ISSN PRINT : 2302-500X
ISSN ONLINE : 2477-3786

IMPLEMENTASI KRIPTOGRAFI DAN KOMPRESI SMS MENGGUNAKAN ALGORITMA RC6 DAN ALGORITMA HUFFMAN BERBASIS ANDROID

Laurentinus [1)]

[1)]*Program Studi Teknik Informatika STMIK Atma Luhur*
Jl. Jendral Sudirman, Selindung Baru, Kec. Gabek Pangkal pinang Kode Pos 33117
Email : laurentinus99@gmail.com[1)]

ABSTRACT

The development of telecommunications technology in the era of globalization happens very rapidly to help people to communication and one of the communication technology is using telephone device. However, the absence of security in securing data/information of delivery Short Message Service (SMS) that is confidential causes the message easily stolen by unauthorized parties. SMS sent via the BTS will be accepted Message Service Center (SMSC) been causing a crack misuse of the information. One way to secure the data from the data communication process is by cryptography. This research using descriptive statistical research methods. This research was conducted by analyzing the security by implementing the RC6 algorithm in cryptography and compression applications based on android SMS. The RC6 algorithm is an symmetric cryptographic algorithm that use the same key in encryption and decryption of SMS message. The encryption process that changes the number bits of data affects the number of SMSs that need to be sent, so text compression is required. The Huffman algorithm is a lossless data compression algorithm in which data compression does not eliminate one byte and is stored as originally. So the result of decompress is same with original ciphertext. Testing is done using Blackbox Method.This research was conducted to solve the security problem of sending SMS with cryptography and send data can be compressed to save SMS cost.
***Keywords** : Cryptography, Encryption, Decryption, SMS, Mobile, RC6, Huffman,android*

1. Pendahuluan

Perkembangan teknologi saat ini telah memberikan pengaruh yang sangat besar, cepat dan pesat bagi dunia teknologi informasi dan telekomunikasi. Munculnya beragam aplikasi memberikan pilihan dalam peningkatan kinerja suatu pekerjaan, baik yang bersifat *desktop base, web based* hingga sekarang ini munculnya aplikasi-aplikasi baru yang berjalan dalam *mobile* pada sistem *platform android.* Saat ini muncul teknologi baru dimana komunikasi tanpa menggunakan kabel, seperti dengan menggunakan Media Internet yang bersifat *client server* pada *mobile android*.

Telepon selular menyediakan berbagai fitur dan salah satunya adalah media SMS (*Short Message Service*). SMS merupakan suatu layanan yang memungkinkan pengguna untuk saling berkomunikasi dengan mengirimkan pesan singkat berupa teks dengan cepat dan biaya yang kecil.

SMS ditujukan untuk mengirim pesan yang bukan bersifat sensitif/rahasia dikarenakan tidak adanya keamanan yang menjaga isi pesan dari pihak tidak berwenang, maka dari itu diperlukan aplikasi yang dapat menjamin keamanan dan keutuhan data dari transaksi pengiriman pesan SMS.Pemilihan algoritma yang tepat merupakan aspek yang penting, dilihat dari tingkat kepentingan dan kerahasiaan data. Algoritma yang baik yaitu menghasilkan enkripsi yang unpredictable dan tidak bisa dibongkar menggunakan cara apapun.

Algoritma RC6 merupakan salah satu metode enkripsi yang menggunakan kunci simetri dan berfungsi menjaga kerahasiaan data.

A. Penelitian Terdahulu

Penelitian "Implementasi Keamanan File dengan Kompresi Huffman dan Kriptografi menggunakan Algoritma RC4 serta Steganografi menggunakan End of File berbasis Desktop pada SMK Negeri 3 Kota Tangerang"telah dilakukan Nurhardian untuk menjaga keamanan dan kerahasiaan data murid dan guru. Ditambahalgoritma Huffman untukmengomprespesan[1].

Penelitian "Kompresi teks menggunakan algoritma Huffman dan Md5 pada Instant Message Smartphone Android"telah dilakukan M. Taofik memungkinkan pengguna dalam jaringan untuk mengirim pesan singkat menggunakan Android social media dengan pengamanan MD5 dankompresi Huffman [2].

