IMPLEMENTASI KRIPTOGRAFI PADA APLIKASI

by Murti 2022

IMPLEMENTASI KRIPTOGRAFI PADA APLIKASI MEMO BERBASIS ANDROID MENGGUNAKAN ALGORITMA RSA

Giri Adi Nuryanto[1], Hari Murti[2]
[1,2]Program Studi Sistem Informasi, Fakultas Teknologi Informasi, Universitas Stikubank
E-mail: [1]giriadn97@gmail.com, [2]hmurti076@gmail.com

ABSTRAK

Informasi merupakan sesuatu yang sangat penting dalam kehidupan. Semakin banyak informasi yang ingin disimpan maka akan semakin sulit untuk diingat. Banyak orang kemudian menyimpan informasi dalam bentuk catatan pada kertas atau buku harian.

Terdapat masalah yang ditemukan pada penggunaan catatan yang ditulis secara manual. Catatan pada kertas bisa rusak, hilang bahkan diubah oleh orang lain. Hal tersebut terjadi karena kurangnya kesadaran masyarakat dalam menjaga dan mengamankan informasi.

Dalam penelitian ini penulis mengimplementasikan algoritma RSA untuk mengamankan isi catatan kedalam pesan rahasia melalui proses enkripsi . Selanjutnya untuk melihat isi catatan asli memerlukan proses dekripsi. Untuk menerapkan hal itu perlu adanya aplikasi yang mampu menjaga kerahasiaan isi catatan. Aplikasi catatan yang diterapkan pada aplikasi android ini mampu menjaga kerahasiaan catatan kepada pengguna. Aplikasi juga dapat digunakan untuk bertukar informasi secara rahasia kepada pengguna lain.

Kata Kunci: *Android, Catatan, Enkripsi, Dekripsi, Algoritma RSA*

1. PENDAHULUAN

Informasi merupakan sesuatu yang sangat penting dalam kehidupan. Semakin banyak informasi yang ingin disimpan maka akan semakin sulit untuk diingat. Banyak orang kemudian menyimpan informasi dalam bentuk catatan pada kertas atau buku harian. Informasi yang ditulis pada catatan biasanya berisi sesuatu yang penting bagi penulisnya. Menyimpan informasi dalam bentuk catatan dirasa efektif untuk mengingat kebutuhan penulisnya. Permasalahan yang ditemukan adalah masih adanya celah atau kelemahan pada metode penyimpanan informasi yang ditulis secaara manual. Catatan yang ditulis pada kertas bisa rusak, hilang bahkan dapat diubah oleh orang lain. Hal tersebut terjadi karena kurangnya kesadaran masyarakat dalam menjaga dan mengamankan suatu informasi.

Dari permasalahan di atas, diperlukan sebuah aplikasi yang mampu melindungi informasi yang ada pada catatan. Salah satu cara melindungi informasi adalah dengan kriptografi. Kriptografi berasal dari Bahasa Yunani, *crpto* yang berarti rahasia dan *graphia* yang berarti tulisan. Dengan kata lain kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan. Algoritma kriptografi terdiri dari tiga fungsi dasar yaitu enkripsi, dekripsi, dan kunci. Terdapat beberapa algoritma yang ada dalam ilmu kriptografi, salah satunya adalah algoritma RSA.

RSA merupakan algoritma kunci publik (asimetris) yang ditemukan pertama kali pada tahun 1977 oleh Ron Rivest, Adi Shamir, dan Len Adleman [1]. Nama RSA diambil dari inisial ketiga penemunya tersebut. Konsep dalam ilmu kriptografi RSA adalah penggunaan bilangan prima dan aritmatika untuk pembuatan kunci, proses enkripsi dan proses dekripsinya.

Berdasarkan permasalah di atas maka pada penelitian ini metode untuk mengamankan informasi pada catatan adalah dengan menggunakan algoritma RSA. Dengan dilakukan penelitian ini diharapkan keluaran atau aplikasi yang telah dibuat dapat berguna dalam menjaga kerahasiaan informasi dari orang yang tidak berwenang.

