 241, dan *e* = 7.

(a) (b)

(a). Proses enkripsi; (b) Proses dekripsi
Gambar 4.2 Perbandingan *layout* proses dekripsi dan enkripsi

4.3 Hasil Implementasi Algoritma Luc

Berikut ini adalah penjelasan tentang hasil perhitungan, program, komponen yang digunakan dalam aplikasi keamanan SMS pada Android dengan Algoritma Luc.

a. Pembangkitan kunci public

Kunci public dibangkitkan dengan menentukan dua bilangan prima, misalkan bilangan prima $p = 47$ dan $q = 241$.

Untuk nilai N adalah hasil dari perkalian dua bilangan prima p dan q

$N = p \times q = 47 \times 241 = 11327$

Selanjutnya menentukan nilai e dimana e adalah bilangan yang relatif prima terhadap N.

RP $(p - 1)$ = RP (46) = {3, 5, 7, 11, 13, 17, 19, 29, 31, 37, 41, 43}

RP $(p + 1)$ = RP (48) = {5, 7, 11 ,13 , 17, 19, 23, 29, 31, 37, 41, 43, 47}

RP $(q - 1)$ = RP (240) = {7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, … , 239}

RP $(q + 1)$ = RP (242) = {3, 5, 7, 13, 17, 19, 23, 29, 31, 37, 41, 43, … , 241}

Dari perhitungan tersebut terdapat beberapa bilangan yang sama, yaitu {7, 13, 17, 19, 29, 31, 37, 41, 43} nilai e dipilih dari bilangan tersebut, misalnya e = 7 maka kunci public adalah (7, 11327).

b. Proses enkripsi

Proses enkripsi dilakukan oleh pihak yang akan mengirim pesan dan diasumsikan pihak pengirim telah mendapatkan kunci public yang diberikan pihak penerima. 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 *space*. Selanjutnya, setiap karakter dalam blok diubah menjadi nilai ASCII dan dihitung dengan menggunakan fungsi lucas

$$c_i = V_e(m_i, 1) mod\ N$$

Dengan e = 7, m_i adalah nilai ASCII tiap blok, N = 11327 dan c_i adalah hasil enkripsi tiap blok. Misal pengirim ingin mengirimkan kata “IVAN”, apabila dipisahkan dalam blok maka teks berubah menjadi “IV” dan “AN”. Selanjutnya adalah merubah tiap blok dalam bentuk ASCII, maka didapatkan bilangan ASCII IV = 7386 dan AN = 6578. Dengan menggunakan kunci public yang dibangkitkan pada tahap sebelumnya (e, N) = (7, 11327), tentukan barisan Rantai Lucas dalam $k[x]$

Tabel 4.1 Penentuan Rantai Lucas

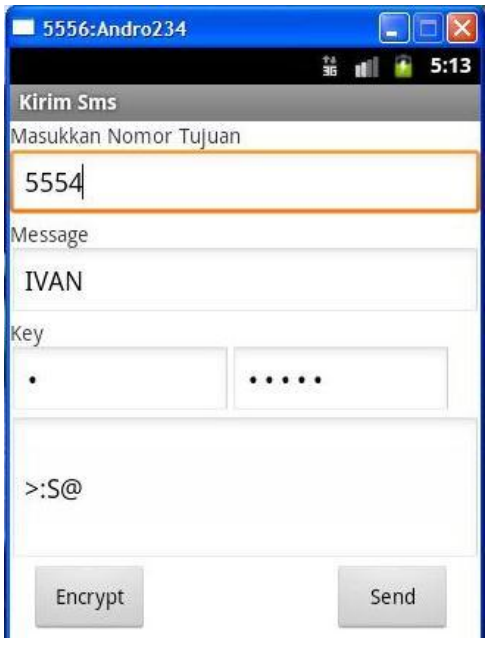
x	$k[x]$	e
1	1	e-1 = 6
2	0	e/2 = 3
3	1	e-1 = 2
4	0	e/2 = 1

Didapatkan $k[x]$ = {1, 0 ,1, 0} dimana $k[x]$ adalah Rantai Lucas maka $k[x]$ = {0, 1, 0, 1}. Proses enkripsi dengan menggunakan persamaan 2.6 atau 2.8 sesuai dengan nilai $k[x]$

Tabel 4.2 Hasil perhitungan enkripsi

$k[x]$	V_n	Hasil
0	V_2	2162
1	V_3	1403
0	V_6	8836
1	V_7	6258

Dengan menggunakan cara yang sama untuk blok berikutnya, maka hasil akhir dari proses enkripsi akan didapatkan nilai 6258 dan 8364, langkah berikutnya adalah mengembalikan nilai tersebut kedalam karakter, sehingga hasil akhir setelah diubah dalam karakter adalah **>:S@** seperti yang tampak pada Gambar 4.3 dibawah ini



Gambar 4.3 Proses Enkripsi dan Pengiriman pesan

c. Pembangkitan kunci privat

Pembangkitan kunci privat dilakukan jika telah menerima chiperteks, hal ini dikarenakan $D = c^2 - 4$ dimana c adalah nilai ascii dari chiperteks. Tahap pertama adalah menghitung deskriminan $D = c^2 - 4$. Dalam contoh diatas pasangan karakter pertama adalah **>:** yang mempunya nilai ASCII 6258, maka

$$D = (6258)^2 - 4$$

Simbol legendre untuk $\frac{D}{p}$ adalah $\frac{39162560}{47} = -1$, sedangkan simbol legendre untuk $\frac{D}{q}$ adalah $\frac{39162560}{241} = 1$.

Langkah berikutnya adalah mencari LCM$\left(p - \frac{D}{p}, q - \frac{D}{q}\right)$.

S(N) = LCM$(p + 1, q - 1)$

S(N) = LCM(47 + 1, 241 – 1) = 240

Nilai *d* didapat dengan menggunakan cara berikut

$$ed \equiv 1 \bmod S(N)$$

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

$$d = \frac{1 + (3\ x\ 240)}{7}$$

$$d = 103$$

maka didapatkan kunci privat adalah (103, 11327).

Kunci privat akan di bangkitkan kembali pada perhitungan setiap blok berikutnya, kunci privat akan dibangkitkan kembali sesuai dengan ASCII pada blok tersebut.

d. Proses dekripsi

Proses dekripsi dapat dilakukan jika penerima telah menerima pesan dalam bentuk chiperteks yang di enkripsi dengan menggunakan kunci public milik penerima, dengan kata lain penerima tidak bisa membaca chiperteks

yang telah di enkripsi dengan menggunakan kunci public yang bukan kunci public miliknya sendiri.

Jika chiperteks tersebut di enkripsi dengan menggunakan kunci public milik penerima, maka penerima dapat membaca chiperteks tersebut dengan menggunakan kunci privat yang telah dibangkitkan pada tahap pembangkitan kunci. Pada contoh diatas didapatkan chiperteks yang berisi **>:S@** dimana karakter-karakter tersebut akan dikembalikan menjadi teks seperti semula.

Langkah awal proses dekripsi adalah membagi chiperteks menjadi blok-blok yang berisikan dua karakter, maka dari chiperteks yang dihasilkan pada contoh diatas didapatkan **>:** dan **S@**. Selanjutnya tiap blok di konversi kedalam nilai ASCII dan didapatkan **>:** = 6258 dan **S@** = 8364. Untuk blok pertama didekripsi dengan menggunakan kunci privat yang telah dibangkitkan (103,11327) proses dekripsi dilakukan dengan menggunakan persamaan dekripsi

$$V_d(c_i \bmod N,1) \equiv M$$

Dimana d = 103, c_i = nilai ASCII tiap blok, N = 11327, dan M adalah pesan asli, misalkan m_i adalah hasil dekripsi tiap blok, langkah berikutnya adalah membangkitkan $k[x]$

Tabel 4.3 Pembangkitan Rantai Lucas dekripsi

x	$k[x]$	d
1	1	d-1 = 102
2	0	d/2 = 51
3	1	d-1 = 50
4	0	d/2 = 25
5	1	d-1 = 24
6	0	d/2 = 12