Penelitian "Implementasi Kombinasi Algoritma Enkripsi AES 128 dan Algoritma Kompresi Shannon-Fano" telah dilakukan Heri Haryanto untuk menjaga agar data atau informasi tetap aman tanpa gangguan pihak ketiga dari algoritma Enkripsi AES 128 dan kompresi algoritma Shannon-Fano dapat membantu ukuran data menjadi lebih kecil [3].

Penelitian "Implementasi Keamanan Teks dengan Algoritma Modifikasi Caesar Cipher dan Mengkompresi File menggunakan Algoritma Shannon Fano" telah dilakukan Sinar Perkasa untuk menerapkan algoritma enkripsi untuk mengamankan pesandan algoritma kompresi untuk menghemat memori penyimpanan dan biaya pengiriman dari pesan [4].

Penelitian "Pengamanan Data denganKompresi LZSS dan Enkripsi IDEA berbasis Web" telah dilakukan Raka Yusuf danArif Wirawan menjamin keamanan data

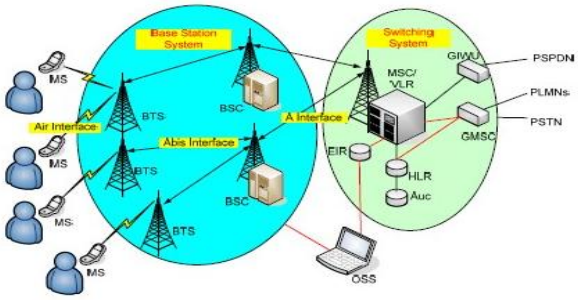
ISSN PRINT : 2302-500X
ISSN ONLINE : 2477-3786

penting supaya tidak dapat dibaca orang yang tidak a. berhak dimana dikompresi dengan teknik LZSS.[5]

B. Landasan Teori

1) SMS -

Layanan SMS telah digunakan disetiap bidang - seperti perbankan untuk transaksi m-banking, bidang *e-commerce*, dan *personal*.



Gambar 1. *Struktur layanan GSM*

Suatu pesan sms dikirim melalui MS (Mobile Station) melewati BTS (*Base Transceiver Station*) dan diarahkan dari MSC ke Short Message Service Center (SMSC). SMSC berfungsi menyimpan pesan tersebut sebelum dikirim kepada penerima.

2) Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler seperti *smartphone* dan komputer tablet. Android bersifat *open source* dan google merilis kodenya dibawah lisensi Apache.

Fitur yang tersedia di Android adalah Kerangka Aplikasi, Virtual Device Manager, Grafik, SQLite, Networking, Audio, Camera, dll.

3) Kriptografi

Kriptografi berasal dari bahasa Yunani *kryptós*, "tersembunyi, rahasia"; dan *graphein*, "menulis". Merupakan keahlian dan ilmu dari cara-cara untuk komunikasi aman pada kehadirannya di pihak ketiga.

Berbagai aspek dalam keamanan informasi seperti kerahasiaan data, integritas, autentikasi dan non-repudansi merupakan pusat dari kriptografi modern.

Algoritma adalah prosedur langkah demi langkah untuk menyelesaikan masalah secara sistematis. Algoritma Kriptografi merupakan langkah-langkah logis dengan tujuan menyembunyikan pesan dari pihak yang tidak berwenang .

Dasar dari Kriptografi adalah :

1. Enkripsi, merupakan proses mengamankan suatu informasi dengan membuat informasi tersebut menjadi kode-kode yang tidak dapat dibaca.
2. Dekripsi, merupakan proses kebalikan dari enkripsi, yaitu mengembalikan kode-kode ke bentuk aslinya.
3. Kunci, merupakan sandi dalam melakukan enkripsi

Berdasarkan kesamaan kuncinya, algoritma kriptografi dibedakan menjadi :

Symetric-key atau Kunci Simetri yaitu algoritma yang menggunakan sebuah kunci sandi yang sama untuk proses enkripsi dan dekripsi.