2. TINJAUAN PUSTAKA

Penelitian pertama yang dilakukan oleh Muammar Renaldy (2015) [2], dari Fakultas Teknologi Informasi Universitas Budi Luhur pada jurnal penelitian yang berjudul "Implementasi Kriptografi pada Diary Berbasis Mobile Android dengan Menggunakan Metode AES-128 (Advanced Encryption Standard-128) dan SHA-1 (Secure Hash Algorithm-1)" ia mengemukakan bahwa pada era digital yang serba otomatis saat ini masih ada kegiatan yang dilakukan secara manual, yaitu diary. Biasanya sesuatu yang ditulis dalam buku diary adalah rahasia yang bersifat privasi atau pribadi. Kekurangannya adalah isi cerita didalamnya masih bias dilihat dan dicuri oleh orang lain jika masih menggunakan buku sebagai medianya. Hal ini mendasari pembuatan aplikasi diary dengan metode enkripsi yang dapat dijalankan pada smartphone berbasis android, sehingga bisa digunakan dimana dan kapan saja. Metode yang digunakan dalam mengenkripsi isi diary adalah algoritma AES-128 sedangkan SHA-1 digunakan dalam pengacakan sandi enkripsi. Algoritma AES merupakan algoritma kriptografi kunci simetris yaitu menggunakan kunci yang sama pada proses enkripsi dan dekripsinya. Dengan diterapkannya algoritma AES-128 isi diary menjadi tidak dapat dibaca dan dengan diterapkannya SHA-1 pada proses enkripsi dekripsi kunci membuat isi kunci tidak terbaca di database oleh pihak luar. Penelitian kedua

yang dilakukan oleh Riad Sahara, Hendra Prastiawan dan Abdul Rohman (2017) [3], dari Fakultas Ilmu Komputer Universitas Mercu Buana pada jurnal penelitian yang berjudul "Implementasi Keamanan SMS Dengan Algoritma RSA pada Smartphone Android" , menjelaskan bahwa pengiriman informasi telah dipermudah dengan adanya fitur SMS (Short Message Service). Namun seiring berkembangnya waktu muncul beberapa kekurangan pada SMS salah satunya adalah tingkat keamanannya. Untuk mengatasi kekurangan tersebut maka diterapkan fungsi kriptografi pada aplikasi SMS. Algoritma kriptografi yang digunakan adalah metode RSA (Rivest, Shamir, Adleman). Pada penelitian ini membahas proses pengimplementasian kriptografi algoritma RSA untuk meningkatkan keamanan pada SMS dengan melakukan enkripsi dan dekripsi isi pesan. Pengirim melakukan enkripsi pada pesan yang akan dikirim menjadi chipper text dengan menggunakan kunci publik, sedangkan penerima pesan akan menerima pesan dalam bentuk chipper text yang kemudian didekripsi menjadi plain text menggunakan kunci privat. Hasil dari pengujian membuktikan bahwa algoritma RSA berhasil diimplementasikan untuk mengamankan pesan pada SMS. Kelebihan algoritma ini adalah perhitungan matematika yang digunakan sangat rumit dan disertai dengan dua kunci yang berbeda untuk proses enkripsi dan dekirpsi sehingga sangat sulit untuk ditembus dan cocok digunakan dalam pengamanan pesan pada SMS. Penelitian ketiga yang dilakukan oleh Richard Apau dan Clement Adamoko (2017) [4], dari Universitas Sains dan Teknologi Kwame Nkrumah, Kumasi, Ghana pada penelitian yang berjudul "Design of Image Steganography based on RSA Algorithm and LSB insertion for Android Smartphones" , menjelaskan dengan kemajuan modern dalam teknologi komunikasi masih banyak menimbulkan masalah karena fitur keamanan yang terbatas. Pendekatan untuk mengatasi masalah ini adalah kriptografi dan steganografi. Kriptografi berkaitan dengan penyembunyian konten secara rahasia sedangkan steganografi menyembunyikan pesan dari orang yang tidak berwenang. Berbagai format file seperti TIFF, JPEG, PNG, GIF dan BMP semuanya dapat diimplementasikan pada steganografi. Namun setiap format file memilike kelebihan dan kekurangannya masing-masing karena perbedaan dalam intensitas pixelnya. Pada penelitian ini pesan rahasia akan ditanamkan kedalam sebuah gambar dimana pesan rahasia telah dienkripsi terlebih dahulu menggunakan algoritma RSA. Hasilnya menunjukkan bahwa keamanan dan ketahanan yang tinggi dicapai dalam smartphone ketika kriptografi dikombinasikan dengan steganografi.