7	0	$d/2 = 6$
8	0	$d/2 = 3$
9	1	d-1 = 2
10	0	$d/2 = 1$

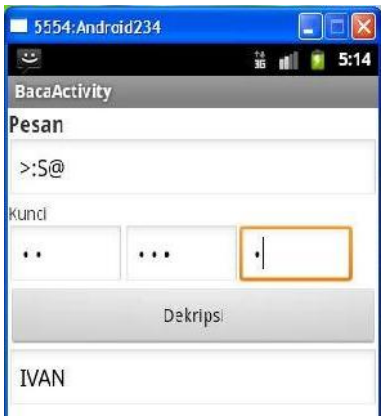
Karena $k[x]$ berfungsi sebagai Rantai Lucas, maka $k[x]$ = {0, 1, 0, 0, 0, 1, 0, 1, 0, 1}, selanjutnya dilakukan proses dekripsi dengan menggunakan persamaan 2.6 atau 2.8 tergantung pada nilai $k[x]$, hasil perhitungan dekripsi dapat dilihat dalam tabel 4.4 berikut

Tabel 4.4 Hasil Perhitungan Dekripsi

$k[x]$	V_n	hasil
0	V_2	5123
1	V_3	9393
0	V_6	2444
0	V_{12}	3805
0	V_{24}	2117
1	V_{25}	9736
0	V_{50}	5358
1	V_{51}	1403
0	V_{102}	8836
1	V_{103}	7386

Hasil akhir proses dekripsi untuk blok pertama adalah 7386, jika diubah dalah bentuk karakter maka hasilnya adalah IV. Blok-blok berikutnya mengikuti langkah yang sama seperti perhitungan diatas, maka jika $m_1 + m_2 + m_3 + \ldots + m_i$ akan dihasilnya teks asli M. Visualisasi program dapat dilihat

pada Gambar 4.4, dengan memasukkan nilai p = 47, q = 241, dan e = 7, maka akan dibangkitkan kunci privat sehingga dapat menembalikan chiperteks menjadi teks asli.



Gambar 4.4 Proses dekripsi

4.4 Pembahasan

Sistem keamanan pesan dengan algoritma Luc pada skripsi ini merupakan salah satu perkembangan implementasi kriptografi dalam bidang telekomunikasi. Manfaat kriptografi telah banyak digunakan dalam teknologi informasi, dari beberapa sumber artikel yang penulis ketahui, salah satu aplikasi yang memanfaatkan kriptografi sebagai metode keamanan dalam pertukaran informasi adalah *Skype*, dalam artikel tersebut disebutkan bahwa *Skype* menggunakan algoritma RC5 sebagai metode keamanan pesan.

Aplikasi ini menggunakan Algoritma algoritma Luc sebagai keamanan pesan dimana algoritma Luc adalah salah satu algoritma kriptografi yang belum terpecahkan. Namun selama melakukan penelitian tentang algoritma Luc ini, penulis menemukan masalah yaitu dalam menentukan bilangan prima pada tahap pembangkitan kunci harus besar, hal ini dikarenakan karena nilai ASCII dari tiap blok harus lebih kecil dari *N*, misalnya seperti contoh pembangkitan kunci public yang telah diuraikan diatas didapatkan nilai N = 11327, jika plainteks menggunakan karakter *lowercase* (huruf kecil) misalkan teks yang akan di

enkripsi adalah “ivan”, dihasilkan blok pertama adalah “iv” yang memiliki nilai ASCII 105118. Karena *N* < 105118 maka pada tahap dekripsi nilai 105118 tidak bisa didapatkan, misalkan terdapat sebuah bilangan Y > 11327, maka Y mod 11327 < 11327.

Penggunaan karakter kecil (*lowercase*) dalam aplikasi ini cenderung akan mengakibatkan nilai yang relatif besar, karena nilai ASCII hanya terbatas hingga 255, maka jika nilai ASCII lebih dari 255 akan menyebabkan *error* dan menghasilkan karakter yang tidak dikenali, hal ini juga dapat mengakibatkan pembengkakan jumlah karakter dalam pesan. Untuk lebih jelas dapat dilihat pada Gambar 4.5 berikut

Gambar 4.5 Contoh *error*

Dalam Gambar 4.5 diatas dapat dilihat bahwa isi pesan adalah “ivan” yang terdiri dari empat karakter, namun pada kolom hasil enkripsi terdapat delapan karakter. Blok pertama yaitu “iv” mempunyai nilai ASCII 105118, dengan menggunakan algoritma Luc, didapatkan hasil akhir 63345, jika nilai tersebut di konversi menjadi dua karakter berpasangan hasilnya adalah 63 dan

345, 63 adalah karakter tanda tanya (?) sedangkan 345 adalah ř. Sedangkan untuk blok kedua yaitu “an” mempunyai nilai ASCII 97110, jika dihitung dengan menggunakan algoritma Luc, didapatkan hasil akhir 40353, jika nilai tersebut dikonversi kedalam bentuk karakter berpasangan, maka hasilnya adalah 40 dan 353 dimana 40 adalah “(“ dan 353 adalah š. Jika hasil perhitungan dibandingkan dengan hasil visualisasi program, maka terdapat dua pasang karakter sebagai karakter tambahan, hal ini disebabkan karena program tidak dapat membaca dengan baik karakter yang berada diluar batasan nilai ASCII yaitu 0 – 255.

Faktor lain yang berpengaruh dalam algoritma Luc adalah penulis menemukan bahwa terdapat minimal sebuah pasangan huruf yang jika di enkripsi atau di dekripsi, akan menghasilkan nilai 0. Selama pengujian program ini penulis menemukan bahwa pasangan huruf OK tidak bisa di enkripsi dengan menggunakan kunci publik (7, 11327), hal ini disebabkan pada perhitungan terakhir dari pasangan huruf tersebut akan menghasilkan nilai 0.

Faktor berikutnya adalah penentuan bilangan prima *p* dan *q*, dimana semakin besar bilangan prima yang dipilih maka kemungkinan kunci public dan kunci privat akan semakin banyak, dan hal ini merupakan kekuatan dari algoritma Luc, namun dengan pemilihan bilangan prima *p* dan *q* yang besar akan mempengaruhi waktu yang dibutuhkan untuk proses enkripsi atau dekripsi.

Berikut ini adalah beberapa contoh hasil visualisasi aplikasi pada telepon seluler Motorola XT530 yang dibuat oleh penulis dengan menggunakan dua kunci public yang berbeda :

Gambar 4.6 Tampilan Enkripsi dan Dekripsi pada Motorola XT530

Dari Gambar 4.6 diatas, dapat dilihat bahwa algoritma Luc dapat diimplementasikan dalam bidang telekomunikasi terutama telepon seluler berbasis Android dengan menggunakan huruf kapital. Gambar 4.6 (a) menggunakan kunci public (*e*,*N*) = (23, 16441) dan (b) menggunakan kunci public (*e*, *N*) = (29, 16441). Proses dekripsi dijalankan pada Gambar 4.6 (c) dan (d) dimana pada gambar (c) kunci yang diisikan adalah $p = 41$, $q = 401$, $e = 23$. Sedangkan untuk Gambar 4.6 (d) menggunakan kunci $p = 41$, $q = 401$, $e = 29$.

BAB 5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan dapat ditarik kesimpulan sebagai berikut :

a. Algoritma Luc dapat diimplementasikan pada telepon seluler berbasis Android, sebagai sistem keamanan pesan.
b. Aplikasi yang dibuat oleh penulis mampu melakukan dekripsi dengan baik untuk huruf kapital.

5.2 Saran

Saran yang dapat disampaikan penulis yang bertujuan memntu pengembangan aplikasi ini adalah :

a. Menyempurnakan aplikasi ini agar dapat melakukan enkripsi dan dekripsi dengan baik terutama untuk karakter kecil;
b. Menambahkan tabel dalam *database* untuk menyimpan kunci privat yang telah dibangkitkan, hal ini disebabkan penulis belum mampu menambahkan tabel dalam *database* berkaitan dengan keterbatasan kemampuan penulis dalam bahasa *java*;
c. Menambahkan beberapa fitur yang dapat mempermudah penggunaan aplikasi, seperti *auto insert* nomor telepon, *auto insert* kunci, *copy* dan *paste* teks, dan fitur-fitur lain;
d. Tampilan aplikasi masih *monoton* sehingga masih perlu banyak tambahan.

DAFTAR PUSTAKA

Dwi, A.K. 2012. *Penerapan Algoritma Vigenere Cipher pada Aplikasi SMS Android*. Bandung: Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung.