Algoritma yang menggunakan Kunci Simetri adalah *Data Encryption Standard* (DES)RC2, RC4, RC5, RC6 *International Data Encryption Algorithm* (IDEEA) *Advanced Encryption Standard Blow fish A symetric-key* atau Kunci Asimetri, menggunakan sepasang kunci yang berbeda dalam proses enkripsi dan dekripsi. Umumnya disebut *public-key* dan *private-key*.

Algoritma yang menggunakan Kunci Publik adalah RSA, Kriptografi Quantum, dll

3) Algoritma RC6

Algoritma RC6 merupakan salah satu kandidat *Advanced Encryption Standard* (AES) yang diajukan oleh RSA *Security Laboratories* kepada NIST. Dirancang oleh Ronald L Rivest, M.J.B. Robshaw, R. Sidney dan Y.L. Yin, algoritma ini adalah pengembangan dari algoritma RC5 dan telah memenuhi semua persyaratan yang diajukan oleh NIST. RC6 adalah algoritma yang menggunakan ukuran blok hingga 128 bit, dengan ukuran kunci yang diguankan bervariasi antara 128, 192 dan 256 bit.

4) Algoritma Huffman

Algoritma Huffman merupakan algoritma kompresi yang tertua dan disusun oleh David Huffman pada tahun 1952.Algoritma ini merupakan jenis *lossless data compression* dimana kompresi data tidak menghilangkan satu byte pun dan disimpan sesuai aslinya.

5) Graf

Graf merupakan gambaran dari masalah yang dibuat menjadi sekumpulan simpul yang berhubungan menggunakan garis[ARB 2010].

Jenis-jenis *graf*yaitu :

1. *Simple Graph*
   Graf ini tidak mengandung gelang maupun sisi ganda.
2. *Unsimple Graph*
   Graf ini mengandung gelang yang dibagi 2 menjadi *multi graph* dan *pseudograph*.

6) Pohon (Tree)

Pohon (*tree*) merupakan konsep yang mengambarkan *graf* tak berarah yang saling terhubung dan tidak mengandung sirkuit.

7) Pembentukan Pohon Huffman

Kode Huffman merupakan kode prefiks yang berisi sekumpulan kode *biner* dan karakteristiknya yaitu tidak ada kode *biner* yang menjadi awal bagi kode *biner* yang lain.

Kode prefiks digambarkan sebagai pohon *biner* yang diberikan nilai atau label. Untuk cabang kiri pada pohon biner diberi label 0 sedangkan pada cabang kanan diberi label 1. Rangkaian bit yang terbentuk pada setiap lintasan dari *root* ke *leaf* merupakan kode prefiks untuk

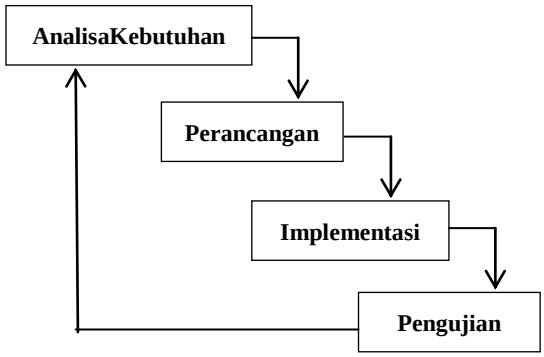
ISSN PRINT : 2302-500X
ISSN ONLINE : 2477-3786

karakter yang berpasangan sehingga menghasilkan pohon Huffman.