3. METODE PENELITIAN

3.1. Metode Prototype

Prototyping merupakan metode pengembangan perangkat lunak, yang berupa model fisik kerja system dan berfungsi sebagai versi awal dari system. Berikut merupakan tahapan dalam *prototype* :

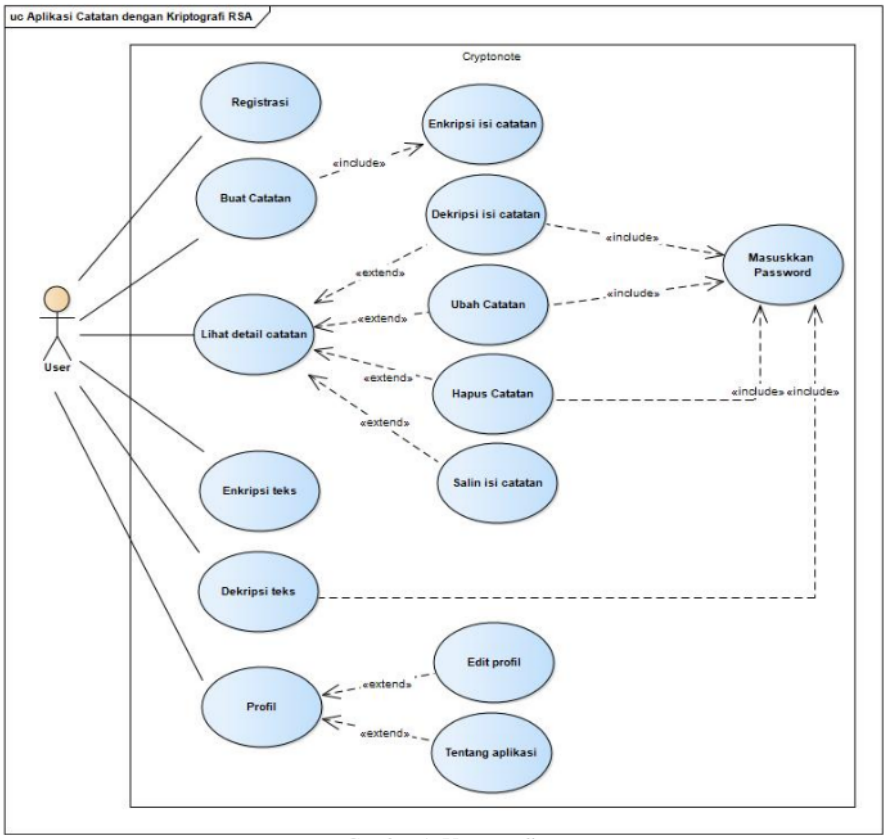
1. Fase Pengumpulan Kebutuhan
   Pada tahap ini pengembang dan pengguna bersama-sama mendefinisikan format dan kebutuhan perangkat lunak dan garis besar system yang akan dibuat. Diperoleh kebutuhan penting yang diperlukan antara lain enkripsi dan dekripsi isi catatan, perlunya *password* yang digunakan untuk menjalankan fungsi tertentu sehingga keamanan pengguna terap terjaga
2. Fase Perancangan
   Pada tahap perancangan atau membangun prototyping ini pengembang dan pengguna mulai membuat rancangan sistem yang akan dibuat seperti database yang dibutuhkan dan perancangan diagram system yang akan dibuat.
   Berikut ini merupakan database yang dibutuhkan dalam pembuatan aplikasi ini.

Tabel 1. Catatan

No.	Nama Field	Type Data	Null	Keterangan
1	ID_	Integer	Not Null	Id Catatan
2	TITLE_MESSAGE	Varchar	Null	Judul Catatan
3	VALUE_MESSAGE	Varchar	Null	Isi Catatan
4	CREATED_AT	Varchar	Not Null	Waktu Catatan Dibuat
5	UPDATE_AT	Varchar	Not Null	Waktu Catatan Diubah

Tabel pada database catatan diatas akan digunakan dalam pengolahan data catatan yang terdiri dari pembuatan catatan baru, melihat catatan, mengubah catatan dan menghapus catatan.

Berikut ini merupakan usecase diagram perancangan aplikasi.