Hildebrand, A.J. 2011. *Elementary Number Theory Definitions and Theorems*. Illinois: Departement of Matematics University of Illinois.

Kelly, S. 2006. *Analisis Perbandingan Teorema Lucas-Lehmer dan Teorema Pocklington Dalam Uji Primalitas.* Skripsi. BINUS.

Kromodimoeljo, S. 2009. *Teori & Aplikasi Kriptografi*, SPK IT CONSULTING.

Lesmana, I. 2010. *Aplikasi Pembangkit Kunci Berbasis Modifikasi Bilangan Fibonacci Pada Sandi Vigenere*, Jakarta : Universitas Pembangunan Nasional Veteran Jakarta.

Menezes, Alfred Paul Van Oorschot and Vanston Sean, 1996. "*Handbook of Applied Cryptography*", USA: CRC Press, Inc.

Network Associates, Inc. 1998. *An Introduction to Cryptography.* Santa Clara: Network Associates, Inc.

Riyanto, M.Z., 2007. *Pengamanan Pesan Rahasia Menggunakan Algoritma Kriptografi Elgamal Atas Grup Pergandaan Zp**. Skripsi. Yogyakarta : Universitas Gajah Mada.

Rochmayanti, M. 2010. *Prototipe Aplikasi Pengiriman Data Melalui MMS Berbasis Java Dengan Digital Ssignature Menggunakan Algoritma RSA*. Skripsi. UNIKOM.

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

Saputra, R., Yismianto, B., dan Suhartono. 2006. *Kriptografi Teks Dengan Menggunakan Algoritma LUC*. Skripsi. Semarang : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Diponegoro.

Schneier, B. 1996. *Applied Cryptography*: *Protocols, Algorithms, and Source Code in C*. Canada : John Wiley & Sons Inc.

Smith, J.P. dan Lennon M.J.J.1993. *LUC: A new public key system*. Auckland: The University of Auckland.

Wang, C.T, Chang, C.C and Lin, C.H. 1999. *A Method for Computing Lucas Sequence*. An International Journal Computers & Matematics with Aplications.

Lampiran A

Main.java

```
package com.kiplink.luc;

import android.os.Bundle;
import android.app.Activity;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import java.lang.Math;

public class Main extends Activity {
  private Button btnSend, btnEnk;
  private EditText txtPhoneNo;
  private EditText txtMessage;
  private EditText txtKeyE;
  private EditText txtKeyN;
  private EditText txtHasil;

  @Override
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    btnSend = (Button)findViewById (R.id.btnSend);
    btnEnk = (Button)findViewById (R.id.btnEnk);
    txtPhoneNo = (EditText)findViewById (R.id.txtNum);
    txtMessage = (EditText) findViewById (R.id.txtMessage);
    txtKeyE = (EditText) findViewById (R.id.txtKeyE);
    txtKeyN = (EditText) findViewById (R.id.txtKeyN);
    txtHasil = (EditText) findViewById (R.id.txtHasil);
    btnEnk.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
    // TODO Auto-generated method stub
      try{
        String str2="";
        int par = 0;
        String st = "";
        int bagi=0;
        int pr1 = 0;
```

```
   // jika jumlah teks ganjil, maka ditambahkan karakter spasi
   pada  akhir teks
   String tes = (txtMessage.getText().toString());
   int kE = new Integer(txtKeyE.getText().toString());
   int kN = new Integer(txtKeyN.getText().toString());

 if (tes.length() % 2 != 0){
    tes += " ";
 }
 //atur blok dan conversi dalam ascii
 for (int k=0;k<=tes.length();k++){
   long D;
   str2 = blok(k, tes);
   //str2 adalah hasil pengaturan blok dan conversi ascii
   //proses enkripsi
   long m = new Long(str2);
   D = (m * m) - 4;
   String tes2 = Integer.toString(LUC2(m, kE, kN));
   // proses enkripsi
   //tampung hasil enkripsi dalam ascii
   //convert ascii dalam karakter
   for (int l=0;l<tes2.length(); l++){
     par = new Integer (tes2.length() / 2);

   //jika digit ascii ganjil
   if (tes2.length()%2 != 0){
      if
      (tes2.length()<(txtKeyN.getText().toString().length())){
      int h = new Integer(txtKeyN.getText().toString().length()
      -     tes2.length());
       for (int f = 1;f<=h;f++){
        tes2 = "0" + tes2;
       }
       }
       pr1 = ascii2(tes2, par);
       // ambil 2 digit ascii hasil enkripsi
       if (pr1<32){
       bagi = 3;
   } else {
     bagi = 2;
   }
st += swit(bagi, tes2, par);
//karakter pertama 3 digit atau 2 digit
txtHasil.setText(st);
}
//jika digit ascii genap
if (tes2.length()%2 == 0){
  st += genap(tes2, par);
  txtHasil.setText(st);
}
tes2 = "";
str2="";
```

```
      par = 0;
      l = l + 1;
   };
      k = k+1;
   };
   }catch (Exception e){
   }
   }
   });
   btnSend.setOnClickListener(new View.OnClickListener() {

      @Override
      public void onClick(View v) {
      // TODO Auto-generated method stub

      String phoneNo = txtPhoneNo.getText().toString();
      String message = txtHasil.getText().toString();

      if(phoneNo.length()>0 && message.length()>0)
         sendSMS(phoneNo, message);
      else
        Toast.makeText(getBaseContext(), "Masukan No dan pesan",
        Toast.LENGTH_SHORT).show();
      }
   });
   }
private boolean MenuChoice(MenuItem item)
   {
   switch (item.getItemId()){
   case R.id.krm_sms:
   startActivity(new Intent(Main.this, Main.class));
   return true;
   case 1:
   Toast.makeText(this, "Tentang", Toast.LENGTH_SHORT).show();
   return true;
   case R.id.keluar:
   Intent exit = new Intent(Intent.ACTION_MAIN);
   exit.addCategory(Intent.CATEGORY_HOME);
exit.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
   Main.this.finish();
   startActivity(exit);
   return true;
   case R.id.menu_settings:
   startActivity(new Intent(Main.this, Listkey.class));
   return true;
   }
   return false;
   }
   @Override
public boolean onCreateOptionsMenu (Menu menu){
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
```

```
}
@Override
public boolean onOptionsItemSelected(MenuItem item){
  return MenuChoice(item);
}
private void sendSMS(String phoneNumber, String message){
  String SENT = "SMS_SENT";
  String DELIVERED = "SMS_DELIVERED";

  PendingIntent sentPI = PendingIntent.getBroadcast(this, 0,
  new Intent(SENT), 0);
   PendingIntent deliveredPI = PendingIntent.getBroadcast(this, 0,new
   Intent(DELIVERED), 0);
  //---when the SMS has been sent---
  registerReceiver(new BroadcastReceiver() {

    @Override
    public void onReceive(Context arg0, Intent arg1) {
    // TODO Auto-generated method stub
    switch (getResultCode())
    {
      case Activity.RESULT_OK:
        Toast.makeText(getBaseContext(), "SMS Terkirim",
        Toast.LENGTH_SHORT).show();
      break;
      case android.telephony.SmsManager.RESULT_ERROR_GENERIC_FAILURE:
        Toast.makeText(getBaseContext(), "Generic failur",
        Toast.LENGTH_SHORT).show();
      break;
      case android.telephony.SmsManager.RESULT_ERROR_NO_SERVICE:
        Toast.makeText(getBaseContext(), "NO SERVICE",
        Toast.LENGTH_SHORT).show();
      break;
      case android.telephony.SmsManager.RESULT_ERROR_NULL_PDU:
        Toast.makeText(getBaseContext(), "Null PDU",
        Toast.LENGTH_SHORT).show();
      break;
      case android.telephony.SmsManager.RESULT_ERROR_RADIO_OFF:
        Toast.makeText(getBaseContext(), "Radio OFF",
        Toast.LENGTH_SHORT).show();
      break;
    }
  }
}, new IntentFilter(SENT));

//---when the SMS has been delivered---
registerReceiver(new BroadcastReceiver() {

@Override
public void onReceive(Context arg0, Intent arg1) {
// TODO Auto-generated method stub
switch (getResultCode()){
```

```
case Activity.RESULT_OK:
  Toast.makeText(getBaseContext(), "SMS Terkirim",
  Toast.LENGTH_SHORT).show();
break;
case Activity.RESULT_CANCELED:
  Toast.makeText(getBaseContext(), "SMS GALAU",
  Toast.LENGTH_SHORT).show();
break;
}
}
},new IntentFilter(DELIVERED));
android.telephony.SmsManager sms =
android.telephony.SmsManager.getDefault();
sms.sendTextMessage(phoneNumber, null, message, sentPI,
deliveredPI);
}

public static String blok(int k, String tes){
  // atur blok berisi 2 karakter
  char m1 = tes.charAt(k);
  char m2 = tes.charAt(k+1);
  int im1 = (int)m1;
  int im2 = (int)m2;
  String sm1 = (Integer.toString(im1));
  String sm2 = (Integer.toString(im2));
  String td = sm1 + sm2;
  return td;
}

public static int LUC(long D, long m, long i, int b, int c, int kE,
long E, int kN){ //algoritma LUC
  int f[] = new int [kN];
  int n = 0;
  for(int a=0; a<=kE; a++){
    if (a<2){
      double d1= Math.ceil(Math.sqrt(D));
      double x1 = (m + d1)/2;
      double x2 = (m - d1)/2;
      int y1 = (int) Math.pow(x1, a);
      int y2 = (int) Math.pow(x2, a);
      int d2 = (y1 + y2);
      f[a] = d2 % kN;

    }
    if (a>=2){
      int x = a - 1;
      i=(long)(m*(f[x]));
      b = a - 2;
      c=f[b];
      E = (i - c) % kN;
      f[a] = (int) E;
      n=a;
```

```
    }
  }
  return f[n];
}

public static int LUC2(long m, int kE, int kN){ //Algoritma Luc2
  int f[] = new int [kN];
  int x = 0;
  int k[] = new int[kE];
  k[0] = 2;
  while (kE != 1){
    x++;
      if (kE % 2 == 1){
        kE = kE - 1;
        k[x]=1;
      }else{
        kE = kE/2;
      k[x] = 0;
    }
  }
  f[kE] = (int)m;
  int j = 0;
  f[j] = 2;
  long g=0;
  long h=0;
  while (x>0){
    if (k[x]==0){
      g =(long) (f[kE] * (long)f[kE] - 2)%kN;
      h = (long)(f[kE] * (long)f[j] - m) % kN;
      f[kE] =(int) g;
      f[j] = (int)h;
    }else{
      g=(((long)m * (long)f[kE]) - f[j]) % kN;
      f[j] = f[kE];
      f[kE] = (int)g;
    }
    x--;
  }
  return f[kE];
}

public static String swit(int bagi, String tes2, int par){
  //convert jika ascii ganjil
  int pr1 = 0;
  int pr2 = 0;
  int pr3 = 0;
  int pr4 = 0;
  String has = "";
  String has2 = "";
  String has1 = "";
  String st = "";
  switch(bagi){
```

```
   case 2:
    //ambil 2 digit pertama sebagai karakter pertama
    for (int y=0;y<par;y++){
    char ci1 = tes2.charAt(y);
    String sci1 = String.valueOf(ci1);
    has = has + sci1;
    int hsl1 = new Integer(String.valueOf(has));
    pr1 = hsl1;
  }
  //digit berikutnya dianggap karapter kedua
  for (int y23=par;y23<tes2.length();y23++){
    char i3 = tes2.charAt(y23);
    String sci3 = String.valueOf(i3);
    has1 = has1 + sci3;
    int hsl2 = new Integer(String.valueOf(has1));
    pr2 = hsl2;
  }
  st += (String.valueOf((char)pr1)) + (String.valueOf((char)pr2));
  break;
  case 3:
    //ambil 3 digit pertama
    for (int y4=0;y4<bagi;y4++){
      char y3 = tes2.charAt(y4);
      String sci3 = String.valueOf(y3);
      has1 = has1 + sci3;
      int hsl2 = new Integer(String.valueOf(has1));
      pr3 = hsl2;
    }
  //digit berikutnya adalah karakter kedua
  for (int y5=bagi;y5<tes2.length();y5++){
    char y3 = tes2.charAt(y5);
    String sci3 = String.valueOf(y3);
    has2 = has2 + sci3;
    int hsl3 = new Integer(String.valueOf(has2));
    pr4 = hsl3;
  }
  st += (String.valueOf((char)pr3)) + (String.valueOf((char)pr4));
  break;
  }
return st;
}

public static String genap(String tes2, int par){
  //convert ascii genap
  int pre1 = 0;
  int pre2 = 0;
  String has = "";
  String has1 = "";
  String st = "";
  for (int y=0;y<par;y++){
    //karakter pertama adalah indeks digit 0 sampai hasil bagi - 1
    char cy1 = tes2.charAt(y);
```

```
    String sci1 = String.valueOf(cy1);
    has = has + sci1;
    int hsl1 = new Integer(String.valueOf(has));
    pre1 = hsl1;
  }
  //karakter kedua
  for (int y22=par;y22<tes2.length();y22++){
    char y3 = tes2.charAt(y22);
    String sci3 = String.valueOf(y3);
    has1 = has1 + sci3;
    int hsl2 = new Integer(String.valueOf(has1));
    pre2= hsl2;
  }
   st += (String.valueOf((char)pre1)) + (String.valueOf((char)pre2));
  return st;
  }
public static int ascii2(String tes2, int par){
  //ambil 2 digit ascii hasil dekripsi
  String has = "";
  int pr1 = 0;
  for (int y=0;y<par;y++){
    char ci1 = tes2.charAt(y);
    char ci2 = tes2.charAt(y+1);
    String sci1 = String.valueOf(ci1);
    String sci2 = String.valueOf(ci2);
    has = sci1 + sci2;
    int hsl1 = new Integer(String.valueOf(has));
    pr1 = hsl1;
    y = y + 1;
  }
  return pr1;
  }
}
```