C. Metode Penelitian

Penjelasan dari deskripsi langkah penelitian tersebut sebagai berikut :

1. Melakukan survei awal
   Survei awal dalam penelitian ini dilakukan dengan teknik pengamatan untuk mendapatkan data-data yang dibutuhkanpengguna.
2. Melakukan studi pustaka
   Penelitian ini dimulai dengan melakukan studi pustaka yang berkaitan dengan algoritma RC6 dan algoritma Huffman. Studi pustaka dilakukan dengan mempelajari kriptografi, metode enkripsi/dekripsi, alurkerja algoritma RC6 serta metodekompresi data menggunakan algoritma Huffman pada aplikasi SMS.
3. Membuat kerangka konsep
   Data-data yang telah dikumpulkan dari observasi dan studi pustaka kemudian dibuat kerangka konsep. Kerangka konsep merupakan landasan atau *blueprint* untuk membuat penelitian menjadi terstruktur.
4. PerancanganSistem
   Hasil kerangka konsep digunakan untuk merancang system
5. Implementasi
   Sistem yang telah selesai didesain maka dilakukan Implementasi dan Pengujian



Gambar 2. *Langkah-langkah pembangunan sistem*

2. Pembahasan

A. Algoritma RC6

Algoritma RC6 dilengkapi dengan beberapa parameter, sehingga dituliskan sebagai RC6-w/r/b. Parameter w merupakan ukuran kata dalam satuan bit, parameter merupakan bilangan bulat yang menunjukan banyaknya iterasi selama proses enkripsi dan parameter b menunjukkan ukuran kunci enkripsi dalam byte. Setelah algoritma ini masuk dalam kandidat AES, maka ditetapkan bahwa nilai w = 32, r=20 dan b bervariasi antara 16, 24 dan 32 byte.

RC6-w/r/b memecah blok 128 bit menjadi 4 buah blok 32-bit, dan mengikuti aturan enam operasi dasar sebagai berikut :

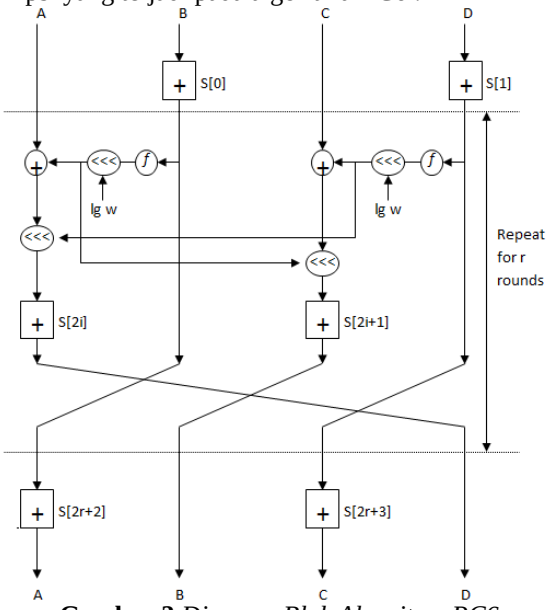
a + b	operasi penjumlahan bilangan *integer*
a - b	operasi pengurangan bilangan *integer*
a ⊕ b	operasi exclusive-OR (XOR)
a x b	operasi perkalian bilangan integer
a<<<b	a dirotasikan kekiri sebanyakvariabel kedua (b)
a>>>b	a dirotasikan kekanan sebanyak variabel kedua(b)

B. Algoritma enkripsi Kriptografi RC6

Karena RC6 memecah blok 128 bit menjadi 4 buahblok 32 bit, maka Algoritma ini bekerja dengan 4 buah register 32-bit A, B, C, D.

Byte yang pertama dari *plaintext* atau *ciphertext* di tempatkan pada byte A, sedangkan byte yang terakhirnya ditempatkan pada byte D. Dalam prosesnyaakan didapatkan (A, B, C, D) = (B, C, D, A) yang diartikan bahwa nilai yang terletak pada sisi kanan berasal dari register disisi kiri.

Diagram blok berikut akan lebih menjelaskan proses enkripsi yang terjadi pada algoritma RC6 :



Gambar 3.*Diagram Blok Algoritma RC6*

Algoritma RC6 menggunakan 44 buah sub kunci yang dibangkitkan dari kunci dan dinamakan dengan S[0] hingga S[43]. Masing-masing sub kunci panjangnya 32 bit. Khiri dengan proses *whitening* yang bertujuan untuk menyamarkan iterasi yang pertamadan yang

Proses enkripsi pada algoritma RC6 dimulai dan dia terakhirdari proses enkripsi dan dekripsi. Pada proses whitening awal, nilai B akan dijumlahkan dengan S[0], dan nilai D dijumlah kandengan S[i].