Gambar 1. Usecase diagram

Gambar diatas menggambarkan rangkaian kegiatan yang terjadi dalam aplikasi yang diberi nama Cryptonote.

3. Fase Evaluasi
   Pada tahap ini pengguna aplikasi akan mengevaluasi prototype yang dibuat dan menjelaskan kebutuhan perangkat lunak yang diperlukan.

3.2. Algoritma RSA

Algoritma RSA merupakan salah satu algoritma yang ada dalam ilmu kriptografi. RSA merupakan salah satu kriptografi asimetris yang menggunakan 2 kunci yang berbeda untuk proses enkripsi dan dekripsinya. Kekuatan algoritma RSA terletak pada besarnya pemfaktoran bilangan prima yang dihasilkan yang akan digunakan dalam pembuatan kunci sehingga dibutuhkan waktu yang sangat lama untuk menentukan kunci yang dipakai.

Tahap awal dalam Algoritma RSA adalah proses pembentukan kunci yang dilanjutkan dengan proses enkripsi dan dekripsi. Tahapan dari algoritma ini adalah sebagai berikut [1]:

1. Proses pembuatan kunci

a. Menentukan nilai bilangan prima p dan q
   Keterangan:
   p : bilangan prima pertama
   q : bilangan prima kedua

b. Menghitung nilai n,
   $n = p.q$ (1)
   Keterangan:
   n : hasil perkalian antara p dan q q : bilangan prima kedua

p : bilangan prima pertama

c. Menghitung nilai totient n,

$$\phi n = (p-1)\,(q-1) \quad (2)$$

Keterangan:

ϕn = hasil perhitungan q =bilangan prima kedua

p = bilangan prima pertama

d. Menentukan nilai e,

$$\frac{\phi n}{e} = \text{bilangan pecahan} \quad (3)$$

dimana:

e = kunci enkripsi, e dicoba dari angka bilangan bulat hingga menghasilkan nilai pecahan

e. Menentukan nilai d,

$$d = \frac{k.\phi n+1}{e} \quad (4)$$

dimana:

d = kunci dekripsi

k = bilangan bulat yang dipilih sampai hasil perhitungan nilai d menjadi bilangan bulat

2. Proses enkripsi

$$c = m^e \bmod n \quad (5)$$

Keterangan:

c = chipper text

m = plain text dalam bilangan ascii

e = kunci enkripsi

n = hasil perhitungan 2 bilangan prima

3. Proses dekripsi

$$m = c^d \bmod n \quad (6)$$

Keterangan:

m = plain text

c = chipper text dalam bilangan ascii

d = kunci dekripsi

n = hasil perhitungan 2 bilangan prima

4. HASIL DAN PEMBAHASAN

4.1. Implementasi Algoritma RSA

Proses algoritma RSA diawali dengan membuat kunci pembangkit. Pada penelitian ini penulis memilih bilangan prima untuk nilai p adalah 61 dan nilai q adalah 53. Berikut merupakan proses pembentukan kunci enkripsi dan dekripsi:

```
2176-2176/com.tenpm.cryptonote D/e-Process: e (3120.0) = Qn (3120) / k (1.0)
2176-2176/com.tenpm.cryptonote D/e-Process: e (3120.0) % 1 = 0.0
2176-2176/com.tenpm.cryptonote D/e-Process: e (1560.0) = Qn (3120) / k (2.0)
2176-2176/com.tenpm.cryptonote D/e-Process: e (1560.0) % 1 = 0.0
2176-2176/com.tenpm.cryptonote D/e-Process: e (1040.0) = Qn (3120) / k (3.0)
2176-2176/com.tenpm.cryptonote D/e-Process: e (1040.0) % 1 = 0.0
2176-2176/com.tenpm.cryptonote D/e-Process: e (780.0) = Qn (3120) / k (4.0)
2176-2176/com.tenpm.cryptonote D/e-Process: e (780.0) % 1 = 0.0
2176-2176/com.tenpm.cryptonote D/e-Process: e (624.0) = Qn (3120) / k (5.0)
2176-2176/com.tenpm.cryptonote D/e-Process: e (624.0) % 1 = 0.0
2176-2176/com.tenpm.cryptonote D/e-Process: e (520.0) = Qn (3120) / k (6.0)
2176-2176/com.tenpm.cryptonote D/e-Process: e (520.0) % 1 = 0.0
2176-2176/com.tenpm.cryptonote D/e-Process: e (445.7142857142857) = Qn (3120) / k (7.0)
2176-2176/com.tenpm.cryptonote D/e-Process: e (445.7142857142857) % 1 = 0.7142857142857224
2176-2176/com.tenpm.cryptonote D/D-Process: d (445.85714285714283) = k (1) * Qn (3120) + 1 / e (7.0)
2176-2176/com.tenpm.cryptonote D/D-Process: d (891.5714285714286) = k (2) * Qn (3120) + 1 / e (7.0)
2176-2176/com.tenpm.cryptonote D/D-Process: d (1337.2857142857142) = k (3) * Qn (3120) + 1 / e (7.0)
2176-2176/com.tenpm.cryptonote D/D-Process: d (1783.0) = k (4) * Qn (3120) + 1 / e (7.0)
2176-2176/com.tenpm.cryptonote D/D-Process: Didapatkan d= 1783.0 , dari k= 4
```