InboxActivity.java

```
package com.kiplink.luc;

import android.app.Activity;
import android.content.Intent;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
```

```
import android.widget.Toast;
import android.widget.Button;

public class InboxActivity extends Activity {
  ListView lv;
  String [] m = { "" };
  String pesanTerpilih;
  @Override
  public void onCreate(Bundle savedInstanceState){
  super.onCreate(savedInstanceState);
  setContentView(R.layout.inbox);
  lv = (ListView) findViewById(R.id.smsList);
  Button sms=(Button)findViewById(R.id.btn_new);
  sms.setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View v) {
  // TODO Auto-generated method stub
  Intent i=new Intent(InboxActivity.this, Main.class);
  startActivity(i);
  }
  });
  Uri uriSMSURI = Uri.parse("content://sms/inbox");
   Cursor cur = getContentResolver().query(uriSMSURI, null, null,
   null, null);

  m = new String[cur.getCount()];
  int g = 0;
  while (cur.moveToNext()){
  m[g++] = cur.getString(11);
  Log.i("Pengirim : ", cur.getString(2));
  Log.i("Isi Pesan : ", cur.getString(11));
  }
   ArrayAdapter<String> a = new
   ArrayAdapter<String>(getApplicationContext(),
   android.R.layout.simple_list_item_1, m);
  lv.setAdapter(a);
  lv.setOnItemClickListener(new OnItemClickListener(){
  @Override
     public void onItemClick(AdapterView<?> arg0, View arg1, int
     arg2, long arg3){
      // TODO Auto-generated method stub
      Intent i = new Intent(InboxActivity.this, BacaActivity.class);
      i.putExtra("pesan", m[arg2]);
      startActivity(i);
     }
  });

}

private boolean MenuChoice(MenuItem item)
{
  switch (item.getItemId()){
```

```
    case R.id.krm_sms:
    startActivity(new Intent(InboxActivity.this, Main.class));
    return true;
    case 1:
    Toast.makeText(this, "Tentang", Toast.LENGTH_SHORT).show();
    return true;
    case R.id.keluar:
    Intent exit = new Intent(Intent.ACTION_MAIN);
    exit.addCategory(Intent.CATEGORY_HOME);
    exit.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    InboxActivity.this.finish();
    startActivity(exit);
    return true;
    case R.id.menu_settings:
    startActivity(new Intent(InboxActivity.this, Listkey.class));
    return true;
  }
  return false;
  }
  @Override
  public boolean onCreateOptionsMenu (Menu menu){
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
  }
  @Override
  public boolean onOptionsItemSelected(MenuItem item){
    return MenuChoice(item);
  }
}
```