Pada masing-masing iterasi pada RC6 menggunakan 2 buah sub kunci. Sub kunci pada iterasi yang pertama menggunakan S[2] dan S[3], sedangkan iterasi-iterasi berikutnya menggunakan sub-sub kunci lanjutannya. Setelah iterasi ke-20 selesai, dilakukan proses *whitening* akhir dimana nilai A dijumlahkan dengan S[42], dan nilai C dijumlahkan dengan S[43].

Setiap iterasi pada algoritma RC6 mengikuti aturan sebagai berikut, nilai B dimasukan ke dalam fungsi f,

ISSN PRINT : 2302-500X
ISSN ONLINE : 2477-3786

yang didefinisikan sebagai f(x) =x(2x+1) , kemudian diputar ke kiri sejauh lg-w atau 5 bit. Hasil yang didapat pada proses ini dimisalkan sebagai u. Nilai u kemudian di XOR dengan C dan hasilnya menjadi nilai C.

Nilai t juga digunakan sebagai acuan bagi C untuk memutar nilainya kekiri. Begitu pula dengan nilai u, juga digunakan sebagai acuan bagi nilai A untuk melakukan proses pemutaran kekiri.

Kemudian sub kunciS[2i] pada iterasi dijumlahkan dengan A, dan sub kunci S[2i+1] dijumlahkan dengan C. keempat bagian dari blok kemudian akan dipertukarkan dengan mengikuti aturan, bahwa nilai A ditempakan pada D, nilai B ditempatkan pada A, nilai C ditempatkan pada B, dan nilai (asli) D ditempatkan pada C. demikianiterasi tersebutakan terusberlangsung hingga 20 kali.

Function
ROTL (X:word32; Y:integer) → word32
(fungsi untuk merotasi bit sebanyak variabel kedua)

Algoritma Pembangkitan sub Kunci :
Input (kunci)
S[0] ← b7e15163
For I ← 1 to 43 do
S[i] ← S[i-1] + 9e3779b9
End for
A ← B ← i ← j ← 0
V ← 44
If (c > v) then
  v ← c
  v ← 4 * 3
For s ← 1 to v do
A ← S[i] ← ROTL ((S[i] + A + B) .3)
B ← L[j] ← ROTL (L[j] +A + B, A + B)
i ← (i+1) mod 44
j ← (j_1) mod c

Algoritma Enkripsi RC6
Proses Enkripsi meliputi 3 hal yaitu :

1. AlgoritmaWhitenning awal
   X[1] ← X[1] + S[0]
   X[3] ← X[3] + S[1]
2. Algoritma Iterasi
   For I ← 1 to 20 do
   t ← ROTL ((X[1] *(2 * X[1] + 1)), 5)
   u ← ROTL ((X[3] *(2 * X[3] + 1)), 5)
   x[0] ← (ROTL((X[0] XOR t), u)) + S[2*i]
   x[2] ← (ROTL((X[2] XOR u), t)) + S[2*I + 1]
   temp ← X[0]
   X[0] ← X[1]
   X[1] ← X[2]
   X[2] ← X[3]
   X[3] ← Temp
   End For
3. Algoritma Whitening Akhir
   X[0] ← X[0] + S[42]
   X[2] ← X[0] + S[43]

C. Algoritma Huffman

Proses Kompresi Algoritma Huffman sebagai berikut:

1. Menghitung Frekuensi
2. Membuat Pohon Huffman setiap karakter dimulai dari *root* ke *leaf* hingga menjadi 1 Pohon Huffman
3. Kemudian dilakukan encoding

```
int[] initFreqs = newint[257];
Arrays.fill(initFreqs, 1);
FrequencyTable freqTable =
newFrequencyTable(initFreqs);
HuffmanEncoder enc = new HuffmanEncoder(out);
enc.codeTree = freqTable.buildCodeTree();  // We
don't need to make a canonical code since we don't
transmit the code tree
int count = 0;
while (true) {
        int b = in.read();
        if (b == -1)
        break;
        enc.write(b);

        freqTable.increment(b);
        count++;
        if (count < 262144 &&isPowerOf2(count) ||
count % 262144 == 0)  // Update code tree
        enc.codeTree = freqTable.buildCodeTree();
        if (count % 262144 == 0)  // Reset frequency
table
        freqTable = new FrequencyTable(initFreqs);
}
enc.write(256);  // EOF
```

```
public CodeTree buildCodeTree() {
Queue<NodeWithFrequency>        pqueue        =
newPriorityQueue<NodeWithFrequency>();

//  Add  leaves  for  symbols  with  non-zero
frequency
for (int i=0; i<frequencies.length;i++) {
if (frequencies[i] > 0)
pqueue.add(new
NodeWithFrequency(newLeaf(i),i,frequencies[i]));
}

// Pad with zero-frequency symbols until queue
has at least 2 items
for(int    i    =    0;    i<frequencies.length&&
pqueue.size()< 2; i++){
if (i >= frequencies.length || frequencies[i] ==
0)
        pqueue.add(new      NodeWithFrequency(new
Leaf(i), i, 0));
   }

if (pqueue.size() < 2)
thrownew AssertionError();
// Repeatedly tie together two nodes with the
lowest frequency
while (pqueue.size() > 1) {
        NodeWithFrequency nf1 = pqueue.remove();
        NodeWithFrequency nf2 = pqueue.remove();
        pqueue.add(new NodeWithFrequency(
        newInternalNode(nf1.node, nf2.node),
        Math.min(nf1.lowestSymbol,
nf2.lowestSymbol),
        nf1.frequency + nf2.frequency));
}

// Return the remaining node
```

ISSN PRINT : 2302-500X
ISSN ONLINE : 2477-3786

```
Returnnew CodeTree((InternalNode)pqueue.remove().
node,frequencies.length);
}
```

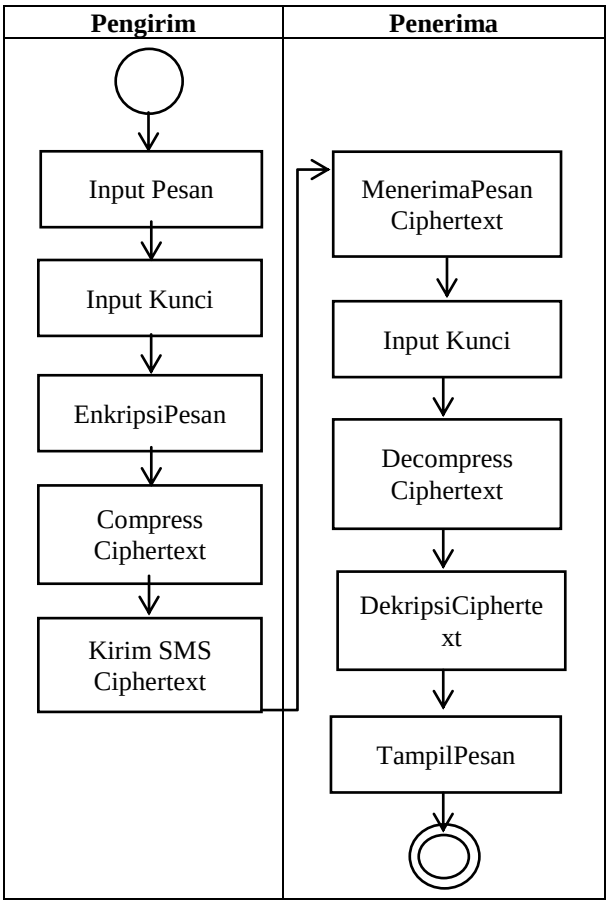
```
public CodeTree buildCodeTree() {
Queue<NodeWithFrequency> pqueue =
newPriorityQueue<NodeWithFrequency>();

// Add leaves for symbols with non-zero frequency
for (int i=0; i<frequencies.length;i++) {
if (frequencies[i] > 0)
pqueue.add(new
NodeWithFrequency(newLeaf(i),i,frequencies[i]));
}

// Pad with zero-frequency symbols until queue has at
least 2 items
for(int i = 0; i<frequencies.length&& pqueue.size()< 2;
i++){
if (i >= frequencies.length || frequencies[i] == 0)
pqueue.add(new NodeWithFrequency(new Leaf(i), i,
0));
}
if (pqueue.size() < 2)
thrownew AssertionError();
// Repeatedly tie together two nodes with the lowest
frequency
while (pqueue.size() > 1) {
NodeWithFrequency nf1 = pqueue.remove();
NodeWithFrequency nf2 = pqueue.remove();
pqueue.add(new NodeWithFrequency(
newInternalNode(nf1.node, nf2.node),
Math.min(nf1.lowestSymbol, nf2.lowestSymbol),
nf1.frequency + nf2.frequency));
}
// Return the remaining node
        Returnnew CodeTree((InternalNode)pqueue.re
move().node,frequencies.length);
        }
```