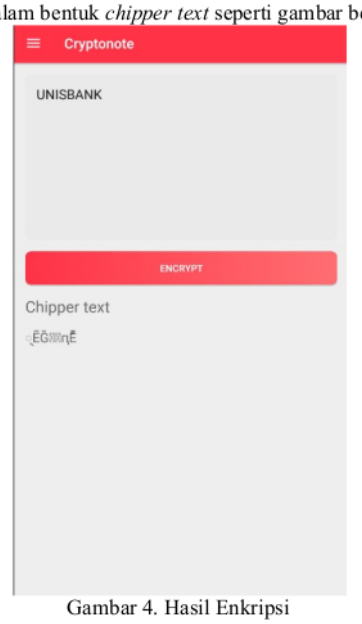
Gambar 2. Proses Pembuatan kunci enkripsi dan dekripsi

Dari proses pembuatan kunci pada gambar 2. menghasilkan nilai e sebagai kunci enkripsi adalah 7 dan nilai d sebagai kunci dekripsi adalah 1783. Kedua nilai tersebut akan digunakan dalam proses enkripsi dan dekripsi isi catatan sebagai berikut :

```
3282-3282/com.tenpm.cryptonote D/Encrypt: Konversi Char (U) ke ASCII (85)
3282-3282/com.tenpm.cryptonote D/Encrypt: Menghitung hasil ASCII (85) pangkat e (7.0) = 3.2057708828125E13
3282-3282/com.tenpm.cryptonote D/Encrypt: Menghitung hasil enkripsi dari (3.2057708828125E13) mod (3233) = 2509.0
3282-3282/com.tenpm.cryptonote D/TIME: finish 1487423319928
3282-3282/com.tenpm.cryptonote D/TIME: duration 63557
3282-3282/com.tenpm.cryptonote D/TIME: start 1487423336803
3282-3282/com.tenpm.cryptonote D/Encrypt: Konversi Char (N) ke ASCII (78)
3282-3282/com.tenpm.cryptonote D/Encrypt: Menghitung hasil ASCII (78) pangkat e (7.0) = 1.7565568854912E13
3282-3282/com.tenpm.cryptonote D/Encrypt: Menghitung hasil enkripsi dari (1.7565568854912E13) mod (3233) = 274.0
3282-3282/com.tenpm.cryptonote D/TIME: finish 1487423365229
3282-3282/com.tenpm.cryptonote D/TIME: duration 28426
3282-3282/com.tenpm.cryptonote D/TIME: start 1487423403476
3282-3282/com.tenpm.cryptonote D/Encrypt: Konversi Char (I) ke ASCII (73)
3282-3282/com.tenpm.cryptonote D/Encrypt: Menghitung hasil ASCII (73) pangkat e (7.0) = 1.1047398519097E13
3282-3282/com.tenpm.cryptonote D/Encrypt: Menghitung hasil enkripsi dari (1.1047398519097E13) mod (3233) = 286.0
3282-3282/com.tenpm.cryptonote D/TIME: finish 1487423438001
3282-3282/com.tenpm.cryptonote D/TIME: duration 34525
3282-3282/com.tenpm.cryptonote D/TIME: start 1487423450487
3282-3282/com.tenpm.cryptonote D/Encrypt: Konversi Char (S) ke ASCII (83)
3282-3282/com.tenpm.cryptonote D/Encrypt: Menghitung hasil ASCII (83) pangkat e (7.0) = 2.7136050989627E13
3282-3282/com.tenpm.cryptonote D/Encrypt: Menghitung hasil enkripsi dari (2.7136050989627E13) mod (3233) = 1825.0
3282-3282/com.tenpm.cryptonote D/TIME: finish 1487423478988
3282-3282/com.tenpm.cryptonote D/TIME: duration 28501
3282-3282/com.tenpm.cryptonote D/TIME: start 1487423491208
3282-3282/com.tenpm.cryptonote D/Encrypt: Konversi Char (B) ke ASCII (66)
3282-3282/com.tenpm.cryptonote D/Encrypt: Menghitung hasil ASCII (66) pangkat e (7.0) = 5.455160701056E12
3282-3282/com.tenpm.cryptonote D/Encrypt: Menghitung hasil enkripsi dari (5.455160701056E12) mod (3233) = 2241.0
3282-3282/com.tenpm.cryptonote D/TIME: finish 1487423519704
3282-3282/com.tenpm.cryptonote D/TIME: duration 28496
```