BacaActivity.java

```
package com.kiplink.luc;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import java.lang.Math;

public class BacaActivity extends Activity {

  public String hasil, hasil2, var_plain, dataNo;
  EditText txtNo, txtPesan, txtHasil, txtkunci, key_p, key_q,
   key_e, tester, txtAscii, txt;
  public int var_key_d;
  public int var_key_N;
```

```
  @Override
  public void onCreate(Bundle savedInstanceState){
  super.onCreate(savedInstanceState);
  setContentView(R.layout.baca_sms);
  Bundle extras = getIntent().getExtras();
  if (extras == null){
  return;
}
String pesan = extras.getString("pesan");
if (pesan != null){
  EditText isi_pesan = (EditText) findViewById(R.id.isi_pesan);
  isi_pesan.setText(pesan);
}
key_p = (EditText)findViewById(R.id.key_p);
key_q = (EditText)findViewById(R.id.key_q);
key_e = (EditText)findViewById(R.id.key_e);
txtPesan = (EditText)findViewById(R.id.isi_pesan);
tester = (EditText)findViewById(R.id.tester);
txtHasil = (EditText)findViewById(R.id.txtHasil);

Button btnDekrip = (Button) findViewById(R.id.btnDekrip);
btnDekrip.setOnClickListener(new View.OnClickListener() {

  @Override
  public void onClick(View view) {
  // TODO Auto-generated method stub
  try{
    String td = "";
    String st = "";
    int d=0;
    long D =0;
    int par = 0;
    int pr1 = 0;
    int bagi = 0;
    int lcm=0;
    String ft = "";
    String teks = (txtPesan.getText().toString());
    int kP = new Integer(key_p.getText().toString());
    int kQ = new Integer(key_q.getText().toString());
    int kN = (kP * kQ);
    //atur blok
    for (int t=0; t<=teks.length();t++){
      td = blok(t, teks);
      //td adalah nilai ascii dari 2 karakter
      //bangkitkan kunci d
      //legendre p & q (-1, 0 ,1)
      int ascii = new Integer(td);
      D = new Long((ascii * ascii) - 4);
      int rp = kP - (LP(D, kP));
      int rq = kQ - (LQ(D, kQ));
      //LCM
      lcm = LCM(rp, rq);
```

```
      int e = new Integer(key_e.getText().toString());
      //invers de = 1 mod S(N)
      d = de(e, lcm);
      long m = new Long(td);
      //proses dekripsi
      String tes2 = Integer.toString(LUC2(m, d, kN));
      //convert ascii dalam karakter
      for (int l=0;l<tes2.length(); l++){
      par = new Integer (tes2.length() / 2);

      //jika digit ascii ganjil
      if (tes2.length()%2 != 0){
      //ambil 2 digit pertama dari nilai ascii
      pr1 = ascii2(tes2, par);
      //bandingkan 2 digit ascii yg didapatkan
      //jika 2 digit pertama lebih kecil dari 32, maka ambil pilihan
      3
      //jika 2 digit pertama lebih besar atau sama dengan 32 maka
      ambil pilihan 2
      if (pr1<32){
        bagi = 3;
      } else {
        bagi = 2;
      }
    st += swit(bagi, tes2, par);
    //cetak hasil konversi dalam hasil
    txtHasil.setText(st);
    }
    //jika digit ascii genap maka panjang ascii dibagi 2
    if (tes2.length()%2 == 0){
      st += genap(tes2, par);
      //cetak hasil konversi dalam hasil
      txtHasil.setText(st);
    }
    //kosongkan variabel sebagai default
    tes2 = "";
    td="";
    par = 0;
    l = l + 1;
  };
  t = t+1;
};
} catch (Exception e){
}
}
});

 }
private boolean MenuChoice(MenuItem item)
{
  switch (item.getItemId()){
    case R.id.krm_sms:
```

```
        startActivity(new Intent(BacaActivity.this, Main.class));
        return true;
      case 1:
        Toast.makeText(this, "Tentang", Toast.LENGTH_SHORT).show();
        return true;
      case R.id.keluar:
        Intent exit = new Intent(Intent.ACTION_MAIN);
        exit.addCategory(Intent.CATEGORY_HOME);
        exit.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        BacaActivity.this.finish();
        startActivity(exit);
        return true;
      case R.id.menu_settings:
        startActivity(new Intent(BacaActivity.this, Listkey.class));
        return true;
    }
    return false;
}
public static long modExp(long a, long b, long p) {
    long rval = 1;
    while(b > 0) {
      if((b & 1) == 1) /* if b is odd */
      rval = (rval * a) % p;
      b >>= 1;
      a = (a * a) % p;
    }
    return rval;
}
public static long trueMod(long a, long b) {
    /*fixes Java % feature when dealing with negative numbers*/
    if (a >= 0)
    return a % b;
    else
    return (-a * (b-1)) % b;
}
public static String blok(int t, String teks){
    // atur blok berisi 2 karakter
    char m1 = teks.charAt(t);
    char m2 = teks.charAt(t+1);
    int im1 = (int)m1;
    int im2 = (int)m2;
    String sm1 = (Integer.toString(im1));
    String sm2 = (Integer.toString(im2));
    if (sm1.length()==1){
      sm1 = "0" + sm1;
    }
    if (sm2.length()==1){
      sm2 = "0" + sm2;
    }
    String td = sm1 + sm2;
    return td;
}
```

```
public static int de (int e, int lcm){
  // invers de = 1 mod S(N)
  int d=0;
  for (int a=1;a<lcm;a++){
    int de = (a * lcm) + 1;
    if (de % e == 0){
    d = de/e;
  break;
}
};
return d;
}
public static int LP(long D, int kP){
  // legendre simbol P
  int xp = 0;
  if(trueMod(D, kP) == 0){
    xp = 0;
  }
  long rval = modExp(D, (kP-1)/2, kP);
  if(rval == 1){
    xp = 1;
  } else if (rval == 0){
    xp = 0;
  }else{
    xp = -1;
  }
return xp;
}
public static int LQ(long D, int kQ){ //legendre simbol Q
  int xq = 0;
  if(trueMod(D, kQ) == 0){
    xq = 0;
  }
  long rval = modExp(D, (kQ-1)/2, kQ);
  if(rval == 1){
    xq = 1;
  } else if (rval == 0){
    xq = 0;
  }else{
    xq = -1;
  }
return xq;
}
public static int LUC(long D, long m, long i, int b, int c, int d,
long E, int kN){
  //algoritma LUC
  int f[] = new int [kN];
  int n = 0;
  for(int a=0; a<=d; a++){
    if (a<2){
      double d1= Math.ceil(Math.sqrt(D));
      double x1 = (m + d1)/2;
```

```
      double x2 = (m - d1)/2;
      int y1 = (int) Math.pow(x1, a);
      int y2 = (int) Math.pow(x2, a);
      int d2 = (y1 + y2);
      f[a] = d2 % kN;
    }
  if (a>=2){
    int x = a - 1;
    i=(long)(m*(f[x]));
    b = a - 2;
    c=f[b];
    E = (i - c) % kN;
    f[a] = (int) E;
    n=a;
  }
}
return f[n];
}
public static int LUC2(long m, int d, int kN){ // luc 2
  int f[] = new int [kN];
  int x = 0;
  int k[] = new int[d];
  k[0] = 2;
  while (d != 1){
    x++;
  if (d % 2 == 1){
    d = d - 1;
    k[x]=1;
  }else{
    d = d/2;
    k[x] = 0;
  }
  }
    f[d] = (int)m;
    int j = 0;
    f[j] = 2;
    long g=0;
    long h=0;
    while (x>0){
      if (k[x]==0){
        g = ((long)f[d] * (long)f[d] - 2)%kN;
        h = ((long)f[d] * (long)f[j] - m) % kN;
        f[d] =(int) g;
        f[j] = (int)h;
      }else{
        g=((long)m * (long)f[d] - (long)f[j]) % kN;
        f[j] = f[d];
        f[d] = (int)g;
      }
    x--;
    }
  return f[d];
```

```
}
public static String swit(int bagi, String tes2, int par){
  //convert jika ascii ganjil
  int pr1 = 0;
  int pr2 = 0;
  int pr3 = 0;
  int pr4 = 0;
  String has = "";
  String has2 = "";
  String has1 = "";
  String st = "";
  switch(bagi){
    case 2:
    //ambil 2 digit pertama sebagai karakter pertama
    for (int y=0;y<par;y++){
      char ci1 = tes2.charAt(y);
      String sci1 = String.valueOf(ci1);
      has = has + sci1;
      int hsl1 = new Integer(String.valueOf(has));
      pr1 = hsl1;

  }
  //digit berikutnya dianggap karapter kedua
  for (int y23=par;y23<tes2.length();y23++){
    char i3 = tes2.charAt(y23);
    String sci3 = String.valueOf(i3);
    has1 = has1 + sci3;
    int hsl2 = new Integer(String.valueOf(has1));
    pr2 = hsl2;
  }
  st += (String.valueOf((char)pr1)) + (String.valueOf((char)pr2));
  break;
  case 3:
  //ambil 3 digit pertama
  for (int y4=0;y4<bagi;y4++){
    char y3 = tes2.charAt(y4);
    String sci3 = String.valueOf(y3);
    has1 = has1 + sci3;
    int hsl2 = new Integer(String.valueOf(has1));
    pr3 = hsl2;
  }
  //digit berikutnya adalah karakter kedua
  for (int y5=bagi;y5<tes2.length();y5++){
    char y3 = tes2.charAt(y5);
    String sci3 = String.valueOf(y3);
    has2 = has2 + sci3;
    int hsl3 = new Integer(String.valueOf(has2));
    pr4 = hsl3;
  }
  st += (String.valueOf((char)pr3)) + (String.valueOf((char)pr4));
  break;
}
```

```
return st;
}

public static String genap(String tes2, int par){
  //convert ascii genap
  int pre1 = 0;
  int pre2 = 0;
  String has = "";
  String has1 = "";
  String st = "";
  for (int y=0;y<par;y++){
    //karakter pertama adalah indeks digit 0 sampai hasil bagi - 1
    char cy1 = tes2.charAt(y);
    String sci1 = String.valueOf(cy1);
    has = has + sci1;
    int hsl1 = new Integer(String.valueOf(has));
    pre1 = hsl1;
  }
  //karakter kedua
  for (int y22=par;y22<tes2.length();y22++){
    char y3 = tes2.charAt(y22);
    String sci3 = String.valueOf(y3);
    has1 = has1 + sci3;
    int hsl2 = new Integer(String.valueOf(has1));
    pre2= hsl2;
  }
   st += (String.valueOf((char)pre1)) + (String.valueOf((char)pre2));
  return st;
}
public static int ascii2(String tes2, int par){
  //ambil 2 digit ascii hasil dekripsi
  String has = "";
  int pr1 = 0;
  for (int y=0;y<par;y++){
    char ci1 = tes2.charAt(y);
    char ci2 = tes2.charAt(y+1);
    String sci1 = String.valueOf(ci1);
    String sci2 = String.valueOf(ci2);
    has = sci1 + sci2;
    int hsl1 = new Integer(String.valueOf(has));
    pr1 = hsl1;
    y = y + 1;
  }
  return pr1;
}
public static int LCM(int rp, int rq){ // LCM
  int lcm = 0;
  for (int a=1; a<rq;a++){
    lcm = a * rq;
    if (lcm % rp == 0){
      break;
    }
```

```
  };
  return lcm;
  }
}
```