Proses Pengiriman Pesan :

1. Membaca No Telepon dan Pesan
2. Melakukan Enkripsi
3. Melakukan Kompresi sehingga menghasilkan root dan code list
4. Melakukan Encode ciphertext sehingga menghasilkan ciphertext terkompresi berupa special character dengan tipe data string.
5. Hasil kompresi dikirim ke nomor telepon tujuan menggunakan fungsi Sms Manager dimana tipenya Multipart Text Message.

```
Publicvoid kirimSMS(String noTelp , String
pesannya){
  try {
        SmsManager smsManager =
SmsManager.getDefault();
        ArrayList<String> msgArray =
smsManager.divideMessage(pesannya);
        smsManager.sendMultipartTextMessage(noT
elp, null, msgArray, null, null);
        Toast.makeText(getApplicationContext(),
"Message Sent",
        Toast.LENGTH_LONG).show();
  } catch (Exception ex) {
        Toast.makeText(getApplicationContext(),
        ex.getMessage().toString(),
Toast.LENGTH_LONG)
        .show();
        ex.printStackTrace();
  }
}
```

D. Flowchart Aplikasi



Gambar4.*Flowchart Aplikasi menggunakan algoritma RC6*

D. Implementasi

1) Halaman Utama

Berikut ini merupakan Halaman tampilan menu aplikasi Enkripsi dan Dekripsi SMS menggunakan algoritma RC6 yaitu: TulisPesan, Pesan Masuk dan About.

ISSN PRINT : 2302-500X
ISSN ONLINE : 2477-3786

Gambar 4.*Menu UtamaAplikasi RC6*

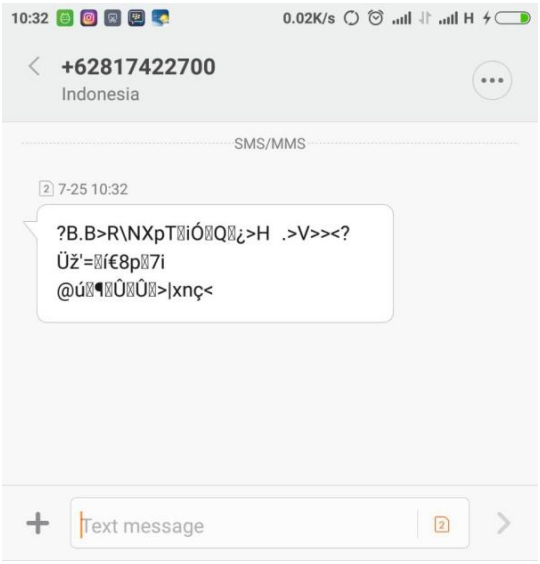
2) Halaman Menulis SMS

Halaman Tulis Pesan berisi : Nomor Tujuan, Kunci, Pesan, Hasil Enkripsi

Gambar 5. *Halaman Tulis Pesan Aplikasi RC6*

3) Halaman Menerima SMS

Menerima SMS tanpa menggunakan aplikasi



Gambar 6. *Tampilan SMS yang belum didekripsi*

4) Decrypt Menggunakan aplikasi

Tampilan Hasil pesan yang telah didekripsi menggunakan kunci yang sama dengan kunci enkripsi.