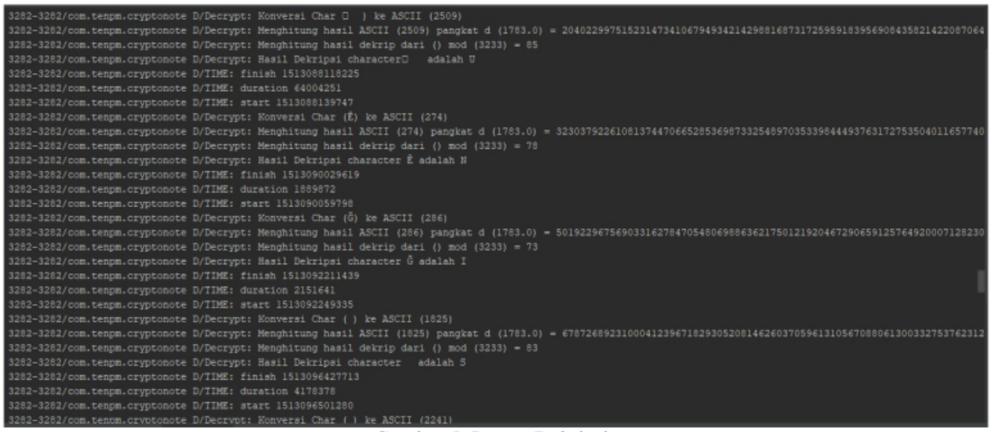
Gambar 3. Proses Enkripsi

Percobaan enkripsi dilakukan dengan kata “UNISBANK”, setiap huruf akan mengalami proses enkripsi dengan waktu tertentu kedalam bentuk *chipper text* seperti gambar berikut :



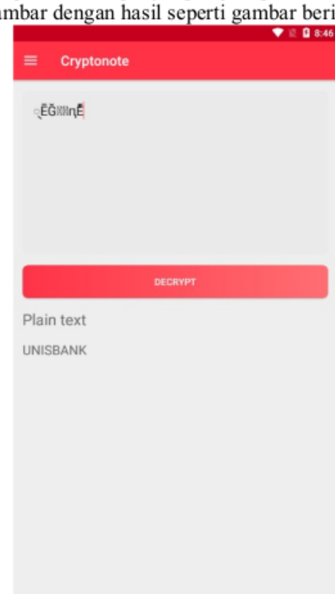
Gambar 4. Hasil Enkripsi

Pada gambar 4. Menunjukkan hasil dari enkripsi yang menghasilkan *chipper text* “ǫĒĞ▻□ɳĒ”. Selanjutnya pengguna dapat mengetahui isi teks asli melalui proses dekripsi sebagai berikut :



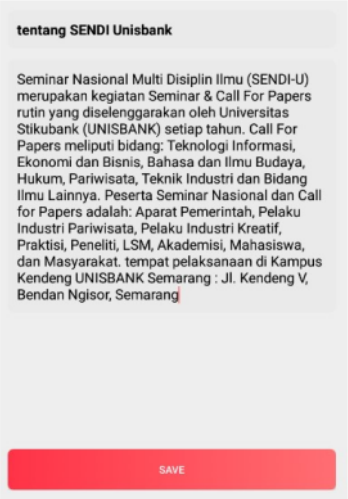
Gambar 5. Proses Dekripsi

Pada gambar 5. menujukkan proses dekripsi yang terjadi pada setiaf huruf *chipper text* yang membutuhkan waktu seperti pada gambar dengan hasil seperti gambar berikut:



Gambar 6. Hasil Dekripsi

Pada gambar 6. Menunjukkan hasil dari proses dekripsi dari *chipper text* "ҀĒĞ܁□חĒ̈" kembali menjadi *plain text* "UNISBANK" seperti semula. Berikut adalah contoh tampilan catatan dalam aplikasi pada bentuk *plain text* dan *chipper text.*



Gambar 7. Plain Text

Pada gambar 7. adalah tampilan form membuat catatan baru yang akan di enkripsi menjadi *chipper text* sebagai berikut :

Gambar 8. Chipper Text

Tabel 2. Hasil Pengujian

No	Pesan	Jumlah Karakter	Keterangan
1	0123456789	10	Berhasil
2	Abcdefghijklmnopqrstuvwxyz	26	Berhasil
3	ABCDEFGHIJKLMNOPQRSTUVWXYZ	26	Berhasil
4	!@#$%^&*()-_=+'"[]{};:,.	23	Berhasil

Pada tabel 2. menunjukkan bahwa pada empat kali pengujian, semua karakter dapat terenkripsi dan terdekripsi dengan baik dengan menggunakan algoritma RSA

5. KESIMPULAN

Berdasarkan hasil penelitian enkripsi dan dekripsi catatan menggunakan algoritma RSA, dapat diambil kesimpulan sebagai berikut :

1. Dari empat kali pengujian semua karakter dapat terenkripsi dan terdekripsi dengan baik.
2. Tidak semua karakter dalam ASCII dapat terbaca dengan baik pada android.
3. Semakin besar nilai bilangan prima p dan q maka semakin banyak pula waktu yang diperlukan untuk proses enkripsi dan dekripsi.

6. SARAN

Adapun saran yang diberikan untuk penelitian lebih lanjut yaitu sebagai berikut:

1. Hasil penelitian yang dilakukan saat ini pada fitur mengirim dan menerima pesan masih membutuhkan aplikasi lain dengan cara menyalin secara manual, kedepannya bisa dilakukan proses pengiriman dan penerimaan pesan secara langsung melalui aplikasi.
2. Pada fitur pertukaran informasi karena nilai p dan q masih statik, kedepannya dapat dilakukan pengembangan sistem yang dapat membuat nilai p dan q dapat diubah sesuai keinginan pengguna langsung dari aplikasi.
3.

DAFTAR PUSTAKA

[1] Ginting, A., Isnanto, R. R., and Windasari, I. P. (2015) 'Implementasi Algoritma Kriptografi RSA untuk Enkripsi dan Dekripsi Email', *Jurnal Teknologi dan Sistem Komputer*, Vol 3.

[2] Renaldy, M. (2015) 'Implementasi Kriptografi pada Diary Berbasis Mobile Android dengan menggunakan Metode AES-128 (Advanced Encryption Standard-128) dan SHA-1 (Secure Hash Algorithm-1)' , *Skripsi.* Program Strata 1 Teknik Informatika, Universitas Budi Luhur, Jakarta.

[3] Sahara, R., Prastiawan, H., and Rohman, A. (2017) 'Implementasi Keamanan SMS Dengan Algoritma RSA Pada Smartphone Android' , *Jurnal Ilmiah FIFO,* Vol 9.

[4] Apau, R. and Adomako, C. (2017) 'Design of Image Steganography based on RSA Algorithm and LSB Insertion for Android Smartphones', *International Journal of Computer Applications.* Vol 164.

[5] Satriawan, D., Sasmita, G. M. A., and Bayupati, P. A. (2014) 'Aplikasi Enkripsi SMS dengan Metode RSA pada Smartphone Berbasis Android' , *Merpati,* Vol 2.

IMPLEMENTASI KRIPTOGRAFI PADA APLIKASI

ORIGINALITY REPORT

13%	13%	11%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

5%

★ Submitted to Universitas Brawijaya

Student Paper

Exclude quotes	On	Exclude matches	< 1%
Exclude bibliography	On		