ListKey.java

```
package com.kiplink.luc;

import android.os.Bundle;
import android.app.ListActivity;
import android.view.Menu;
import android.content.Context;
import android.content.Intent;
import android.database.Cursor;
import android.view.MenuItem;
import android.view.MenuInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.LayoutInflater;
import android.widget.CursorAdapter;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.RadioGroup;
import android.widget.TextView;

public class Listkey extends ListActivity {
  public final static String ID_EXTRA = "com.kiplink.luc._id";
  Cursor model=null;
  AlmagAdapter adapter=null;
  EditText nama=null;
  EditText key_e = null;
  EditText key_N = null;
  RadioGroup jekel = null;
  AlmagHelper helper=null;
  @Override
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.listkey);

    helper=new AlmagHelper(this);

    nama=(EditText) findViewById(R.id.nama);
    key_e=(EditText) findViewById(R.id.key_e);
    key_N=(EditText) findViewById(R.id.key_N);
    jekel=(RadioGroup) findViewById(R.id.jekel);

    model=helper.getAll();
    startManagingCursor(model);
    adapter=new AlmagAdapter(model);
```

```
    setListAdapter(adapter);
  }
  @Override
  public void onDestroy(){
    super.onDestroy();
    helper.close();
  }
  @Override
public void onListItemClick(ListView list, View view, int posotion,
long id){
  Intent i = new Intent(Listkey.this, Detailkey.class);
  i.putExtra(ID_EXTRA, String.valueOf(id));
  startActivity(i);
}
private View.OnClickListener onSave=new View.OnClickListener() {
  public void onClick(View v){
    String type=null;
    switch (jekel.getCheckedRadioButtonId()){
      case R.id.pria:
      type="Pria";
      break;
      case R.id.perempuan:
      type="Perempuan";
    break;
    }
   helper.insert(nama.getText().toString(),
   key_e.getText().toString(), key_N.getText().toString(), type);
  model.requery();
  }
};
class AlmagAdapter extends CursorAdapter{
  AlmagAdapter(Cursor c){
    super (Listkey.this, c);
  }
  @Override
public void bindView(View row, Context ctxt, Cursor c){
  AlmagHolder holder=(AlmagHolder)row.getTag();
  holder.populateFrom(c, helper);
}
@Override
public View newView(Context ctxt, Cursor c, ViewGroup parent){
  LayoutInflater inflater=getLayoutInflater();
  View row=inflater.inflate(R.layout.row, parent, false);
  AlmagHolder holder=new AlmagHolder(row);
  row.setTag(holder);
  return(row);
}
}
static class AlmagHolder {
  private TextView nama=null;
  private TextView key_e=null;
  private TextView key_N=null;
```

```
  private ImageView icon=null;
  private View row=null;

  AlmagHolder(View row){
    this.row=row;
    nama=(TextView)row.findViewById(R.id.title);
    key_e=(TextView)row.findViewById(R.id.key_e);
    key_N=(TextView)row.findViewById(R.id.key_N);
    icon=(ImageView)row.findViewById(R.id.icon);
  }
void populateFrom(Cursor c, AlmagHelper helper){
  nama.setText(helper.getNama(c));
  key_e.setText("key e : "+helper.getKey_e(c));
  key_N.setText("key N : "+helper.getKey_N(c));
  if (helper.getJekel(c).equals("Pria")){
  icon.setImageResource(R.drawable.boy);
}
else if (helper.getJekel(c).equals("Perempuan")){
  icon.setImageResource(R.drawable.girl);
}
}
}
@Override
public boolean onCreateOptionsMenu (Menu menu){
  new MenuInflater(this).inflate(R.menu.newkey, menu);
  return (super.onCreateOptionsMenu(menu));
}
@Override
public boolean onOptionsItemSelected(MenuItem item){
  switch (item.getItemId()){
  case R.id.add:
    startActivity(new Intent(Listkey.this, Detailkey.class));
    return true;
  case R.id.privacy:
    startActivity(new Intent(Listkey.this, Privat.class));
    return true;
  }

  return(super.onOptionsItemSelected(item));
}
}
```