Gambar 7. *Halaman Decrypt SMS Algoritma RC6*

E. Pengujian Blackbox

Pengujian Sistem ini menggunakan metode pengujian *blackbox*. Pengujian *blackbox* ini tidak perlu tahu apa

ISSN PRINT : 2302-500X
ISSN ONLINE : 2477-3786

yang terjadi dalam sistem, pengujian *blackbox* berfokus pada persyaratan fungsional sistem.

Tabel 1. *Pengujian Blackbox*

Uji	ButirUji	Info
Interface	AksesHalamanUtama	Baik
Interface	AksesHalaman Create SMS	Baik
Interface	AksesHalaman Inbox	Baik
Interface	AksesHalaman Baca SMS	Baik
Interface	AksesHalaman About	Baik
RC6	Proses Enkripsi	Baik
RC6	Proses Dekripsi	Baik
Huffman	Proses mengcompresschipertext	Baik
Huffman	Proses Decompress chipertext	Baik

3. Kesimpulan

Berdasarkan hasil dari pengujian algoritma RC6, didapatkan kesimpulan seperti berikut :

1. Penerapan algoritma RC6 pada aplikasi SMS terbukti meningkatkan keamanan. Pesan yang terenkripsi tidak dapat dibaca jika tidak menggunakan kunci yang benar.
2. Algoritma RC6 menggunakan 1 kunci untuk dienkripsi, dan tidak bergantung pada panjang kunci. Semakin besar jumlah rotasi pada algoritma RC6 maka tingkat keamanan akan lebih baik.
3. Algoritma RC6 menggunakan 1 kunci untuk dienkripsi, dan tidak bergantung pada panjang kunci. Sehingga, waktu dalam generate kunci adalah sama, meskipun kunci yang diinput adalah 128, 192 atau 256 bit. Proses Enkripsi dan Dekripsi juga dilakukan secara cepat dan stabil.
4. Algoritma Huffman terbukti membantu kompresi data. Semakin panjang karakter maka semakin banyak byte yang dikompres.

Daftar Pustaka

[1] Nurhardian, Ahmad Pudoli. 2016. Implementasi Keamanan File dengan Kompresi Huffman dan Kriptografi menggunakan Algoritma RC4 serta Steganografi menggunakan End of File berbasis Desktop pada SMK Negeri 3 Kota Tangerang. *Jurnal TICOM* : 5(1)

[2] Taofik M, Chulkamdi, Sholeh Hadi Pramono, dan Erni Yudaningtyas, 2015. Kompresi Teks Menggunakan Algoritma Huffman dan MD5 pada Instant Messaging Smartphone Android.*Jurnal EECCIS* : 9(1)

[3] Haryanto, Heri Romi Wiryadinata, Muhammad Afif. 2014. Implementasi Kombinasi Algoritma Enkripsi AES 128 dan Algoritma Kompresi Shannon-Fano. *SETRUM*:3(1).

[4] Perkasa, Sinar, Muhammad Syahrizal, dan Pandi B. N. S., 2017. Implementasi Keamanan Data Teks dengan Algoritma Modifikasi Caesar Cipher dan Mengkompresi File Menggunakan Algoritma Shannon Fano. *Jurnal INFOTEK* : 2(1).

[5] Yusuf, Raka, dan Arif Wirawan. 2010. Pengamanan Data dengan Kompresi LZSS dan Enkripsi IDEA berbasis web.*SINAPTIKA.*

[6] Safaat, Nazruddin. 2013. *Aplikasi Berbasis Android*,Bandung : Informatika.

[7] Munir, Rinaldi. 2006. *Diktat Kuiah IF5054 Kriptografi*.Bandung

[8] Ariyus, Dony. 2006.*Kriptografi keamanan data dan komunikasi.* Yogyakarta:Graha Ilmu.

[9] Ariyus, Dony. 2008.*Pengantar ilmu Kriptografi, Teori, Analisis & Implementasi*. Yogyakarta:Andi.