DetailKey.java

```
package com.kiplink.luc;
import android.app.Activity;
import android.database.Cursor;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RadioGroup;
```

```
public class Detailkey extends Activity{
  EditText nama = null;
  EditText key_e = null;
  EditText key_N = null;
  RadioGroup jekel = null;
  AlmagHelper helper = null;
  String almagId = null;

  @Override
  public void onCreate(Bundle savedInstanceState){
    super.onCreate(savedInstanceState);
    setContentView(R.layout.detailkey);

    helper = new AlmagHelper(this);
    nama = (EditText)findViewById(R.id.nama);
    key_e = (EditText)findViewById(R.id.key_e);
    key_N = (EditText)findViewById(R.id.key_N);
    jekel = (RadioGroup)findViewById(R.id.jekel);
    Button save = (Button)findViewById(R.id.btnSave);
    save.setOnClickListener(onSave);
    almagId=getIntent().getStringExtra(Listkey.ID_EXTRA);
    if (almagId != null){
      load();
    }
  }
@Override
public void onDestroy(){
  super.onDestroy();
  helper.close();
}
private void load(){
  Cursor c = helper.getbyId(almagId);
  c.moveToFirst();
  nama.setText(helper.getNama(c));
  key_e.setText(helper.getKey_e(c));
  key_N.setText(helper.getKey_N(c));
  if (helper.getJekel(c).equals("Pria")){
  jekel.check(R.id.pria);
}
else if (helper.getJekel(c).equals("Perempuan")){
  jekel.check(R.id.perempuan);
}
  c.close();
}
private View.OnClickListener onSave=new View.OnClickListener() {

  @Override
  public void onClick(View v) {
  // TODO Auto-generated method stub
  String type=null;
  switch (jekel.getCheckedRadioButtonId()){
```

```
    case R.id.pria:
      type = "Pria";
      break;
    case R.id.perempuan:
      type = "Perempuan";
      break;
  }
  if (almagId == null){
   helper.insert(nama.getText().toString(),
   key_e.getText().toString(), key_N.getText().toString(), type);
  }
  else {
    helper.update(almagId, nama.getText().toString(),
    key_e.getText().toString(), key_N.getText().toString(), type);
  }
  finish();
}
};
}
```

Privat.java

```
package com.kiplink.luc;

import android.app.Activity;
import android.content.Intent;
import android.database.Cursor;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import java.util.Random;

public class Privat extends Activity{
  private EditText keyP, keyQ;
  private Button btnPriv, btnSave;
  private TextView pubKey, privKey, NKey;
  AlmagHelper helper=null;
  String almagId = null;
  @Override
  public void onCreate(Bundle savedInstanceState){
   super.onCreate(savedInstanceState);
   setContentView(R.layout.privat);

   keyP = (EditText)findViewById(R.id.keyP);
   keyQ = (EditText)findViewById(R.id.keyQ);
   pubKey = (TextView)findViewById(R.id.pubKey);
   privKey = (TextView)findViewById(R.id.privKey);
   NKey = (TextView)findViewById(R.id.NKey);
```

```
   btnPriv = (Button)findViewById(R.id.btnPriv);
   btnSave = (Button)findViewById(R.id.btnSave);
  btnPriv.setOnClickListener(new View.OnClickListener() {

  @Override
  public void onClick(View v) {
   // TODO Auto-generated method stub
   try {
   int p =new Integer(keyP.getText().toString());
   int q =new Integer(keyQ.getText().toString());
   int p1 =  p - 1;
   int p2 =  p + 1;
   int q1 =  q - 1;
   int q2 =  q + 1;
   Random rdm = new Random();
   int a = 0;
   //nilai N
   int hitung= p * q;
   NKey.setText(Integer.toString(hitung));
   //nilai key e
   int bil = 0;
   String bilangan = "";
   final int eu[] = new int[p1];
   Random R = new Random();
   int s = 0;

  for (int k = 2; k < eu.length;k++ ){
    if((p1%k != 0) && (p2%k != 0) && (q1%k != 0) && (q2%k != 0)){
    for (int j=2;j<k;j++){
      if ((k % j == 0)){
      bil = 0;
      break;
    }
    else{
      bil = 1;
    }
  }
  switch (bil){
    case 0:
    break;
    case 1:
    eu[a]= k;
    a++;
    break;
  }
  }
}
int g[] = new int[eu.length];
for (int i=1;i<g.length;i++){
  s = R.nextInt(i+1);
  if (eu[s] != 0){
    bilangan = Integer.toString(eu[s]);
```

```
  }
}
pubKey.setText(bilangan);
}catch (Exception e){
}
}
});
helper = new AlmagHelper(this);
  almagId=getIntent().getStringExtra(Listkey.ID_EXTRA);
  if (almagId != null){
    load();
  }
};
@Override
public void onDestroy(){
  super.onDestroy();
  helper.close();
}
private void load(){
  Cursor c = helper.getbyId(almagId);
  c.moveToFirst();
  privKey.setText(helper.getNama(c));
  pubKey.setText(helper.getKey_e(c));
  NKey.setText(helper.getKey_N(c));
  c.close();
}

private boolean MenuChoice(MenuItem item)
{
  switch (item.getItemId()){
    case R.id.krm_sms:
      startActivity(new Intent(Privat.this, Main.class));
      return true;
    case 1:
      Toast.makeText(this, "Tentang", Toast.LENGTH_SHORT).show();
      return true;
    case R.id.keluar:
      Intent exit = new Intent(Intent.ACTION_MAIN);
      exit.addCategory(Intent.CATEGORY_HOME);
      exit.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
      Privat.this.finish();
      startActivity(exit);
      return true;
    case R.id.menu_settings:
      startActivity(new Intent(Privat.this, Listkey.class));
  return true;
}
return false;
}
@Override
public boolean onCreateOptionsMenu (Menu menu){
  getMenuInflater().inflate(R.menu.main, menu);
```

```
  return true;
}
@Override
public boolean onOptionsItemSelected(MenuItem item){
  return MenuChoice(item);
}
}
```

AlmagHelper.java

```
package com.kiplink.luc;

import android.content.Context;
import android.content.ContentValues;
import android.database.Cursor;
import android.database.sqlite.SQLiteOpenHelper;
import android.database.sqlite.SQLiteDatabase;
import android.util.Log;


class AlmagHelper extends SQLiteOpenHelper {
  private static final String TAG = "AlmagHelper";
  private static final String DATABASE_NAME = "LUC.db";
  private static final int SCHEMA_VERSION = 1;

  public AlmagHelper(Context context) {
    super(context, DATABASE_NAME, null, SCHEMA_VERSION);
  }

@Override
public void onCreate(SQLiteDatabase db) {
  db.execSQL("" +
   "CREATE TABLE publik (_id INTEGER PRIMARY KEY AUTOINCREMENT,  nama
   TEXT, key_e TEXT, key_N TEXT, jekel TEXT);" +
   "create table privat (id integer primary key autoincrement,
   privKey text, pubKey text, NKey text;");
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
   Log.w(TAG, "Upgrading database from version " + oldVersion + " to
   " + newVersion + ", which will destroy all old data");
  db.execSQL("drop table if exist titles");

}
public Cursor getAll() {
   return(getReadableDatabase().rawQuery("SELECT _id,  nama, key_e,
   key_N, jekel FROM publik ORDER BY nama", null));
}
public Cursor getbyId(String id){
  String[] args = {id};
```

```
   return (getReadableDatabase().rawQuery("select _id, nama, key_e,
   key_N, jekel from publik where _id=?", args));
}
public void insert(String nama, String key_e, String key_N, String
jekel){
  ContentValues cv=new ContentValues();
  cv.put("nama", nama);
  cv.put("key_e", key_e);
  cv.put("key_N", key_N);
  cv.put("jekel", jekel);
  getWritableDatabase().insert("publik", "nama", cv);
}
public void input(String privKey, String pubKey, String NKey){
  ContentValues cv = new ContentValues();
  cv.put("privKey", privKey);
  cv.put("pubKey", pubKey);
  cv.put("NKey", NKey);
}
public void update(String _id, String nama, String key_e, String
key_N, String jekel){
  ContentValues cv = new ContentValues();
  String [] args = {_id};
  cv.put("nama", nama);
  cv.put("key_e", key_e);
  cv.put("key_N", key_N);
  cv.put("jekel", jekel);
  getWritableDatabase().update("publik", cv, "_id=?",args);
}
public void up2date(String id, String privKey, String pubKey, String
NKey){
  ContentValues cv = new ContentValues();
  String [] args = {id};
  cv.put("privKey", privKey);
  cv.put("pubKey", pubKey);
  cv.put("NKey", NKey);
}
public String getNama(Cursor c){
  return (c.getString(1));
}
public String getKey_e(Cursor c){
  return (c.getString(2));
}
public String getKey_N(Cursor c){
  return (c.getString(3));
}
public String getJekel(Cursor c){
  return (c.getString(4));
}
}
```

Lampiran B. Layout

Main.xml

```
<RelativeLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:orientation="vertical"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  tools:context=".Luc" >
<TextView
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:text="Masukkan Nomor Tujuan"
  android:id="@+id/vNo" />
<EditText
  android:id="@+id/txtNum"
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:layout_below="@id/vNo"
  android:hint="Masukkan Nomor Tujuan"
  android:inputType="number|numberSigned|numberDecimal" />
<TextView
  android:id="@+id/vMessage"
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:text="Message"
  android:layout_below="@id/txtNum" />
<EditText
  android:id="@+id/txtMessage"
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:layout_below="@id/vMessage"
  android:hint="Pesan"
  android:inputType="textCapCharacters" />
<TextView
  android:id="@+id/vKey"
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:text="Key"
  android:layout_below="@id/txtMessage" />
<EditText
  android:id="@+id/txtKeyE"
  android:layout_width="150dp"
  android:layout_height="wrap_content"
  android:layout_below="@id/vKey"
  android:hint="Kunci e"
  android:inputType="number|numberSigned|numberDecimal" />
<EditText
  android:id="@+id/txtKeyN"
  android:layout_width="150dp"
```

```
  android:layout_height="wrap_content"
  android:layout_alignBaseline="@id/txtKeyE"
  android:layout_toRightOf="@id/txtKeyE"
  android:hint="kunci N"
  android:inputType="number|numberSigned|numberDecimal" />
<EditText
  android:id="@+id/txtH"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_alignBaseline="@id/txtKeyN"
  android:layout_toRightOf="@id/txtKeyN"
  android:visibility="invisible" />
<EditText
  android:id="@+id/txtH2"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_alignBaseline="@id/txtH"
  android:layout_toRightOf="@id/txtH"
  android:visibility="invisible" />
<EditText
  android:id="@+id/txtHasil"
  android:layout_width="fill_parent"
  android:layout_height="100dp"
  android:layout_below="@id/txtKeyE"/>
<Button
  android:id="@+id/btnEnk"
  android:layout_width="80dip"
  android:layout_height="wrap_content"
  android:layout_alignParentLeft="true"
  android:layout_below="@+id/txtHasil"
  android:layout_marginLeft="14dp"
  android:text="Encrypt" />
<Button
  android:id="@+id/btnSend"
  android:layout_width="80dip"
  android:layout_height="wrap_content"
  android:layout_alignBaseline="@+id/btnEnk"
  android:layout_alignBottom="@+id/btnEnk"
  android:layout_alignRight="@+id/txtKeyN"
  android:text="Send" />
</RelativeLayout>
```

Inbox.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:orientation="vertical"
  android:background="#40E0D0" >
<Button
  android:id="@+id/btn_new"
```

```
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:text="New Message" />
<ListView
  android:id="@+id/smsList"
  android:layout_width="match_parent"
  android:layout_height="wrap_content"/>
  <RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:gravity="bottom"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
  </RelativeLayout>
</LinearLayout>
```

Baca_sms.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:orientation="vertical" >
<TextView
  android:id="@+id/title"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="Pesan"
  android:textSize="17dp"
  android:textStyle="bold"/>
<EditText
  android:id="@+id/isi_pesan"
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:layout_below="@id/title"
  android:ems="10" />
<TextView
  android:id="@+id/label2"
  android:layout_below="@id/isi_pesan"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="Kunci"/>
<EditText
  android:id="@+id/key_p"
  android:layout_width="100dp"
  android:layout_height="wrap_content"
  android:layout_below="@id/label2"
  android:inputType="number|numberSigned|numberDecimal"
  android:hint="Kunci p">
<requestFocus />
</EditText>
<EditText
```

```
  android:id="@+id/key_q"
  android:layout_width="100dp"
  android:layout_height="wrap_content"
  android:layout_alignBaseline="@id/key_p"
  android:layout_toRightOf="@id/key_p"
  android:inputType="number|numberSigned|numberDecimal"
  android:hint="Kunci q" />
<Button
  android:id="@+id/btnDekrip"
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:layout_below="@id/key_p"
  android:text="Dekripsi" />
<EditText
  android:id="@+id/txtHasil"
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:layout_below="@id/btnDekrip"
  android:ems="10"
  android:hint="Hasil" />
<EditText
  android:id="@+id/tester"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_below="@+id/txtHasil"
  android:visibility="invisible" />
<EditText
  android:id="@+id/key_e"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_above="@+id/btnDekrip"
  android:layout_toRightOf="@+id/key_q"
  android:inputType="number|numberSigned|numberDecimal"
  android:hint="Kunci e" >
</EditText>
</RelativeLayout>
```

Listkey.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ListView
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:id="@android:id/list" >
</ListView>
```

Detailkey.xml

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:id="@android:id/tabhost"
  android:layout_width="fill_parent"
```

```
  android:layout_height="wrap_content"
  android:stretchColumns="1">
<TableRow>
<TextView android:text="Nama : "/>
<EditText
  android:id="@+id/nama"
  android:hint="Masukkan Nama" />
</TableRow>
<TableRow >
<TextView android:text="Key e"/>
<EditText
  android:id="@+id/key_e"
  android:hint="Kunci e"
  android:inputType="number|numberSigned|numberDecimal" />
</TableRow>
<TableRow >
<TextView android:text="Key N"/>
<EditText
  android:id="@+id/key_N"
  android:hint="Kunci N"
  android:inputType="number|numberSigned|numberDecimal" />
</TableRow>
<TableRow >
<TextView android:text="Jenis Kelamin"/>
<RadioGroup android:id="@+id/jekel">
<RadioButton
  android:id="@+id/pria"
  android:text="Pria"/>
    <RadioButton
    android:id="@+id/perempuan"
    android:text="Perempuan"/>
  </RadioGroup>
</TableRow>
<Button android:id="@+id/btnSave"
  android:layout_width="80dp"
  android:layout_height="wrap_content"
  android:text="Save"/>
</TableLayout>
```

Privat.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical" >
<TextView
  android:id="@+id/title"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_alignParentTop="true"
  android:layout_centerHorizontal="true"
```

```
  android:text="Kunci Public" />
<TextView
  android:id="@+id/labelP"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_below="@+id/title"
  android:text="Bilangan Prima P" />
<EditText
  android:id="@+id/keyP"
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:layout_below="@+id/labelP"
  android:ems="10"
  android:hint="Bilangan Prima Pertama"
  android:inputType="number|numberSigned|numberDecimal" />
<TextView
  android:id="@+id/labelQ"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_below="@id/keyP"
  android:text="Bilangan Prima Q" />
<EditText
  android:id="@+id/keyQ"
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:layout_below="@id/labelQ"
  android:ems="10"
  android:hint="Bilangan Prima Kedua"
  android:inputType="number|numberSigned|numberDecimal" />
<TextView
  android:id="@+id/privKey"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_below="@id/keyQ" />
<TextView
  android:id="@+id/lblPub"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_below="@id/keyQ"
  android:text="Key Public   : " />
<TextView
  android:id="@+id/pubKey"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_alignBaseline="@id/lblPub"
  android:layout_below="@id/privKey"
  android:layout_toRightOf="@id/lblPub" />
<TextView
  android:id="@+id/lblN"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_below="@id/lblPub"
```

```
  android:text="Key N           : " />
<TextView
  android:id="@+id/NKey"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_alignBaseline="@id/lblN"
  android:layout_below="@id/pubKey"
  android:layout_toRightOf="@id/lblN" />
<Button
  android:id="@+id/btnPriv"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_below="@id/lblN"
  android:text="Bangkitkan" />
</RelativeLayout>
```

Row.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:orientation="horizontal"
  android:padding="1dp" >
<ImageView
  android:id="@+id/icon"
  android:layout_width="wrap_content"
  android:layout_height="fill_parent"
  android:layout_alignParentTop="true"
  android:layout_alignParentBottom="true"
  android:layout_marginRight="4dp"/>
<LinearLayout
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:orientation="vertical">
<TextView
  android:id="@+id/title"
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:layout_weight="1"
  android:gravity="center_vertical"
  android:textStyle="bold"
  android:singleLine="true"
  android:ellipsize="end"
  android:textColor="#4682B4"/>
<TextView
  android:id="@+id/key_e"
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:layout_weight="1"
  android:gravity="center_vertical"
  android:singleLine="true"
```

```
  android:ellipsize="end"/>
<TextView android:id="@+id/key_N"
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:layout_weight="1"
  android:gravity="center_vertical"
  android:singleLine="true"
  android:ellipsize="end"/>
</LinearLayout>
</LinearLayout>
```

Lampiran C. Tabel ASCII

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□